Arbeitspapiere des Sonderforschungsbereichs 340
"Sprachtheoretische Grundlagen der Computerlinguistik"
Bericht Nr. 87, März 1997

Papers on Dynamic Semantics and Related Topics

Tim Fernado

Universität Stuttgart • Universität Tübingen
IBM Beutschland

ISSN 0947-6954/97
© 1997 Der Autor

Contents

Preface

Collected here are most of the papers that I have completed since arriving at the Institut für maschinelle Sprachverarbeitung in December 1993. As much as I wanted to, I have decided not to make any revisions, as that would have further delayed the appearance of the present report. Missing from this collection are (in addition to a number of attempts that got nowhere)

[Bpl] Bisimulations and predicate logic
[Wdr] What is a DRS?
[Lca] A logical connective for ambiguity requiring disambiguation
[Ccp] Are context change potentials functions?

and

[Xm] Expecting *many* (with Hans Kamp).

[Bpl] appears in the *Journal of Symbolic Logic* (Volume 59, Number 3, September 1994), while [Wdr] and [Lca] are in DYANA deliverables (R2.1.B, December 1994 and R2.2.B, September 1995, respectively). [Ccp] and [Xm] are planned for SFB reports with other authors. ([Xm] can also be found in the proceedings of SALT VI.)

As for the papers included here, each is intended to be more or less self-contained. Accordingly, they can be read in any order, although a few words might be helpful for orientation.

The DRT analysis of plurals and generalized quantifiers poses, as noted in a recent paper by Hans Kamp,[1] a problem for non-representational approaches (such as DPL). [1][2] proposes an arguably drastic solution, extending, as it

[1] "DRT and dynamic semantics: representational and non-representational accounts of anaphora."

[2] Papers in this report are numbered (for present reference) in the order in which they appear.

[1] Generalized quantifiers as second-order programs — "dynamically" speaking, naturally. In *Proceedings, Ninth Amsterdam Colloquium* (December 1993), Universiteit van Amsterdam.
[2] On the computational significance of consistency: from ordinary to generalized recursion theory. Presented at a conference on Proof Theory, Provability Logic and Computation (Berne) March 1994.
[3] Are boxes just parentheses? Manuscript, December 1994.
[4] Computational foundations for dynamic accounts of presuppositions. Presented at a workshop on Computational Logic for Natural Language Processing (Edinburgh), April 1995.
[5] A persistent notion of truth in dynamic semantics. In *Logic, Language and Computation*, J. Seligman and D. Westerståhl, eds. CSLI Lecture Notes Number 58, Stanford, 1996.

does, the notion of state in dynamic logic (in DRT terms, a model-embedding pair) to support parallel computation. Although the presentation in [1] is, I fear, somewhat dense (if not cluttered), it is easy enough now (three years later) to isolate its key idea: the program construct $\cdot_\forall$ that generates *conjunctive* branching, dual to the *disjunctive* branching underlying the dynamic treatment of indefinites. (Conjunctive and disjunctive branching co-exist in "and-or" graphs, familiar in analyses of games and in Concurrent Dynamic Logic.) [1] can, I think, be pushed much further: on the purely formal side, it provides some common ground for relating the different notions of state in dynamic semantic, game-theoretic and type-theoretic approaches; on the linguistic side, it raises the question of just what is at stake in choosing between the various options available. (I hope to say something about these matters in the near future.)

Instead of pursuing the complications presented by plurals, I turned my attention next to presupposition, out of the concern that the treatment of singular noun phrases that lay in the background of [1] was fatally flawed. Oversimplifying things a bit, [4] was my somewhat panicked reaction to *not* knowing "what comes first in dynamic semantics." Particularly alarming to me were claims that the model-embedding-pair notion of state (which I had taken for granted from DRT) is hopelessly inadequate for presupposition (not to mention Veltman's construct *might*). The conclusion I reached in [4] is that while model-embedding pairs do not tell the full story, they can nevertheless serve as useful starting points — at least, that is, for semantics conceived as an enterprise centered on questions of truth. But then how is "truth" to be understood within dynamic semantics? A reassuringly "classical" and disappointingly "static" answer is supplied in [5], a paper I have often regretted not giving the title "A Boolean-valued notion of truth for (a) change." The connection of [5] with presupposition rests on characterizations of presupposition error as the failure of bivalence. A bivalent interpretation amounts, in Boolean-valued semantics, to a "Boolean-valued" model, in which the Boolean algebra is the 2-element algebra **2**. As it is well known how

($*$) to turn an arbitrary Boolean-valued model into an ordinary **2**-valued model (through division by an ultrafilter satisfying certain "genericity" conditions necessitated by the quantifiers),

---

[6] Ambiguity under changing contexts. Presented at *Mathematics of Language 4* (Philadelphia), October 1995. To appear in a special issue of *Linguistics and Philosophy* consisting of selected papers from the conference.

[7] Non-monotonic consequences of preferential contextual disambiguation. In *Proceedings, Tenth Amsterdam Colloquium* (December 1995), Universiteit van Amsterdam.

[8] Analyzing ambiguity denotationally, operationally and logically. Manuscript, September 1996.

[9] Labeling representations for effective disambiguation (with Nicholas Asher). Presented in the Second International Workshop on Computational Semantics (Tilburg, The Netherlands), January 1997.

[10] In conjunction with qualitative probability. Manuscript, December 1996.

the problem of obtaining a bivalent interpretation reduces to

($**$) how to obtain a Boolean-valued model from dynamic semantics.

Now, [5] attacks ($**$) using a double-negation translation familiar from Heyting-valued semantics to reconcile two different conceptions of truth in dynamic semantics, and then applies a standard generic model construction for ($*$). Although there are perhaps a couple of nice twists in [5], the picture it provides is ultimately unsatisfying because of its *semantic* limitations: it leaves out the task of repair (or accommodation), which is arguably the difficult (*pragmatic*) part; and its solution to ($**$) depends on very strong assumptions about dynamic semantics.

For, whatever it is that comes first *in* dynamic semantics, there is no denying that what must come *before* formal semantics (of which dynamic semantics is an example) is an algorithm transforming a natural language statement into some logical form (fit for, say, model-theoretic analysis).[3] Such transformations are in general non-trivial; indeed, I believe that intrinsic limits on formalization (of natural language, in its fullest form) preclude the existence of a formal system $\mathcal{L}$ of logical forms that would support such a functional transformation. That is,

> **Thesis A**. There is *no* formal system $\mathcal{L}$ such that for every English statement s, there is, up to logical equivalence in $\mathcal{L}$, a unique logical form in $\mathcal{L}$ capturing s.

Implicit in Thesis A is the claim that there *is* a problem — in particular, that

(i) English statements are *in*formal, and hence cannot themselves serve as the aforementioned logical forms

and that

(ii) the binary relation holding between a logical form and an English sentence it "captures" is functional in *neither* its first *nor* its second argument.

Given (ii), Thesis A suggests a mismatch between formal and natural languages that smells a lot like *ambiguity*. A fully precise formulation of Thesis A is, by (i), presumably out of the question, although I think it is enough, as it stands, to convey the message that

(A) ambiguity is a problem that is here to stay.

Note that for those who take Church's thesis seriously, (A) would follow already from a weaker form of Thesis A, in which the universal-existential assertion mentioned can be realized mechanically.

> **Thesis A$'$**. There is *no* formal system $\mathcal{L}$ and *no* Turing machine e such that for every English statement s, e returns, on input s, a logical form in $\mathcal{L}$, unique up to logical equivalence in $\mathcal{L}$, capturing s.

[3] As stressed in Kamp's recent paper (identified in footnote 1), the DRS construction algorithm is abstracted out of the reformulation of DRT in dynamic semantics.

Missing from either Thesis A or A′ is a reference to *context-dependence* in *discourse* interpretation. I have left this out only for the sake of simplicity; the moral would, under a more complicated formulation, be the same: any algorithm taking natural language statements to logical forms must allow for non-determinism — or, in other words, ambiguity. That is not to say that context is unimportant for ambiguity; ambiguity is about nothing if not context. To illustrate this point, suppose an English statement s is "ambiguous" between two non-equivalent logical forms φ and ψ in $\mathcal{L}$. Why not combine φ and ψ into one logical form $\varphi \times \psi$, where the connective $\times$ is, say, disjunction $\vee$? Many reasons have been advanced in the literature; but for me the simplest and best reply is

> (D) what makes an expression ambiguous is the need for disambiguation, and that need is abstracted away when reducing a statement ambiguous between φ and ψ to $\varphi \times \psi$, for some connective $\times$ that (like $\vee$) requires no more disambiguation than conjunction.[4]

(D) is the point of departure in [Lca], the conclusion of which is that disambiguation crucially involves contexts — contexts that disambiguate and get disambiguated (the disambiguation in neither case being necessarily total). But *what contexts?*

This is, of course, a complicated question. Precious little is, I think, to be gained in attacking it in its full glory. The matter becomes more manageable if different notions of context — or (focussing on formal aspects I mentioned at the outset) *state* — are recognized, roughly according to the familiar semiotic division between syntax, semantics and pragmatics. This plea is made explicit in [Ccp], which asks: are context change potentials functions? In view of the previous paragraph, one might predict that my answer to this somewhat simple-minded query is: no, context change potentials cannot be functions, as non-determinism is required to support ambiguity and indefiniteness (among perhaps a trillion other things). But just as I would not lump the non-determinism in the indefinite noun phrase "a farmer" with ambiguity, neither would I deny the possibility of defining, at a suitable level of abstraction, context change potentials as (partial) functions. There are, I believe, more levels to consider than that of model-theoretic interpretation (with which some would equate formal semantics, fixing contexts to model-embedding pairs). Notable among these levels are, as already mentioned above, those at which logical forms are constructed. Good reasons exist for asserting that logical forms are constructed not only *before* formal semantics (dynamic or not), but also *after* and *during* it. In addition to *pre*supposition, there are *post*supposition/implicature and the usual kinds of disambiguation (e.g., anaphora resolution, lexical and structural disambiguation) that require semantic information. These phenomena cry out (to my ear) for logical analysis; some tentative steps in response are taken in [6-10]. Of these papers, I

[4] Surely, the union of two sets is no less well-defined than the intersection of two sets is?

wish to draw the reader's attention most to [6], which has a few things to say about compositionality and logical consequence in the presence of ambiguity. To bring out natural affinities with theoretical work in computer science, I have also tried a different presentation of [6] in [8]. All proofs have been taken out in [8], although (by way of compensation) [8] includes a few things not in [6]. A basic issue in logical consequence, the acceptability of a conjunction of two acceptable propositions, is explored in [10], a rather technical paper whose connection with ambiguity may seem a bit obscure to the reader unfamiliar with [7]. (By contrast, the relevance of [10] to [Xm] is, I trust, obvious enough.)

And what about the rest of the papers that have found their way into this collection? Despite reservations about the treatment of aspect presupposed there, I have decided to include [3] (a note small enough to slip between the cracks) because it fills a hole in Chapter 5 of Kamp and Reyle's *From Discourse to Logic* (Kluwer, 1993). Furthermore, [3] also led me, strangely enough, to [9], which turns to more computationally oriented work on ambiguity, revolving around underspecification. The computational concerns here are far more down-to-earth than in [2], a paper that may strike the reader as odd in the present company. Indeed, [2] was written in a quite different frame of mind, growing out of [Bpl], where an early draft of [2] is mentioned (titled "A path from generalized recursion back to mechanical computation"). Be that as it may, I am convinced that the abstract work on bisimulations described in [Bpl] and [2] is helpful for coming to grips with dynamic semantics. Much more is lurking here than is broached in [Wdr] — including applications to the construct $\cdot_\forall$ (presented in [1]) and to ambiguity. Having kept both [Bpl] and [Wdr] out of this collection, I leave [2] in as a hint at these broader applications.

February 11, 1997

Added: March 11, 1997. Delays in printing have made possible the inclusion here of "Processing ambiguity non-deterministically: a modal logic for disambiguation." (Given another delay, a paper addressing points mentioned above concerning [1] might, with some luck, be ready.)

1

Generalized quantifiers as second-order programs — "dynamically" speaking, naturally

ABSTRACT. Generalized quantifiers (understood as functions $\mathcal{Q}$ mapping a set M to a binary relation $\mathcal{Q}_M$ on $\mathrm{Pow}(M)$) are analyzed within the so-called "dynamic" approach to natural language semantics. Under that approach, the meaning of a formula is taken to be the set of input/output transitions it induces, according to a particular translation into programs from some formalism, such as that of (quantified) dynamic logic. The concrete problem addressed here is that variables bound by generalized quantifiers must introduce discourse markers for pronouns (that are typically plural). The solution adopted builds heavily on the "duality" between non-determinism and parallelism by interpreting programs as binary relations on non-empty subsets of states from dynamic logic, thereby accomodating the conjunctive branching in *and-or* computation graphs (which can then be weakened relative to a generalized quantifier). (The distinction between definite and indefinite noun phrases is linked to that between deterministic and non-deterministic programs.) The underlying notion of computation is argued to require a global (as opposed to a local) interpretation, complementing the uniformity in a generalized quantifier. The subsets of interest are hypothesized to have a "determinate" character (i.e., to be first-order definable), in accordance with so-called E-type pronouns. An alternative "indeterminate" reading is also formulated within the framework, as are readings that are weak or strong, and distributive or collective.

INTRODUCTION

An approach to natural language semantics that has attracted, and continues to attract, a good deal of attention is *Discourse Representation Theory* (DRT), initiated in Kamp [**15**] at around the same time (the early eighties) as that of the closely related but independent work of Heim [**13**]. Central to DRT is the proposal of a level of explicit semantic representation called *discourse representation structure*. This feature of the analysis, henceforth referred to as *representational*, was challenged in Barwise [**1**], where the essential point was argued instead to be to

> view the utterance of an expression more dynamically, as having an effect on the environment shared by speaker and hearer, the effect being represented by various sorts of changes in variable assignments. (p. 2)

Shortly afterwards, Groenendijk and Stokhof [**11**], confining their attention to first-order formulas, defined a simple non-representational translation, dubbed *Dynamic Predicate Logic* (DPL), of first-order formulas into programs from

(Quantified) Dynamic Logic (e.g., Harel [**12**]). More recently, further developments of DRT have appeared in the textbook Kamp and Reyle [**14**], far outstripping Kamp [**15**], Barwise [**1**] and Groenendijk and Stokhof [**11**]. These include an account of generalized quantifiers and plurals (going back to 1985/86) reported in chapter 4 of the book. Although certain notions in that chapter admit straightforward transcriptions to a DPL-framework (e.g., the two different semantics of *duplex conditions* in Kamp and Reyle [**14**], that are christened *weak* and *strong* readings in Chierchia [**5**]) and although some interesting extensions of DPL are suggested in Van den Berg [**2, 4**] (as well as Chierchia [**5**] and Van Eijck and De Vries [**7**]), it is not clear that a non-representational treatment of generalized quantifiers comparable in scope and perspicuity to that in Kamp and Reyle [**14**] has been worked out. The present paper is intended as a contribution towards that end.

The linguist is, of course, entitled to ask *why bother* developing yet another formalism with the same empirical coverage? In fact, it is not obvious to the present author that the scope of the proposal below is confined to chapter 4 of Kamp and Reyle [**14**]. But at any rate, the hope is that different formal approaches reflect different intuitions that throw light on issues with genuine "linguistic" content. An intuition stressed by the "dynamic" approach (exemplified by DPL) is that the meaning of a statement lies not so much in its *truth* relative to a context, but in the *change* in context that it induces. Formalizing statements as formulas, and contexts as states, the idea is captured by the slogan "formulas-as-programs", with a program π reduced semantically to its binary input/output relation $\rho(\pi) \subseteq S \times S$ on some set S of states. As will become clear below, the same scheme can be applied to Kamp [**15**], with the essential exception that the relevant states in Kamp [**15**] are the discourse representation structures mentioned above.

By contrast, to support a translation from first-order formulas to programs in quantified dynamic logic, the states in DPL are defined, relative to a first-order model $\mathbf{M}$, to be functions, henceforth called ($\mathbf{M}$-)*substitutions*, from a set of variables to the universe of $\mathbf{M}$. A substitution is nothing more than the "variable assignment" mentioned in the quotation above from Barwise [**1**], and is also referred to elsewhere as an "environment" or a "valuation." Although our choice of the term "substitution" is largely arbitrary, it can be motivated in part by (i) its compatibility with S, (ii) the unmistakable distinction made from certain programs (to be described shortly) called "random assignments", and (iii) the suggestion (e..g, from Barwise [**1**]) that an object in S have as its domain a finite subset of the full (infinite) set X of variables. (This suggestion departs form the conventions of dynamic logic, which are otherwise observed below.) While the present paper will refrain from making a commitment to (iii), except at one point indicated below, let us note in passing that item (iii) yields a more computationally accessible (i.e., realistic) picture that is also more faithful to DRT (in contrast, that is, to the usual convention that a variable assignment is defined on all of X). Keeping (iii) in mind, let us characterize the interpretation of a *random assignment* $x :=?$, where $x \in X$, by the condition that for all f and $g \in S$,

$$\begin{array}{lll} f\ \rho(x :=?)\ g & \text{iff} & x \in dom(g) \text{ and } f = g \text{ except} \\ & & \quad \text{possibly on } x \\ & (\text{iff} & dom(g) = dom(f) \cup \{x\} \text{ and} \\ & & (\forall x' \in dom(f) - \{x\})\ f(x') = g(x'))\ . \end{array}$$

All changes to a substitution in DPL can be traced to such a random assignment, which is used to interpret existential quantification $\exists x$. Conjunction & and negation $\sim$ are analyzed in terms of sequential composition (written ;) and a test for divergence (written $\neg$, instead of the somewhat more cumbersome notation $([\cdot]\bot)?$ of dynamic logic), respectively. The remaining logical connectives (including the universal quantifier) can then be derived as abbreviations in the familiar manner, sound for classical logic.

The details will be presented more carefully below, but for now let us stress the computational and dynamic aspects of DPL: a formula φ is translated into a program φ^{DPL}, resulting in an interpretation $[\varphi]$ of φ as the input/output relation $\rho(\varphi^{DPL}) \subseteq S \times S$. DPL's interpretation of negation, however, features two disturbing conjuncts,

$$f[\sim\varphi]g \quad \text{iff} \quad f = g \text{ and } f \notin dom[\varphi]\ ,$$

belying the propaganda above about change and computation, and raising two somewhat embarrassing questions.

(Q1) How well is the view that "meaning lies in change" served by reducing

$$\varphi \supset \psi \quad \text{to} \quad \sim(\varphi \& \sim\psi)$$

and

$$\forall x\varphi \quad \text{to} \quad \sim\exists x \sim\varphi\ ,$$

given that the negation of a program can only output its input

$$f[\sim\varphi]g \quad \text{implies} \quad f = g$$

(assuming it halts at all)? Do not such programs trivialize the shift from a static (truth-centered) semantics to a dynamic semantics?

(Q2) How seriously can the term "program" be taken, given that (under a recursion-theoretic conception) testing for divergence (i.e., the complement of the halting problem),

$$f \notin dom[\varphi]\ ,$$

falls beyond the realm of computation? In what sense are such programs computational?

We will address points (Q1) and (Q2) somewhat indirectly, by first focussing on a crucial feature of some programs in DPL — non-determinism.

A program π is *non-deterministic* if its input/output relation is *not* functional — i.e., if on some input, π can return one of at least two distinct outputs. Such a possibility is hard-wired in dynamic logic by the binary program connective $+$, which is interpreted as union

$$f\ \rho(\pi + \pi')\ g \quad \text{iff} \quad f\ \rho(\pi)\ g \ \text{ or } \ f\ \rho(\pi')\ g\ .$$

In Peleg [**16**], a binary program construct $\cap$ dual to $+$ is introduced yielding a notion of parallelism. Roughly put, if π and π' are programs from ordinary dynamic logic, then

$$f\ \rho(\pi)\ g \ \text{ and } \ f\ \rho(\pi')\ h \quad \text{imply} \quad f\ \rho(\pi \cap \pi')\ \{g, h\}\ ,$$

the intuition being that whereas $+$ is disjunctive, $\cap$ is conjunctive (which should be kept in mind when specifying transitions from $\{g, h\}$). More rigorously, $\cap$ is accomodated by lifting the function ρ to a function $\hat{\rho}$ with $\hat{\rho}(\pi) \subseteq S \times \mathrm{Pow}_+(S)$, where $\mathrm{Pow}_+(S) = \mathrm{Power}(S) - \{\emptyset\}$. In fact, we will not have any use for $\cap$, which (for the record) is governed by

$$\begin{aligned} f\ \hat{\rho}(\pi \cap \pi')\ U \quad \text{iff} \quad & (\exists V, W)\ U = V \cup W \ \text{ and} \\ & f\ \hat{\rho}(\pi)\ V \ \text{ and } \ f\ \hat{\rho}(\pi')\ W\ . \end{aligned}$$

Nonetheless, the passage from ρ to $\hat{\rho}$ (along the embedding $s \mapsto \{s\}$ of S into $\mathrm{Pow}_+(S)$) will be crucial to the constructs we will introduce, as will Peleg [**16**]'s clause for sequential composition

$$\begin{aligned} f\ \hat{\rho}(\pi; \pi')\ U \quad \text{iff} \quad & (\exists V)\ f\ \hat{\rho}(\pi)\ V \ \text{ and there is a function} \\ & F \text{ with domain } V \text{ s.t.} \\ & \qquad (\forall g \in V)\ g\ \hat{\rho}(\pi')\ F(g) \ \text{ and} \\ & \qquad U = \bigcup \{F(g) \mid g \in V\}\ . \end{aligned}$$

To understand this clause, think of a substitution as a process (or some snapshot of a process), and regard a set of substitutions (such as U and V above) as a family of proceses running in parallel. The universal quantification $(\forall g \in V)$ above formalizes the conjunctive nature of parallelism.

Our move from S to $\mathrm{Pow}_+(S)$ will be necessitated by the non-determinism not in $+$, but in random assignments $x := ?$. To a first approximation, the idea is to introduce a unary program construct $\cdot_\forall$ such that on programs π from dynamic logic,

$$f\ \hat{\rho}(\pi_\forall)\ U \quad \text{iff} \quad U = \{g \in S \mid f\ \rho(\pi)\ g\} \neq \emptyset\ ,$$

and, in particular,

$$\begin{aligned} f\ \hat{\rho}((x := ?)_\forall)\ \{g \in S \mid x \in \mathit{dom}(g) \text{ and } f = g \\ \text{except possibly on } x\}\ . \end{aligned}$$

(The effect of $\cdot_\forall$ can be pictured as turning the disjunctive branching of a non-deterministic program to the conjunctive branching of a deterministic program.) As $\hat{\rho}$ is somewhat uneven in that the left and right arguments of

$\hat{\rho}(\pi)$ have different types, it is natural to lift $\hat{\rho}$ further to a map $[\![\cdot]\!]$ that interprets a program π as a binary relation $[\![\pi]\!] \subseteq \text{Pow}_+(S) \times \text{Pow}_+(S)$. (The restriction to non-empty subsets of S is made because $\emptyset$ arises from the impossibility of making a ρ-transition.) Just how the lifting is carried out will require some elaboration, but very briefly, the point is that π becomes a second-order program that relates sets of substitutions operating in the conjunctive manner alluded to above. The unary program construct $\cdot_\forall$ can be derived from a binary program construct $\Rightarrow$ introduced in Fernando [**10**] to extend the DPL translation by the clauses

$$\begin{aligned} (\varphi \supset \psi)^{DPL} &= \varphi^{DPL} \Rightarrow \psi^{DPL} \\ (\forall x \varphi)^{DPL} &= (x :=?) \Rightarrow \varphi^{DPL} . \end{aligned}$$

This extension to DPL is "conservative" in that

$$\begin{aligned} {\sim}{\sim}(\varphi \supset \psi) &\equiv {\sim}(\varphi \& {\sim}\psi) \\ {\sim}{\sim}(\forall x \varphi) &\equiv {\sim}\exists x {\sim}\varphi , \end{aligned}$$

where $\varphi \equiv \psi$ means that φ and ψ are interpreted by the same binary relation (i.e., $[\varphi] = [\psi]$). Note that the double negation $\neg\neg\pi$ of a program π is interpreted as $\{(f, f) \mid f \in dom[\![\pi]\!]\}$. The troublesome negation $\neg\pi$ of a program π can be recast as $\pi \Rightarrow \bot$, where $[\![\bot]\!] = \emptyset$. But, in general, the point is that programs of the form $\pi \Rightarrow \pi'$ externalize the dynamic effects of programs $\neg(\pi; \neg\pi')$ used in DPL to interpret implication. In short, the construct $\Rightarrow$ forms our reply to (Q1).

As for (Q2), let us first try to put to rest any doubt about its significance. Referring to dynamic approaches to generalized quantifiers, Van den Berg [**3**] asserts that Van den Berg [**4**] "shares with Chierchia and Van Eijck" ([**5, 7**]) the property "that the dynamic behavior of the quantifier is totally **ad hoc**. The definition gives a condition in terms of the generalized quantifier and on top of that it gives some condition on the output ... There is no reason why either of these definitions, or any of the other proposed in the literature, is natural, let alone the correct one" (p. 2). Such reservations suggest that the computational intuitions behind the term "program" might be worth developing. In fact, the idea of using sets of substitutions (or "assignments") as "witnesses" occurs already in Van den Berg [**2, 4**], although the computational metaphor is left undeveloped there. We will return to Van den Berg [**2, 4**] later on, but for now let us briefly outline the present proposal.

Discourse markers arising from quantification are generated below by a deterministic program of the form

$$(*) \qquad (x :=? \ ; \ \pi)_\forall ,$$

where, roughly, the program π yields the *discourse representation structure* K in Kamp and Reyle [**14**]'s *abstraction* condition

$$X = \sum x : K .$$

The construct $\cdot_\forall$ has the additional virtue that it can be understood without mentioning substitutions or variables, and that, moreover, $(*)$ keeps the functional dependencies of x. Moving on, if a set of substitutions can grow by a program $(x :=?; \pi)_\forall$, that set can also be trimmed by a program

$$\mathcal{Q}x.\pi \ ,$$

where $\mathcal{Q}$ is some generalized quantifier. The program $\mathcal{Q}x.\pi$ (which will play a role similar to Kamp and Reyle [**14**]'s duplex condition) is essentially π, except that the conjunctive (universal) force of π on an input set U of substitutions is weakened so that not all substitutions in U need to execute π. Rather, we require that U be "$\mathcal{Q}x$-related" to *the* set $W \subseteq U$ of substitutions in U that can halt on π. The italics on "the" is meant to mark the "determinate"[1] character of the resulting discourse marker, although an "indeterminate" alternative will also be presented below. In either case, the condition $W \subseteq U$ builds in the conservativity of generalized quantifiers. Our interpretation of generalized quantifiers will be assembled from the programs $(x :=?; \pi)_\forall$ and $\mathcal{Q}x.\pi$, taking into account fine points concerning distributive-versus-collective and weak-versus-strong readings. Beyond these linguistic concerns, there are (inevitably?) various formal options that seem to carry no linguistic content, but which nevertheless demand our attention.

With these options as well as the question (Q2) of computability in mind, let us consider the passage (analyzed in Fernando [**9**]) from DPL transitions (defined relative to a fixed first-order model) to DRT transitions between discourse representation structures (DRS's). Observe that DRT transitions between DRS's not only

(S1) refer to transitions from various first-order models (in the same manner that first-order formulas apply to various first-order models),

but also

(S2) "determinize" the non-deterministic transitions in DPL (following the classic construction of deterministic finite automaton from non-deterministic finite automaton).

This suggests a two-step passage (spelled out in the concluding section below) from DPL transitions to DRT transitions that can drown out certain details of a DPL transition. The internalization in (S2) of the non-determinism of DPL transitions results in a somewhat more complex notion of state (corresponding to a DRS), built from "disjunctive" non-deterministic sets, as well as dual "conjunctive" parallel sets. Item (S1) points to a certain uniformity in DPL, upon which our answer to (Q2) will be founded. Our contention is that the notion of computation underlying DPL (and DRT) should not be construed in the *local* sense of ordinary (classical) recursion theory (a theory that applies

[1] The term "determinate" is intended to refer here to something akin to both defini*te* descriptions and *determin*istic programs. Its antonym "indeterminate" is meant to refer, in the same way, to "unselective" binding, as used in Van der Does [**6**] (assuming the present author understands Van der Does' intentions correctly).

specifically to *the* standard model of arithmetic; viz., the natural numbers), but in the *global* sense of Gödel's completeness theorem (as opposed to his incompleteness theorems). The terms "program" and "computation" are perhaps somewhat misleading when used in DPL; because the set Π of programs in DPL is closed under negation, there can be no program in Π universal for Π (contrary to the tradition of recursion theory). If, however, the term "program" is replaced by "formula" (and "computation" by "logic"), then this failure of universality is hardly surprising, in view of Tarski's theorem on the undefinability of truth. (Taking the natural numbers as a concrete example, negation leads to non-arithmeticity.) On the other hand, if one insists in using the words "program" and "computation", then — and this is our answer to (Q2) — DRT provides for DPL an appropriate *representational* level of abstraction for such discourse. Computation lives on syntax, and syntax is what DRT supplies in the form of a DRS. The additional computational power that a generalized quantifier $\mathcal{Q}$ might be expected to contribute should be understood at the global level, where uniformities in the map from a set M to the binary relation $\mathcal{Q}_M$ on Power(M) matter.

The remainder of the present paper is organized as follows. After some technical background on dynamic semantics in section 1, a bit of linguistic data is presented in section 2. Section 3 then considers the passage from S to $\text{Pow}_+(S)$, concentrating on propositional rather than quantificational matters. A closer look inside substitutions is needed in section 4 for the analysis (proper) of generalized quantifiers. Section 5 turns to some related work (viz., Van den Berg [**2, 4**] and Van der Does [**6**]), before section 6 concludes with a review of the essential points of the present work.

1. Between formulas and programs

Let Φ be a set of formulas φ, and Π be a set of programs π. Assume that the formulas and programs are interpreted relative to a fixed set S of states by functions $|\cdot|$ and ρ respectively such that for every $\varphi \in \Phi$, $|\varphi| \subseteq S$, while for every $\pi \in \Pi$, $\rho(\pi) \subseteq S \times S$, the intuition being that

$$|\varphi| \quad \approx \quad \{s \in S \mid \varphi \text{ is true at } s\}$$

and

$$\rho(\pi) \quad \approx \quad \{(s,t) \in S \times S \mid \text{given input state } s, \\ \pi \textit{ can} \text{ halt on output state } t\} \ .$$

Beyond the broad question of how formulas relate to programs, the present paper is concerned with what the word "can" (in the line above for $\rho(\pi)$) means in the presence of non-determinism.

Consider the embedding $\cdot ? : \Phi \to \Pi$ of formulas φ into programs (tests) $\varphi?$, given by $\rho(\varphi?) = \Delta|\varphi|$, where ΔU is the diagonal $\{(s,s) \mid s \in U\}$. Going the opposite direction, define the projection $\langle\cdot\rangle\top : \Pi \to \Phi$ of programs π into formulas $\langle\pi\rangle\top$, by $|\langle\pi\rangle\top| = dom\rho(\pi)$. A system of dynamic semantics such as

DPL amounts to a translation of formulas φ into programs π_φ, that in turn assert $\langle\pi_\varphi\rangle\top$. Now, certainly, for the case where S is the set of substitutions (mentioned in the introduction), the usual Tarskian definition can be obtained by setting

$$\begin{aligned} |\varphi\&\psi| &= |\langle\varphi?;\psi?\rangle\top| \\ |\sim\varphi| &= |\langle\neg(\varphi?)\rangle\top| \\ |\exists x\varphi| &= |\langle x:=?;\varphi?\rangle\top|\ , \end{aligned}$$

where

$$\begin{aligned} \rho(\pi;\pi') &= \rho(\pi)\circ\rho(\pi') \\ &(= \{(f,g)\in S\times S \mid (\exists h)\ f\ \rho(\pi)\ h \text{ and} \\ &\qquad\qquad h\ \rho(\pi')\ g\}) \\ \rho(\neg\pi) &= \{(f,f)\mid f\in S-dom\rho(\pi)\} \end{aligned}$$

and $\rho(x:=?)$ is as described in the introduction. It turns out, however, that the translation $\varphi\mapsto\varphi^{DPL}$ given by

$$\begin{aligned} R(\bar{t})^{DPL} &= R(\bar{t})? \\ (\varphi\&\psi)^{DPL} &= \varphi^{DPL}\ ;\ \psi^{DPL} \\ (\sim\varphi)^{DPL} &= \neg(\varphi^{DPL}) \\ (\exists x\varphi)^{DPL} &= x:=?\ ;\ \varphi^{DPL} \end{aligned}$$

has a different effect than the translation $\varphi\mapsto\varphi?$ in that the proposition $\langle\varphi^{DPL}\rangle\top$ (asserted by φ^{DPL}) is not equivalent (under ρ) to the proposition $\langle\varphi?\rangle\top$ (asserted by $\varphi?$). The reason, quite simply, is that although a formula φ is equivalent to $\langle\varphi?\rangle\top$, a program π may differ from $\langle\pi\rangle\top?$, the latter of which is equivalent to the static projection $\neg\neg\pi$ (interpreted as $\Delta dom\rho(\pi)$). The dynamic effect of $x:=?$ is useful for introducing a (singular) pronoun by an existential quantifier, as exemplified by the notorious donkey sentence

if a farmer owns a donkey then he beats it ,

which can be formalized naively as

$$(*) \qquad (\exists x,y)\ (\mathsf{f}(x)\ \&\ \mathsf{o}(x,y)\ \&\ \mathsf{d}(y)) \quad\supset\quad \mathsf{b}(x,y)\ .$$

DPL translates $(*)$, in turn, to the program

$$\neg\ (x:=?;y:=?;\mathsf{f}(x)?;\mathsf{o}(x,y)?;\mathsf{d}(y)?\ ;\ \neg\mathsf{b}(x,y)?)\ ,$$

following the equalities

$$\begin{aligned} (\varphi\supset\psi)^{DPL} &= (\sim(\varphi\&\sim\psi))^{DPL} \\ &= \neg(\varphi^{DPL}\ ;\ \neg\psi^{DPL})\ . \end{aligned}$$

The program scheme $\neg(\pi;\neg\pi')$ generated by the DPL translation of implication occurs again in the DPL translation of universal quantification

$$\begin{aligned} (\forall x\varphi)^{DPL} &= (\sim\exists x\sim\varphi)^{DPL} \\ &= \neg(x:=?\ ;\ \neg\varphi^{DPL})\ . \end{aligned}$$

The outer negation blocks the introduction of discourse markers (for plurals).[2] To overcome the static effect of negation $\neg$, a binary program construct $\Rightarrow$ is introduced in Fernando [**10**] that externalizes the dynamism within $\neg(\pi;\neg\pi')$ by collecting (in the output) witnesses g' to the $\forall\exists$-clause in

$$\begin{array}{lll} f\ \rho(\neg(\pi;\neg\pi'))\ g & \text{iff} & f = g \text{ and } (\forall f' \text{ s.t. } f\ \rho(\pi)\ f') \\ & & \qquad\qquad (\exists g')\ f'\ \rho(\pi')\ g'\ , \end{array}$$

the idea being that the "process f spawns" the processes g'. The first proposal in Fernando [**10**] is

(P1) to introduce functional states:

$$\begin{array}{lll} s\ [\![\pi \Rightarrow \pi']\!]_1\ F & \text{“iff”} & (s \notin dom[\![\pi]\!]_1 \text{ and } s = F) \text{ or} \\ & & (s \in dom[\![\pi]\!]_1 \text{ and } F \text{ is a} \\ & & \ \text{function with domain} \\ & & \ \{s' \mid s[\![\pi]\!]_1 s'\} \text{ s.t.} \\ & & \ (\forall s' \text{ s.t. } s[\![\pi]\!]_1 s') \\ & & \qquad s'[\![\pi']\!]_1 F(s'))\ . \end{array}$$

Implicit above is a passage from S to the least solution S_1 to $S_1 = S \cup (S_1 \rightharpoonup S_1)$, where $A \rightharpoonup B$ is the set of partial functions from A to B. A relation $R \subseteq S \times S$ can be lifted canonically to $\hat{R}$ by requiring that for functions $F \in S_1 - S$ (and for all $G \in S_1$),

$$\begin{array}{lll} F\ \hat{R}\ G & \text{iff} & dom(F) = dom(G) \text{ and} \\ & & (\forall s \in dom(F))\ F(s)\ \hat{R}\ G(s)\ . \end{array}$$

A simplification of (P1) is

(P1.5) to replace the function F by its image U

$$\begin{array}{lll} s\ [\![\pi \Rightarrow \pi']\!]_{1.5}\ U & \text{“iff”} & (s \notin dom[\![\pi]\!]_{1.5} \text{ and } s = U) \\ & & \text{or } (s \in dom[\![\pi]\!]_{1.5} \\ & & \ \text{and there is a function} \\ & & \ F : \{s' | s[\![\pi]\!]_{1.5} s'\} \rightarrow_{onto} U \\ & & \ \text{s.t. } (\forall s' \text{ s.t. } s[\![\pi]\!]_{1.5} s') \\ & & \qquad s'[\![\pi']\!]_{1.5} F(s'))\ . \end{array}$$

That is, pass from S to the least solution $S_{1.5}$ to $S_{1.5} = S \cup \mathrm{Pow}_+(S_{1.5})$. A further simiplification of (P1) is

[2] The reader familiar with dynamic logic can cast $(\forall x\varphi)^{DPL}$ as $[x :=?]\langle\varphi^{DPL}\rangle\top?$, and view the present problem as identifying a program counterpart $\Rightarrow$ to the box/square modalities, playing a role analogous to that of ; for the diamond/angle modalities

$$\begin{array}{lll} (\langle\pi\rangle\varphi)? & \equiv & \neg\neg(\pi\ ;\ \varphi?) \\ ([\pi]\varphi)? & \equiv & \neg\neg(\pi \Rightarrow \varphi?)\ , \end{array}$$

where $\equiv$ is semantic equivalence.

(P2) to project $S_{1.5}$ into $\mathrm{Pow}_+(S)$ by embedding S into $\mathrm{Pow}_+(S)$ via $f \mapsto \{f\}$ and collapsing $\mathrm{Pow}_+(\mathrm{Pow}_+(S))$, $\mathrm{Pow}_+(\mathrm{Pow}_+ (\mathrm{Pow}_+(S)))$, etc. to $\mathrm{Pow}_+(S)$ by suitable iterations of union. That is, project $S_{1.5}$ into $\mathrm{Pow}_+(S)$ by the map $\tilde{\cdot}$ defined by

$$\tilde{s} = \begin{cases} \{s\} & \text{if } s \in S \\ s & \text{if } s \in \mathrm{Pow}_+(S) \\ \bigcup\{\tilde{t} \mid t \in s\} & \text{otherwise.} \end{cases}$$

There are at least two complications facing the proposals above.

(i) First, the map $R \mapsto \hat{R}$ in (P1) does *not* commute with, for example, the interpretation $\cup$ of $+$,

$$(R \hat{\cup} R') \quad \text{need not equal} \quad \hat{R} \cup \hat{R}' \ .$$

Neither does it commute with $\neg$. Nor do matters improve when passing to (P1.5) or to (P2).

(ii) In (P1.5) or (P2), it is not clear how, proceeding from (P1), to collapse S_1 to $S_{1.5}$ or S_2, without the information (in the mapping F) about how processes are spawned. In place of the canonical map $R \mapsto \hat{R}$ of (P1), there are (as explained in section 3) at least two non-equivalent ways of lifting a binary relation on S to a binary relation on $\mathrm{Pow}_+(S)$, consistent with a conjunctive reading of the non-empty sets of substitutions.

As will be shown below, item (i) carries genuine linguistic content (pointing, as it does, to the difference between so-called collective and distributive readings). Whether or not item (ii) does is unclear to the present author.

2. Some linguistic data

Sentences such as the following suggest that singular pronouns tend to have indeterminate references, while plural pronous introduced by generalized quantifiers tend to be determinate.

(1) A/some congressman admires Kennedy. He is very junior.

(2) Some congressmen admire Kennedy. They are very junior.

"He" in (1) is some (indeterminate) congressman who admires Kennedy. "They" in (2) is, according to Evans [**8**], an E-type pronoun referring to all congressmen who admire Kennedy. Its determinateness suggests that a plural object must be definable (possibly in first-order logic). The truth conditions for (2) contrast with those for

(3) Some congressmen admire Kennedy and are very junior.

(3) has the same truth conditions as

(4) Some congressmen admire Kennedy. Some of them are very junior.

However, "they" would seem to have different available meanings in (3) and (4): follow up (4) with

(4b) Most are quite senior.

A somewhat minor point is whether a discourse marker should be kept.

(5) Some congressmen admire Kennedy. Two of them are very junior. Three are somewhat senior. Both are in their early thirties.

(4b) and (5) suggest that discourse markers should not be destroyed (but perhaps copied to some new variable).

In fact, the determinatenes of generalized quantifiers is somewhat controversial. It is felt by some that the line

two congressmen admire Kennedy

does *not* imply that at most two congressmen admire Kennedy. (Consider the analysis in Kamp and Reyle [**14**], p. 378 of "two of them" by some indeterminate V.) But then an E-type reading of

Two congressmen admire Kennedy. They are very junior.

makes "they" refer to all congressmen who admire Kennedy. Even if we concede that "two" is ambiguous between "exactly two" and "at least two", the creation of a discourse marker "they" suggests that there are exactly two such congressmen. Or is an indeterminate choice of such a pair from a model where there are more than two allowed? Both options will be explored below.

Then there is also the question of what it means for the group to admire Kennedy. Even if, simplifying matters, we agree that each one in the group must admire Kennedy, there is more trouble ahead. The sentence

(6) they are very junior or very silly

has the collective reading

all of them are very junior, or all of them are very silly

(where "or" has scope outside "they"), and the distributive reading

each one of them is either very junior or very silly

(where "or" has scope inside "they"). Similarly, turning to negation,

(7) they are not very junior

has the distributive reading

each one of them is not very junior ,

and the collective reading

at least one of them is not very junior .

There will be no avoiding such complications in our formalization.

Finally, let us turn to a point often made about the proposal in Fernando [**10**], on which the present paper builds. Although it is true enough that the line

Each student told a joke. She forgot the punchline.

(from Barwise [**1**]) sounds anomalous, suggesting that the discourse marker for a noun phrase that is universally quantified should not be available beyond the sentence in which it occurs, it is equally true that lines such as

Every player chooses a pawn. He puts it on square one.

(from Groenendijk and Stokhof [**11**]) suggest otherwise. The former suggestion is available in Fernando [**10**] via the usual classical reduction (adopted in DPL) of $\forall$ and $\supset$ in terms of negation, $\exists$ and conjunction. The latter suggestion is also within the scope of Fernando [**10**]. But alas, just how to choose between these two options is not. The present paper will be full of this sort of embarrassment of riches.

3. Second-order programs

The present section is concerned with a passage from relations on S to relations on $\mathrm{Pow}_+(S)$, pictured by

$$\frac{\rho(\pi) \subseteq S \times S}{[\![\pi]\!] \subseteq \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S)}$$

along the embedding $f \mapsto \{f\}$ of S into $\mathrm{Pow}_+(S)$, so that for the programs π from ordinary dynamic logic,

$$\{f\}[\![\pi]\!]\{g\} \quad \text{iff} \quad f\ \rho(\pi)\ g\ .$$

(Recall that the empty set $\emptyset$ is excluded as it arises from the failure of a substitution to make a ρ-transition.) There is some freedom in choosing exactly how to pass from ρ to $[\![\cdot]\!]$, even if we require that a set $U \in \mathrm{Pow}_+(S)$ of substitutions should be treated in a conjunctive (parallel) manner, dual to the disjunctive manner of non-determinism. A binary relation R on S can be lifted to a binary relation on $\mathrm{Pow}_+(S)$ as

$$\begin{aligned} \uparrow R \;=\; & \{(U,V) \in \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S) \mid \exists \text{ function } F \text{ from} \\ & \qquad U \text{ onto } V \text{ s.t. } (\forall s \in U)\ s\ R\ F(s)\} \end{aligned}$$

(which is implicit in the interpretation of ; in Peleg [**16**]) or else (non-equivalently) as

$$\begin{aligned} \Uparrow R \;=\; & \{(U,V) \in \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S) \mid \\ & (\forall s \in U)(\exists t \in V)\ sRt \text{ and } (\forall t \in V)(\exists s \in U)\ sRt\}\ . \end{aligned}$$

Comparing these with the the inductive clauses from dynamic logic, note that $(\uparrow R) \circ (\uparrow R')$ need not equal $\uparrow (R \circ R')$, as pointed out to the author by A. Visser. On the other hand, $(\Uparrow R) \circ (\Uparrow R')$ is $\Uparrow (R \circ R')$, at the cost of some looseness in the embedding $f \mapsto \{f\}$ (already for random assignments).

Unfortunately, neither $\uparrow$ nor $\Uparrow$ commutes with $\cup$, although it is easy enough to motivate the failure here by the ambiguity in sentence (6) from the previous section. Similar complications surround negation, as illustrated by (7).

For the sake of concreteness, let us agree that if π is a random assignment $x :=?$ or a test $\varphi?$, then $[\![\pi]\!] = \uparrow \rho(\pi)$. Then, proceeding inductively, define (for collective readings)

$$\begin{aligned} [\![\pi;\pi']\!] &= [\![\pi]\!] \circ [\![\pi']\!] \\ [\![\pi+\pi']\!] &= [\![\pi]\!] \cup [\![\pi']\!] \\ [\![\neg\pi]\!] &= \{(s,s) \mid s \in \mathrm{Pow}_+(S) - dom[\![\pi]\!]\} \ , \end{aligned}$$

further closing the programs (if the reader desires) under unary program constructs $\cdot_\uparrow$ and $\cdot_\Uparrow$, subject to

$$\begin{aligned} [\![\pi_\uparrow]\!] &= \{(U,V) \mid U \in \mathrm{Pow}_+(S) \text{ and there is a} \\ &\qquad \text{function } F \text{ with domain } U \text{ s.t. } V = \bigcup_{s\in U} F(s) \\ &\qquad \text{and } (\forall s \in U)\ \{s\}[\![\pi]\!]F(s)\} \\ [\![\pi_\Uparrow]\!] &= \{(U,V) \in \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S) \mid (\forall s \in U) \\ &\qquad (\exists W \subseteq V)\ \{s\}[\![\pi]\!]W \text{ and} \\ &\qquad (\forall t \in V)(\exists W \subseteq V)(\exists s \in U) \\ &\qquad \{s\}[\![\pi]\!]W \text{ and } t \in W\} \ . \end{aligned}$$

This allows a formulation of distributive alternatives $(\pi;\pi')_\uparrow$, $(\pi+\pi')_\uparrow$, $(\pi+\pi')_\Uparrow$, and $(\neg\pi)_\uparrow$ $(= (\neg\pi)_\Uparrow)$ to $\pi;\pi'$, $\pi+\pi'$, etc. Turning finally to the reason for passing from S to $\mathrm{Pow}_+(S)$, define

$$\begin{aligned} U[\![\pi \Rightarrow \pi']\!]V \quad \text{iff} \quad & (U \notin dom[\![\pi]\!] \text{ and } U = V) \text{ or} \\ & (U \in dom[\![\pi]\!] \text{ and there is a function } F \\ & \ \text{with domain } \{U' \mid U[\![\pi]\!]U'\} \text{ s.t.} \\ & \quad V = \bigcup\{F(U') \mid U' \in dom(F)\} \text{ and} \\ & \quad (\forall U' \in dom(F))\ U'[\![\pi']\!]F(U')) \ . \end{aligned}$$

Observe that

$$\begin{aligned} [\![\neg\neg(\pi \Rightarrow \pi')]\!] &= [\![\neg(\pi;\neg\pi')]\!] \\ [\![\neg\pi]\!] &= [\![\pi \Rightarrow \bot]\!] \end{aligned}$$

(where $[\![\bot]\!] = \emptyset$), although the equivalence between $\neg\pi$ and $([\pi]\bot)?$ is lost. A program counterpart $\pi_\forall$ to the box modality $[\pi]$ on formulas of dynamic logic that turns the usual "or" branching of Kripke frames into "and" branching,

$$U[\![\pi_\forall]\!]V \quad \text{iff} \quad V = \bigcup\{U' \mid U[\![\pi]\!]U'\} \ (\neq \emptyset) \ ,$$

can then be derived as

$$\pi_\forall \quad = \quad \neg\neg\pi \ ; \ (\pi \Rightarrow \top) \ ,$$

where $\top = \neg\bot$. Going the opposite direction,

$$[\![\pi \Rightarrow \pi']\!] \quad = \quad [\![(\pi_\forall ; \pi') + \neg\pi]\!]$$

provided π' is distributive in the sense of $\uparrow$ (i.e., $[\![\pi']\!] = [\![\pi'_\uparrow]\!]$) or π is deterministic. Although random assignments and tests are defined to be distributive, it is easy to manufacture a non-deterministic choice $\pi_1 + \pi_2$ (e.g., from tests) that is not.

Of the two ways of turning an arbitrary program into a deterministic program, $\neg\neg\cdot$ is static, whereas $\cdot_\forall$ is dynamic. Notice that if π is deterministic, then $[\![\pi]\!] = [\![\pi_\forall]\!]$. In particular, as $\neg\pi$ is always deterministic, whether or not π is, $[\![\neg\pi]\!] = [\![(\neg\pi)_\forall]\!]$. It is also easy to see that $[\![\neg\pi]\!] = [\![\neg(\pi_\forall)]\!]$. Recalling the construct $\cap$ of Peleg [**16**], note that if π and π' are deterministic, then $\pi \cap \pi'$ can be coded as $(\pi + \pi')_\forall$.

4. Generalized quantifiers

Given a generalized quantifier $\mathcal{Q}$ that maps the universe M of a first-order model to a binary relation $\mathcal{Q}_M$ on Power(M), Chierchia [**5**] (as well as Van Eijck and De Vries [**7**]) extend quantified dynamic logic by binary programs $\mathcal{Q}x(\pi, \pi')$ (for $x \in X$ and programs π and π'), interpreting it (by an extension of ρ) as the externally static test

$$\begin{aligned}\{(s,s) \quad | \quad & s \in S \text{ and } \{m \in M \mid \exists s' \; s[x/m] \; \rho(\pi) \; s'\} \; \mathcal{Q}_M \\ & \{m \in M \mid \exists s' \; s[x/m] \; \rho(\pi;\pi') \; s'\}\} \;.\end{aligned}$$

This corresponds to what Chierchia [**5**] calls the "weak" reading, which is presented as line (4.166) in p. 421 of Kamp and Reyle [**14**], for the semantics of duplex conditions. The alternative "strong" reading

$$\begin{aligned}\{(s,s) \quad | \quad & s \in S \text{ and } \{m \in M \mid \exists s' \; s[x/m] \; \rho(\pi) \; s'\} \; \mathcal{Q}_M \\ & \{m \in M \mid \exists s' \; s[x/m] \; \rho(\neg(\pi;\neg\pi')) \; s'\}\} \;.\end{aligned}$$

(corresponding to line (4.172), p. 424 of Kamp and Reyle [**14**]) is also offered in Chierchia [**5**] (and Van Eijck and De Vries [**7**]). In either case, a basic point of the present work is to externalize the dynamic effects so as to support the imposition of further constraints on variables introduced by generalized quantifiers. That is, our analysis of a generalized quantifier $\mathcal{Q}$ will incorporate the abstraction condition of Kamp and Reyle within our mechanism for their duplex condition. We will modify the notation $\mathcal{Q}x(\pi, \pi')$ slightly by writing $(\mathcal{Q}x : \pi)\pi'$ for our proposal.

Concentrating first on the strong reading, consider the interpretation

$$(\mathit{All}\ x : \pi)\pi' \quad = \quad (x :=?; \pi) \;\Rightarrow\; \pi'$$

implicit in Fernando [**10**]. This is semantically equivalent, if $[\![\pi']\!] = [\![\pi'_\uparrow]\!]$, to

$$(x :=?; \pi)_\forall \;;\; \pi' \quad + \quad \neg(x :=?; \pi) \;.$$

For the remainder of this section, let us not be distracted by the question of the distributivity of π', as this can always be arranged by replacing π' by

$\pi'_{\uparrow}$. (Similarly, throughout this section, the non-determinism suggested by $+$ will be spurious, as the disjuncts will be mutually exclusive.) Using certain programs that will be explained shortly, let us generalize the interpretation above of *All* to $\mathcal{Q}$ by equating $(\mathcal{Q}x : \pi)\pi'$ with

$$\begin{array}{ll}(x := ?; \pi)_{\forall} \ ; \ (\mathcal{Q}x.\pi' \ + \ \langle \mathcal{Q}^{-}x\rangle; \neg\pi') & + \\ & \neg(x := ?; \pi) \ ; \ \langle \mathcal{Q}^{0}x\rangle\end{array}$$

where $All\ x.\pi$ is (under $[\![\cdot]\!]$) just π, $\langle All^{-}x\rangle$ is $\perp$ (i.e., $[\![\langle All^{-}x\rangle]\!]$ is $\emptyset$), and $\langle All^{0}x\rangle$ is $\top$ (i.e., $[\![\langle All^{0}x\rangle]\!]$ is equality on $\mathrm{Pow}_{+}(S)$). More generally, the new programs $\mathcal{Q}x.\pi$ (for every program π), $\langle \mathcal{Q}^{-}x\rangle$, and $\langle \mathcal{Q}^{0}x\rangle$ are interpreted on the basis of the relation

$$\begin{array}{rcl}\mathcal{Q}_{M}x & = & \{(U,V) \in \mathrm{Pow}_{+}(S) \times \mathrm{Pow}_{+}(S) \ \mid \\ & & \quad \{s(x) \mid s \in U\} \ \mathcal{Q}_{M} \ \{s(x) \mid s \in V\}\}\end{array}$$

as follows

$$\begin{array}{rcl}[\![\mathcal{Q}x.\pi]\!] & = & \{(U,V) \in \mathrm{Pow}_{+}(S) \times \mathrm{Pow}_{+}(S) \mid \\ & & \quad U \ \mathcal{Q}_{M}x \ \bigcup\{W \subseteq U \mid W \in dom[\![\pi]\!]\} \ \text{ and} \\ & & \quad \bigcup\{W \subseteq U \mid W \in dom[\![\pi]\!]\} \ [\![\pi]\!] \ V\} \\ [\![\langle \mathcal{Q}^{-}x\rangle]\!] & = & \{(U,U) \mid U \in \mathrm{Pow}_{+}(M) \text{ and } U \ \mathcal{Q}_{M}x \ \emptyset\} \\ [\![\langle \mathcal{Q}^{0}x\rangle]\!] & = & \{(U,U) \mid U \in \mathrm{Pow}_{+}(M) \text{ and } \emptyset \ \mathcal{Q}_{M}x \ \emptyset\} \ .\end{array}$$

Conservativity is essentially built in, not only in the semantic clause for $\mathcal{Q}x.\pi$ (via the condition $W \subseteq U$), but also in the test $\langle \mathcal{Q}^{0}x\rangle$, which is an all-or nothing affair (being either $\top$ or $\perp$). The intuition behind $\mathcal{Q}x.\pi$ is that π is executed without necessarily insisting that every substitution in the input state U can halt on π, but rather that U is $\mathcal{Q}_{M}x$-related to the set of substitutions in U that can halt on π. The program $\mathcal{Q}x.\pi$ can be decomposed as the sequential composition

$$\mathcal{Q}x.\neg\neg\pi \quad ; \quad \pi$$

of a deterministic program $\mathcal{Q}x.\neg\neg\pi$ with π, since $[\![\mathcal{Q}x.\neg\neg\pi]\!]$ incorporates the first of $[\![\mathcal{Q}x.\pi]\!]$'s two conjuncts, plus a tiny bit more

$$\begin{array}{rcl}[\![\mathcal{Q}x.\neg\neg\pi]\!] & = & \{(U,V) \in \mathrm{Pow}_{+}(S) \times \mathrm{Pow}_{+}(S) \ \mid \\ & & \quad V = \bigcup\{W \subseteq U \mid W \in dom[\![\pi]\!]\} \ \text{ and} \\ & & \quad U \ \mathcal{Q}_{M}x \ V \ \text{ and } \ V \in dom[\![\pi]\!]\} \ .\end{array}$$

Although the unary program constructs $\mathcal{Q}x.\cdot$ and $\mathcal{Q}x.\neg\neg\cdot$ are, in the presence of sequential composition and negation, interdefinable (so that either could be taken as primitive), it is quite another matter (as we will see shortly) to reduce these unary constructs to a nullary construct $\langle \mathcal{Q}^{+}x\rangle$, independent of π.

The external dynamic effects of a program fragment π can always be suppressed by prefixing the appropriate program fragment by double negation $\neg\neg$.

Such effects are, of course, crucial in interpreting inter-sentential anaphora, which allows constraints on pronouns to be imposed by incremental updates. Double negation $\neg\neg$ is useful for encoding the weak reading of $(\mathcal{Q}x : \pi)\pi'$ as

$$\begin{aligned}(x :=?; \neg\neg\pi)_\forall \ ; \ (\mathcal{Q}x.(\pi;\pi') \ + \ \langle\mathcal{Q}^- x\rangle; \neg(\pi;\pi')) \quad & + \\ \neg(x :=?; \pi) \ ; \ \langle\mathcal{Q}^0 x\rangle \quad & .\end{aligned}$$

Because the dynamic effects of π are suppressed in $(x :=?; \neg\neg\pi)_\forall$, the program π must (in contrast to the strong reading) be repeated in $(\mathcal{Q}x.(\pi;\pi') + \langle\mathcal{Q}^- x\rangle; \neg(\pi;\pi'))$.

Next, let us turn to a deviant treatment of quantification, resulting from replacing $\mathcal{Q}x.\pi$ by the sequential composition

$$\langle\mathcal{Q}^+ x\rangle \ ; \ \pi$$

of a program $\langle\mathcal{Q}^+ x\rangle$ with π, where the program $\langle\mathcal{Q}^+ x\rangle$ chooses *some* non-empty subset V of the input set U to which U is $\mathcal{Q}_M x$-related

$$\begin{aligned}[\![\langle\mathcal{Q}^+ x\rangle]\!] \quad = \quad & \{(U,V) \in \mathrm{Pow}_+(S) \times \mathrm{Pow}_+(S) \mid \\ & U \mathcal{Q}_M x V \text{ and } V \subseteq U\}.\end{aligned}$$

The sequential composition with π then means that V must execute π.[3] Setting aside dynamic complications, to say that for all sets A and B,

$$(*) \qquad A\mathcal{Q}_M B \quad \text{iff} \quad (\exists C) \ \ A\mathcal{Q}_M C \ \text{ and } \ C \subseteq B$$

is to suppose that $\mathcal{Q}$ is upwards-monotone on its second argument (e.g. $\forall, \exists$). Even in that case, however, the discourse marker ("they") introduced by the quantification must refer (according to the E-type account) to $A \cap B$, and not to $A \cap C$, for some indeterminate choice of C. If $\mathcal{Q}$ is not upwards-monotone on its second argument, then $(*)$ fails even under a modification of $(*)$ where C is required to be maximal (as can be seen by taking, for example, $\mathcal{Q}$ to be "exactly two"). If the non-deterministic program $\langle\mathcal{Q}^+ x\rangle$ is ruled out, then it is plausible that the only sets of substitutions required (as inputs to and outputs of programs) are first-order definable, with parameters (for the "singular" variables).

5. Related work

As has already been pointed out in the introduction, the present work borrows from Peleg [**16**] the idea of a "conjunctive" set, but differs from it by introducing such sets not by $\cap$ but by $\Rightarrow$ (to capture a notion of parallelism dual to the non-determinism in $x :=?$, rather than in $+$). Conjunctive sets also appear in Van den Berg [**2, 4**], but again not via $\Rightarrow$ (or $\cdot_\forall$). Instead, a 3-valued logic is employed in Van den Berg [**2, 4**] to make a "plural logic" dynamic (although the present author cannot claim to understand exactly how this feat is carried out).

[3] A comparison with the indeterminate treatment of "two of them" in Kamp and Reyle [**14**], p. 378 is instructive here.

Next, let us say a few words about the system *Dynamic Quantifier Logic* (DQL) of Van der Does [**6**]. A basic premise of DQL is that singular pronouns involve "unselective binding"[4], while plurals are given by (E-type) descriptions (rather than some possibly indeterminate notion of maximality). Given a model with universe D, DQL interprets formulas relative to triples $(\mathbf{a}, \mathbf{c}, \mathbf{h})$ of

(i) *assignments* $\mathbf{a}$; i.e., functions from the set of variables to D (called substitutions above);

(ii) *contexts* $\mathbf{c}$; i.e., partial functions from the set of variables to formulas;

and

(iii) *choice functions* $\mathbf{h}$; i.e., functions from subsets of D to D (used for unselective binding on singulars).

(Assignments $\mathbf{a}$ are merged with choice functions $\mathbf{h}$ in Van der Does' later work, although this is a rather minor point.) The main idea, as Van der Does writes in his abstract for the Ninth Amsterdam Colloquium, is as follows.

> In my system, as in others, pronouns are quantifiers which depend for their interpretation on parametrized context sets. Plural pronouns are interpreted as a universal quantifier over such a context set, while singular pronouns involve a choice from such a set ...

Is he thus committed to a distributive reading of plurals? Whether or not he is, the general picture seems to be in accord with our determinate analysis above. A somewhat alarming difference, however, is that DQL "is best thought as a standard 'static' logic, which comes with a separate dynamic module to handle the context generated by a text" (Van der Does [**6**], p. 20); "DQL uncouples the interpretation of a formula from its context change potential" (p.25). The context change potentials associated with formulas are functions from contexts to contexts. If, however, transitions are given by *functional* context change potentials, what becomes of the non-determinism in DPL transitions? As the non-determinism in DPL arises (only) from the random assignments used for singular pronouns, presumably DQL has no trouble dealing with that via its choice functions. But alas, the present author must confess that it is not clear to him (at the time of writing) exactly how DQL relates to DPL, let alone to the extension of DPL proposed above.

6. Discussion

The extension to DPL proposed above is based on

(i) generating "conjunctive" sets U of substitutions through a program construct $\cdot_\forall$,

and

(ii) weakening the "universal" force of a program π on such a set U according to a generalized quantifier $\mathcal{Q}$ by forming the program $\mathcal{Q}x.\pi$.

[4]Roughly, indeterminateness; see footnote 1.

The reader familiar with chapter 4 of Kamp and Reyle [**14**] might regard $\cdot_\forall$ as a mechanism for abstraction, and $\mathcal{Q}x.\cdot$ as a tool for the duplex condition.[5] But, in fact, the constructs $\cdot_\forall$ and $\mathcal{Q}x.\cdot$ can be explained without mentioning DRT — and, in the case of $\cdot_\forall$, without mentioning variables —, proceeding instead from computational intuitions involving a form of duality between non-determinism and parallelism. The linguistic relevance of the present work rests on the degree to which the distinction between definite and indefinite noun phrases can be captured by that between deterministic and non-deterministic programs. By providing for conjunctive branching, $\cdot_\forall$ yields, from a dynamic viewpoint, more interesting deterministic programs than the static projection $\neg\neg\cdot$. Furthermore, it is a trivial matter to derive $\cdot_\forall$ from a binary program construct $\Rightarrow$ introduced in Fernando [**10**] to extend the DPL translation.

As earnestly as the present author has tried to motivate his proposals, the reader sensitive to "ad hoc-kery" might worry all the same about some five branch-points that were faced along the way:

(B1) the choice in section 1 (of an extension to higher types) between proposals (P1), (P1.5) and (P2)
(B2) within (P1.5) and (P2), the choice (of a lifting from relations on S to relations on $\mathrm{Pow}_+(S)$) between $\uparrow\cdot$ and $\Uparrow\cdot$ (in section 3)
(B3) distributive versus collective readings
(B4) weak versus strong readings

and

(B5) the determinateness or indeterminateness of discourse markers introduced by generalized quantifiers.

Of (B1) to (B5), the last three obviously have, for a linguist, some substance. As for the first two points, however, the question is *who cares*? Whether these options matter can be investigated (by formulating items (S1) and (S2) in the introduction) using the somewhat simple-minded notion of a *(pointed)*

[5]More precisely, the present author claims that if $\mathcal{K}$ is the set of DRS's given up to chapter 4 of Kamp and Reyle [**14**], and if Π is the set of programs from above, then a mapping $K \mapsto \pi_K$ can be defined from $\mathcal{K}$ to Π, with the property that

$$\pi_K \text{ sends the empty DRS to the DRS } K.$$

Briefly, the main point of the inductive definition of the mapping would be to encode

(i) abstraction $X = \sum x : K$ by $(x :=?; \pi_K)_\forall \; + \; \neg(x :=? \; ; \; \pi_K)$,

and

(ii) the duplex condition with restrictor K, quantifier $\mathcal{Q}$, principal discourse referent x and scope K' by $\neg\neg(\mathcal{Q}x : \pi_K)\pi_{K'}$, for some binary program construct $(\mathcal{Q}x : \cdot)\cdot$ defined as in section 4 using $\cdot_\forall$ and $\mathcal{Q}x.\cdot$.

As indicated in the introduction, $(x :=?; \pi_K)_\forall$ preserves the functional dependencies of x in K. Furthermore, inasmuch as $(\mathcal{Q}x : \cdot)\cdot$ already incorporates the effect of abstraction, it can be argued that $(\mathcal{Q}x : \cdot)\cdot$ is *not* subject to the criticism made in Van den Berg [**3**] "that the dynamic behavior of the quantifier is totally **ad hoc**. The definition gives a condition in terms of the generalized quantifier and on top of that it gives some condition on the output ..." (On the other hand, this very decomposition points to the conjecture that all sets of substitutions needed are first-order definable.)

transition system, a 4-tuple $\langle S, L, \rightarrow, s_0 \rangle$, where S and L are non-empty sets, whose elements are called *states* and *labels* respectively, $\rightarrow \subseteq S \times L \times S$ is a *transition predicate*, and $s_0 \in S$ is an *initial state*. The semantics of DPL can be characterized, relative to a first-order model $\mathbf{M}$ and an infinite set X of variables, by a transition system $[\![\mathbf{M}]\!] = \langle X \rightarrow_{fin} M, \Phi, [\![\cdot]\!]_{\mathbf{M}}, \emptyset \rangle$, where

(i) the state set $X \rightarrow_{fin} M$ is the set of $\mathbf{M}$-substitutions; i.e., functions from finite subsets of X to the universe M of $\mathbf{M}$,
(ii) the label set Φ is the set of first-order formulas (with the signature of $\mathbf{M}$) over the set X of variables,
(iii) the transition predicate $[\![\cdot]\!]_{\mathbf{M}}$ associates with every $\varphi \in \Phi$ the binary relation $[\![\varphi]\!]_{\mathbf{M}}$ on $X \rightarrow_{fin} M$, given by $\rho(\varphi^{DPL})$ (modified in the obvious way for finite functions), and
(iv) the initial state $\emptyset$ is the totally undefined function.

Similarly, the various options in (B1) and (B2) yield transition systems, albeit somewhat more complicated ones. A transition system for Kamp [**15**] can also be constructed, independent of some first-order model $\mathbf{M}$, by taking DRS's as its states, among which the empty DRS is initial; call that transition system $\mathcal{DRT}$. Notice that whereas transitions in $\mathcal{DRT}$ are deterministic, those in $[\![\mathbf{M}]\!]$ are not. But it is easy enough to make the transitions in $[\![\mathbf{M}]\!]$ deterministic by passing to the transition system $\mathcal{D}([\![\mathbf{M}]\!])$, given by the map $\mathcal{D}$ on transition systems that sweeps the non-determinism underneath states (formed as "disjunctive" sets, dual to the conjunctive sets above). (As mentioned in item (S2) of the introduction, $\mathcal{D}$ is essentially the classic construction of deterministic finite automata from non-deterministic ones.) It turns out (as shown in Fernando [**9**]) that the transition system $\mathcal{DRT}$ is isomorphic to the obvious transition system sum

$$\sum \{\mathcal{D}([\![\mathbf{M}]\!]) \mid \mathbf{M} \in \mathcal{M}\}$$

over the class $\mathcal{M}$ of (say countable) first-order models $\mathbf{M}$ (of the appropriate signature). (The formation of the sum above captures the uniformity mentioned in item (S1) of the introduction.) That is, as states go, DRS's embody a good deal of partiality and complexity, hiding ghastly details from the transition predicate. The transformations $\mathcal{D}$ and $\sum$ together bridge the gap between DPL and DRT (mentioned as items (S1) and (S2) in the introduction) that can be depicted as follows.

	DPL	**DRT**
states	substitution	DRS
transitions	non-deterministic	deterministic
	semantic	syntactic
	local	global
labels	formula	program

Now returning to the ad hoc-kery in (B1) and (B2) above, the point is to compare the transition systems they induce modulo $\mathcal{D}$ and $\sum$. Much work remains to be done.

Acknowledgments

My thanks to colleagues in Amsterdam (Martin van den Berg, Jan van Eijck), Utrecht (Kees Vermeulen, Albert Visser), Stuttgart (Ede Zimmermann) and Edinburgh (David Milward, Massimo Poesio) for comments on oral and/or written presentations of this work.

References

[1] Jon Barwise. Noun phrases, generalized quantifiers and anaphora. In E. Engdahl and P. Gärdenfors, editors, *Generalized Quantifiers*, Studies in Language and Philosophy. Dordrecht: Reidel, 1987.

[2] Martin van den Berg. Dynamic generalized quantifiers. In J.M. van der Does and J. van Eijck, editors, *Generalized Quantifier Theory and Applications.* Dutch Network for Language, Logic and Information Prepublication, Amsterdam, 1991. Revised version to appear as CSLI Lecture Notes.

[3] Martin van den Berg. A direct definition of generalized dynamic quantifiers. Preliminary abstract submitted to the Ninth Amsterdam Colloquium, 1993.

[4] Martin van den Berg. Full dynamic plural logic. In K. Bimbó and A. Máté, editors, *Proc. 4th Symposium on Logic and Language.* Budapest, 1993.

[5] Gennaro Chierchia. Anaphora and dynamic logic. ITLI Prepublication, University of Amsterdam, 1990.

[6] Jaap M. van der Does. The dynamics of sophisticated laziness. Manuscript, ILLC, Amsterdam, September 1993.

[7] J. van Eijck and F.J. de Vries. Dynamic interpretation and Hoare deduction. *J. Logic, Language and Information*, 1, 1992.

[8] Gareth Evans. Pronouns. *Linguistic Inquiry*, 11(2), 1980.

[9] Tim Fernando. Transition systems and dynamic semantics. In D. Pearce and G. Wagner, editors, *Logics in AI*, LNCS 633 (subseries LNAI). Springer-Verlag, Berlin, 1992. A slightly corrected version has appeared as CWI Report CS-R9217, June 1992.

[10] Tim Fernando. The donkey strikes back. In *Proc. of the 6th Conference of the European Chapter of the Association for Computational Linguistics.* OTS, Utrecht, The Netherlands, 1993.

[11] J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14, 1991.

[12] David Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2. D. Reidel, 1984.

[13] Irene Heim. The semantics of definite and indefinite noun phrases. Dissertation, University of Massachusetts, Amherst, 1982.

[14] H. Kamp and U. Reyle. *From Discourse to Logic.* Kluwer Academic Publishers (Dordrecht), 1993.

[15] Hans Kamp. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof, editors, *Formal Methods in the Study of Language.* Mathematical Centre Tracts 135, Amsterdam, 1981.

[16] David Peleg. Concurrent dynamic logic. *J. Assoc. Computing Machinery*, 34(2), 1987.

2

On the computational significance of consistency: from ordinary to generalized recursion theory

June 30, 1994[1]

ABSTRACT. The logical character of a transition-based notion of computation is explored in relation to Turing's ordinal logics and certain subsequent developments. The addition of Π^0_1 consistency statements in ordinal logics is portrayed as an attempt to extract fast programs computing a theory's provably recursive functions, and is noted to require (for the attempt to succeed) ordinal notations coding computations extraneous to the ordinal named. This motivates a richer semantic conception of computation based on transitions that include input/output transitions as special cases. In this regard, the equivalence underlying the construction of the next admissible set from trees is linked with the notion of a bisimulation, subsuming both input/output equivalence of programs and equivalence between ordinal notations naming the same ordinals. Complications introduced by non-well-foundedness (reflecting, for instance, non-termination of computational systems) are investigated, including $\Sigma^1_1 - \Pi^1_1$ pre-quotients of r.e. transition predicates. Π^1_1 paths through the Church-Kleene system $\mathcal{O}$ of ordinal notations are employed to analyze this leap from syntactic Π^0_1 consistency statements to a semantic Σ^1_1 notion of a computational process.

The starting point of various generalizations of ordinary recursion theory is the statement

$$(1) \qquad \{e\}(x) \ \simeq \ y$$

asserting that the program named e returns output y, given input x. In view of Church's thesis, however, it is not clear what relevance a non-r.e. set of input/output triples e, x, y, can have for mechanical computation. Accordingly, the object of generalization in the present paper is not so much the computational power behind (1), but rather the form of (1) itself, which is

[1] *Added, February 11, 1997.* This paper was the basis of a talk at a conference on Proof Theory, Provability Logic and Computation in Berne (March 1994), and was submitted for refereeing for a special issue of the *Annals of Pure and Applied Logic* consisting of selected papers from the conference. The two referees reviewing the paper recommended publication on the condition that it be "more or less completely" revised. The style in which it was written was deemed unacceptable for a mathematical journal. But as the thrust of the work was largely philosophical and programmatic, I feared the paper would become pointless if I were to carry out the referees' advice to "concentrate on the mathematical contents." Consequently, I withdrew the paper, hoping to return to it at some later point. Some two and a half years later, I have been unable to carry out a single revision and am as uncertain as ever of attracting for it the attention of a sufficiently indulgent reader. But as I still believe the point of the paper is worth making, I include it here for the record.

generalized into a possibly non-deterministic and non-terminal "transition," collectively forming a set that is r.e. under some syntactic presentation. An abstract semantic analysis of that syntax then takes us beyond the realm of the r.e., and through both syntactic and semantic notions of consistency.

More concretely, consider transitions of the form $(s, d) \to_e (s', d')$ where the transition predicate $\to_e$ relates pairs (s, d) and (s', d') of "control states" s, s' and "data configurations" d, d'. The input/output relation (1) then assumes the special form $(s_0, x) \to_e (\surd, y)$, where s_0 and $\surd$ are fixed initial and terminal states, respectively. Not all transitions need end at some $(\surd, y)$, however, which is to say that reducing a program to the function it computes is inadequate, if intermediate states (s', d') (with s' not terminal, nor initial) in the computation of that function are also of interest. To say that a program is its name (or text, Gödel numbered) e may ultimately be the safest answer to the question *what is a program?* A more "semantic" characterization would, on the other hand, abstract away syntactic details and provide an answer on the basis simply of the transitions the program can make. That, at least, is what the present paper investigates, employing the notion of a bisimulation (Park [**19**]) to obtain an equivalence on states based solely on transitions between them, and eventually pointing to a set-theoretic notion of computation. Towards that end, the transition $(s, d) \to_e (s', d')$ is recast as $s \overset{d,d'}{\to}_e s'$, and then further to

$$e \overset{d,d'}{\leadsto} e' , \tag{2}$$

where a program $\hat{e}$ is identified with the initial state of the transition predicate $\to_{\hat{e}}$, which is, in turn, embedded in some monster transition predicate $\leadsto$. Shifting the basic picture from input/output statements (1) to transitions (2) is, it is argued, far from a triviality, considering the partiality of computation.

Apart from some faint hope in gaining a "semantic" understanding, the move from (1) to (2) can be motivated by the "no-nonsense" problem of isolating programs that run quickly — or more precisely, the problem of

($*$) extracting fast programs computing a theory's provably recursive functions.

A "path" to generalized recursion theory is paved by an explosion in logical complexity from what can be viewed as an early attempt (i.e., Turing [**21**]) to solve ($*$) — an attempt that, upon reflection, leads to a far richer (set-theoretic) notion of computation. Briefly, the point is that the notions of equivalence on (i) ordinal notations from Turing [**21**], (ii) hyperprojective trees in the Barwise, Gandy and Moschovakis (1971) construction of the next admissible set,[2] and (iii) accessible pointed graphs in Aczel [**1**]'s relative consistency argument for the Anti-Foundation Axiom are all bisimilarity (i.e., the largest bisimulation). Thus, the well-known table

[2]The presentation of the next admissible in Barwise [**4**] is different, appealing instead to the theory of constructible sets.

ordinary recursion th	generalized recursion th
number ($\in \omega$)	set ($\in L_\alpha$)
Π^0_1	co-α-r.e

extends to

ordinary recursion th	generalized recursion th
syntax (code e)	semantics (set of transitions)
speed-up	substitutivity
consistency	bisimilarity

complementing the move from (1) to (2). The move suggests a step beyond not only equational reasoning, as advocated (for example) in Baeten and Weijland [**2**], but also first-order logic and, further than that, ω-logic (styled respectively as Σ^0_1- and Π^1_1-logic in Girard [**11**]). Model-theoretic arguments concerning bisimilarity that involve (respectively) saturation (a tool of first-order logic) and omitting types (a property of ω-logic) illustrate the point. The basic fact is that the Π^1_1 bound on bisimilarity relative to well-founded Δ^1_1 transition predicates (implicit in Barwise, Gandy and Moschovakis [**3**]) is exceeded by bisimilarity relative to a non-well-founded logspace computable predicate.

Of course, the idea of applying generalized recursion theory to programming languages and processes is far from new. Consider, for example, recent papers of Moschovakis (e.g., Moschovakis [**18**]). The "path" described in the present paper, however, goes back further to ancient work of Alan Turing's (Turing [**21**]), analyzed in relation to more modern developments in Feferman [**6, 7**], on which, in turn, a good deal of the present work builds (motivating the concept of relative computability by analyses based on different transition predicates).

1. Background: between absoluteness and open ended-ness

In a conference held in 1946, Gödel described recursion theory as being singularly successful "in giving an absolute definition of an interesting espistemological notion, i.e., one not depending on the formalism chosen" (Gödel [**12**], p. 150). Whereas the recursion-theoretic notion of computation is, according to Church's thesis, stable (capturing, as is claimed, all functions that can be computed mechanically), formal notions of deduction are defined "relative" to a formal theory that is (by contrast) subject to (Gödel) incompleteness. Curiously, the same functional reduction of computation in Church's thesis has been employed as the basis for a proof-theoretic approach to computation, which, as will be argued shortly, can only imperfectly approximate the "absolute" notion of a program.

Given an input/output specification $S(x, y)$, it is possible, under suitable circumstances, to extract from a constructive proof of

$$(\forall \text{ input } x)\ \ (\exists \text{ output } y) \qquad S(x, y)$$

a program e *realizing* (Kleene [**14**]) the theorem, where

$$e \text{ realizes } (\forall x)(\exists y)S(x,y) \quad \text{implies}$$
$$(\forall x)\ \{e\}(x)\!\downarrow \,\&\, S(x,\{e\}(x)) \ . \tag{3}$$

The correspondence between proofs and programs is developed further in the *Curry-Howard isomorphism* (e.g., Girard, Lafont and Taylor [**10**]) between a typed λ-calculus and a natural deduction system, under which computation becomes normalization (roughly, the unwinding of lemmas in a proof). Underlying the isomorphism is a theorem stating

$$\begin{aligned} \text{(weak) normalization} \quad : \quad & (\forall \text{ term } t) \ \ (\exists \text{ reduction } r) \\ & r \text{ takes } t \text{ to a normal form } . \end{aligned} \tag{4}$$

Now, although deduction may not be "absolute", the proofs of (3) and (4) can be formalized, which is to say that every program e provably extracted in some theory T is provably total in T —

$$T \ \vdash \ \mathsf{Tot}(e) \ ,$$

where $\mathsf{Tot}(e)$ is $(\forall x)(\exists y)\mathrm{T}_1(e,x,y)$ and T_1 is (the appropriate form of) Kleene's computation predicate.

1.1. Provably recursive functions versus provably recursive programs. The family of programs e that are *provably total* may look nice functionally, but is flawed upon closer inspection. More precisely, if the *provably recursive functions of T* are defined (following Kreisel [**15**]) as members of the set

$$\mathrm{Fcn}(T) \ := \ \{\{e\} \mid T \vdash \mathsf{Tot}(e)\} \ ,$$

then

Proposition 1. *If T contains primitive recursive arithmetic then $\mathrm{Fcn}(T)$ forms a (time-)complexity class — i.e.,*

$$T \vdash \mathsf{Tot}(e) \text{ and } t_{e'} < t_e \text{ a.e. } \ \text{imply} \ \ \{e'\} \in \mathrm{Fcn}(T) \tag{5}$$

where t_e counts the number of steps in the computation given by the Kleene T_1-predicate.

On the other hand,

Proposition 2. *The conclusion of (5) can be strengthened to $T \vdash \mathsf{Tot}(e')$ (with or without the assumption $\{e\} = \{e'\}$) if and only if T proves all true Π^0_1 sentences.*

Proposition 1 and the sufficiency in Proposition 2 follow from $t_e(x)$ being primitive recursively related to $(\mu y)\mathrm{T}_1(e,x,y)$, making the clause '$t_{e'} < t_e$ a.e.' Π^0_1. As for the necessity in Proposition 2, given a true Π^0_1 sentence

$(\forall x)R(x)$ (where R is a primitive recursive predicate), define

$$\begin{array}{lcl} e & : & \text{on input } n\text{, check } (\forall m \leq n+1)\ R(m); \\ & & \text{return } 0 \text{ (in any case).} \end{array}$$

$$\begin{array}{lcl} e' & : & \text{on input } n\text{, check } R(n); \\ & & \text{if so return } 0\text{, else loop.} \end{array}$$

Then $t_{e'} < t_e$, $\{e'\} = \{e\}$, $T \vdash \mathsf{Tot}(e)$, and if $T \vdash \mathsf{Tot}(e')$, then $T \vdash (\forall x)R(x)$. A similar argument occurs in Turing's completeness theorem below, but first, we pause to ask if such pathological programs can be avoided. Blum's speed-up theorem stating that some functions have no optimal programs would seem to suggest not, as discussed at further length in Fernando [**8**] (where the propositions above are drawn from, and where normalization for a calculus under a Curry-Howard isomorphism is cited as an example of such a function). To round off the relationship between Π^0_1 sentences and programs of bounded complexity, Kreisel [**15**] has shown that

$$\mathrm{Fcn}(T) \quad = \quad \mathrm{Fcn}(T + \{\textit{all true } \Pi^0_1 \textit{ sentences}\}) \ .$$

That is, overcoming Π^0_1-incompleteness is equivalent to extracting efficient programs, rather than new, more complex functions. (Normalization implies consistency but not conversely.) The restriction to programs of bounded complexity reduces the Π^0_2-completeness of totality (and input/output equivalence) to Π^0_1. A roller-coaster ride through logical complexity will, however, soon take us far beyond Π^0_2.

1.2. Turing through programs of bounded complexity. The remainder of this section draws considerably from Feferman [**6**], which describes Turing's attack on the incompleteness of T by repeatedly adding (for instance) consistency statements transfinitely. To carry this out, the theories must be *presented* and (in particular) the ordinals $0, 1, \ldots, \omega, \omega+1, \ldots$ must be *named*. Such a system of ordinal notations can be built, relative to a standard enumeration of the partial recursive functions $\{e\}$, by defining an r.e. binary relation $\prec$ on ω by

$$\frac{a > 0}{a \prec 2^a} \qquad\qquad \frac{\{e\}(n) \text{ is defined and } > 0}{\{e\}(n) \prec 3 \cdot 5^e}$$

$$\frac{a \prec b \qquad b \prec c}{a \prec c} \ ,$$

and assigning ordinal *heights*

$$|a| \quad = \quad \{|b| \mid b \prec a\}$$

whenever $\prec$ is well-founded on $\{b \mid b \prec a\}$.[3] In particular, $|\cdot|$ is defined on the set $\mathcal{O} \subset \omega$ of *Church-Kleene ordinal notations* generated by

$$\frac{}{1 \in \mathcal{O}} \qquad \frac{a \in \mathcal{O}}{2^a \in \mathcal{O}}$$

$$\frac{\{e\}(n) \in \mathcal{O} \text{ and } \{e\}(n) \prec \{e\}(n+1) \text{ for all } n}{3 \cdot 5^e \in \mathcal{O}} \; .$$

The *recursive ordinals* are the ordinals $|a|$ *named* by $a \in \mathcal{O}$, and collectively form the least non-recursive ordinal ω_1^{CK}. The progression $\{T_a\}_{a \in \mathcal{O}}$ of theories

$$\begin{aligned} T_1 &= T \\ T_{2^a} &= T_a + \mathsf{Con}[T_a] \\ T_{3 \cdot 5^e} &= \bigcup_{n<\omega} T_{\{e\}(n)} \end{aligned}$$

then constitutes an ordinal logic.

An ordinal logic provides, in Turing's view, a very rough and admittedly oversimplified model of mathematical reasoning (see Feferman [**6**], p. 129). A proof of a mathematical assertion A consists of an "intuitive" step (of complexity Π^1_1) in which an ordinal notation $a \in \mathcal{O}$ is chosen, and a "mechanical" step (of complexity Σ^0_1) establishing that $T_a \vdash A$. It turns out (perhaps not too surprisingly) that the burden of proving a true Π^0_1 sentence can be shoved over to the choice of a.

Theorem (Turing). *For every true Π^0_1 sentence A, there is an $a \in \mathcal{O}$ naming ω such that*

$$T_a + \mathsf{Con}[T_a] \quad \vdash \quad A \; .$$

Proof sketch. For A equal to $(\forall x)R(x)$ with R primitive recursive, take $a = 3 \cdot 5^e$ where e satisfies (by the second recursion theorem)

$$\{e\}(n) \; = \; \begin{cases} n_{\mathcal{O}} & \text{if } (\forall x < n) \; R(x) \\ Suc_{\prec}(3 \cdot 5^e) & \text{otherwise} \end{cases}$$

with $0_{\mathcal{O}} = 1$, $(n+1)_{\mathcal{O}} = 2^{n_{\mathcal{O}}}$, and $Suc_{\prec}(a) = 2^a$. $\dashv$

Commenting on the proof of the theorem, Turing [**21**] had this to say.

> The completeness theorem as usual is of no value. Although it shows, for instance, that it is possible to prove Fermat's last theorem ... (if it is true), yet the truth of the theorem would really be assumed by taking a certain formula as an ordinal formula [ordinal notation].

[3] The assumption of well-foundedness can be dropped so long as $|a|$ is allowed to be a non-well-founded set, as in the set theory of Aczel [**1**]; see subsection 2.2 below.

With an eye on a more general notion of ordinal logics T_a not necessarily based on the addition of consistency statements, Turing had further established the incompatibility of completeness with *invariance* under the choice of an ordinal notation: the property that whenever $|a| = |a'|$, $T_a \equiv T_{a'}$. Referring the interested reader to Feferman [**6**] for details on Turing's work, let us turn instead to a fundamental negative result concerning ordinal logics obtained after Turing, two ideas behind which will be of use to us later, namely

> Σ^1_1 *boundedness*: the height of every Σ^1_1 well-order is a recursive ordinal.
>
> *Non-standard ordinal notations*: a Σ^1_1 set $\mathcal{O}^* \supset \mathcal{O}$ indistinguishable from $\mathcal{O}$ up to Δ^1_1 sets can be defined by requiring that for such $a \in \mathcal{O}^*$, there are no Δ^1_1 $\prec$-descending sequences from a. More precisely, $\mathcal{O}^*$ is the intersection of all Δ^1_1 sets X of natural numbers that contain (i) 1, (ii) 2^a whenever $a \in X$, and (iii) $3 \cdot 5^e$ whenever $(\forall n)\ \{e\}(n) \in X$ & $\{e\}(n) \prec \{e\}(n+1)$.

Defining a *path* Z *through* $\mathcal{O}$ as a subset Z of $\mathcal{O}$ linearly ordered by $\prec$ such that for every $a \in \mathcal{O}$ there is a unique $b \in Z$ with $|a| = |b|$, the negative result is

Theorem (Feferman and Spector). *Π^1_1 paths through $\mathcal{O}$ exist, and for all such paths Z, there is a true Π^0_1 sentence not provable in $\bigcup_{a \in Z} T_a$.*

It is not altogether surprising then that Turing should be dissatisfied with ordinal logics as a practical method towards proving theorems (Feferman [**6**], p. 140). As an investigation into computation, rather than deduction, however, it would seem fair to say the following. Although Turing is remembered now mainly for "low level" programming (i.e., Turing machines, and perhaps even more importantly, cryptanalysis hacking), at least two somewhat abstract lines of research can be extracted from the work above that he initiated. First, one might carry on with an ordinal classification of programs of bounded complexity, paying careful attention to the notations used, and perhaps (in view of incompleteness along Π^1_1 paths through $\mathcal{O}$) abandoning any hope of assigning all such programs a recursive ordinal. Second, as to his completeness proof being "of no value", one might argue that, on the contrary, *by loading computations into an ordinal notation, he, in fact, anticipated a richer notion of computation, of not inconsiderable interest to current theoretical investigations.* That notion of computation is essentially the "intuitive" component of his ordinal logics that, moreover, by resisting elimination, led (already in Turing [**21**]) to the concept of *relative computability*. Pursuing this line of investigation further, the present paper is intended as a contribution to the argument made in Feferman [**7**] that relative computability is relevant to ordinary computation, in view of the myriad levels of abstraction relative to which a computation can be analyzed.

2. From ordinal notations to processes: bisimulations

The ordinal notations isolated in Turing's completeness proof constitute early examples of computational processes with the following "open-ended" features: intermediate computational states, non-provable termination, and countable non-determinism. These features are presently fleshed out.

2.1. Programs as states in a transition system. Without spelling out precisely what a transition is, it would appear plausible from Church's thesis to stipulate that the transitions undergone by a machine form, under a suitable syntactic presentation, an r.e. set. That is, to say that a transition predicate is mechanical is to assert that its transitions t can be named as theorems of some formal system. Indeed, an automaton (= program) e is typically characterized by its legal moves $i \vdash i'$ between so-called "instantaneous descriptions" i and i'. An instantaneous description i consists of a control state s and a data (e.g., tape) component d (plus an indication of the tape symbol currently being scanned, which we will assume is made implicit in the presentation of d). Roughly, what we will do is rewrite $(s_0, d) \rightarrow_e (s', d')$ as $(e, d) \rightarrow (e', d')$ where e' is the program described by $\rightarrow_e$ with initial state s', and $\rightarrow$ contains a copy of $\rightarrow_e$. Replacing e by p, the picture

$$(p, d) \quad \rightarrow \quad (p', d')$$

appears explicitly in so-called *mixed computation* (Bjørner, Ershov and Jones [**5**], p. xxvi), which we will recast as

$$p \quad \overset{d,d'}{\rightsquigarrow} \quad p' \tag{6}$$

to set the stage for an analysis of programs p relative to data d. That is, packaging the pairs (d, d') as *labels* l forming a set L, and passing from a program p back to the somewhat more abstract notion of a *state* s populating some set S, (6) leads to the notion of a *(labelled) transition system*, a triple $(L, S, \rightarrow)$ where $\rightarrow \subseteq S \times L \times S$. By presenting programs as states (following so-called *structural operational semantics*, Plotkin [**20**]), we can use a notion of equivalence on states to get a notion of equivalence on programs. But first, returning to the transition predicate $\rightsquigarrow \subseteq \omega \times \omega \times \omega$ occuring in (6), observe

Proposition 3. *There is an r.e. predicate $\rightsquigarrow \subset \omega \times \omega \times \omega$ that contains a copy of every r.e. transition predicate in that for every r.e. $\rightarrow \subseteq \omega \times \omega \times \omega$, there is a function f (that is, in fact, 1-1 and recursive) with domain equal to the set $\{s \mid (\exists l, t)\ s \overset{l}{\rightarrow} t \text{ or } t \overset{l}{\rightarrow} s\}$ of $\rightarrow$-states such that (i) for all s, l, t,*

$$s \overset{l}{\rightarrow} t \quad \textit{iff} \quad f(s) \overset{l}{\rightsquigarrow} f(t) \ ,$$

and (ii) for every u,

$$f(s) \overset{l}{\rightsquigarrow} u \ \text{ or } \ u \overset{l}{\rightsquigarrow} f(s) \quad \textit{implies} \quad (\exists t \in \mathit{dom}(f))\ f(t) = u \ .$$

Proof. Fix a binary primitive recursive function $\langle\cdot,\cdot\rangle : \omega \times \omega \to \omega$ and unary primitive recursive functions $i, j : \omega \to \omega$ such that for all $n, m < \omega$,

$$i(\langle n,m\rangle) = n \ , \quad j(\langle n,m\rangle) = m \quad \text{and} \quad \langle i(n), j(n)\rangle = n \ .$$

Let U be an r.e. set such that $\{U_e\}_{e<\omega}$ enumerates all r.e. sets, where

$$U_e \quad = \quad \{n \mid \langle e,n\rangle \in U\} \ ,$$

and define $\leadsto$ to be

$$\{(s,l,s') \in \omega \times \omega \times \omega \quad | \quad i(s) = i(s') \text{ and } \langle j(s), \langle l, j(s')\rangle\rangle \in U_{i(s)}\} \ .$$

Note that $\leadsto$ is slightly different from a standard parametrization of binary r.e. predicates (although the modifications are minor enough). It is, in a suitable sense, an effective disjoint sum of all r.e. transition systems: given an r.e. $\to$, define f on $\to$-states s by $f(s) = \langle e, s\rangle$ where $U_e = \{\langle s, \langle l,t\rangle\rangle \mid s \xrightarrow{l} t\}$. $\dashv$

There are countably many transition predicates meeting the specification of $\leadsto$ given in Proposition 3, although, as will become plain shortly, they share the same abstract structure. For now, observe that $\leadsto$ generalizes the functional analysis of programs, $\{e\}(x) \simeq y$ being the special case $[e] \overset{x,y}{\leadsto} \surd$, for some fixed encoding $[e]$ of e. More generally, returning to a transition predicate $\to$ with data pair labels as in (6), it is simple enough to define transformations from $\to$ to $\to_*$

$$\frac{p \xrightarrow{d,d'} q}{p \xrightarrow{d,d'}_* q} \qquad\qquad \frac{p \xrightarrow{d,d''}_* q \qquad q \xrightarrow{d'',d'} p'}{p \xrightarrow{d,d'}_* p'} \ ,$$

and then to $\to_{at}$

$$\frac{p \xrightarrow{d,d'}_* \surd}{p \xrightarrow{d,d'}_{at} \surd} \ ,$$

setting the stage for a characterization of input/output equivalence by a more general notion to be defined next. (Note that if $\to$ is r.e., then $\to_*$ and $\to_{at}$ are r.e. as well; i.e., $\cdot_*$ and $\cdot_{at}$ can be interpreted inside $\leadsto$.)

2.2. Bisimulations and extensionality generalized. For an equivalence R on states to "respect" $\to$, it is natural to expect that if sRs' then s' can "simulate" every transition from s

$$(\forall l \in L) \qquad (\forall x \xleftarrow{l} s)(\exists y \xleftarrow{l} s') \ xRy \ ,$$

and conversely, s can simulate s'

$$(\forall l \in L) \qquad (\forall y \xleftarrow{l} s')(\exists x \xleftarrow{l} s) \ xRy \ .$$

It will prove useful to consider binary relations $R \subseteq S \times S'$ between the states of possibly two distinct transition systems $\langle L, S, \to\rangle$ and $\langle L, S', \to'\rangle$ and to formulate the requirements above within the "back-and-forth" operator $\cdot^{bf}$

that sends R to the relation R^{bf} consisting of pairs $(s, s') \in S \times S'$ such that for every $l \in L$,

$$(\forall x \overset{l}{\leftarrow} s)(\exists y \overset{l}{\leftarrow}' s')\ xRy \quad \text{and} \quad (\forall y \overset{l}{\leftarrow}' s')(\exists x \overset{l}{\leftarrow} s)\ xRy\} \ .$$

R is a *bisimulation* (Park [**19**]) if $R \subseteq R^{bf}$. Because R occurs positively in R^{bf}, the operator $R \mapsto R^{bf}$ is ($\subseteq$-)monotone, and the largest bisimulation $\underline{\leftrightarrow}$ (between $\langle L, S, \rightarrow \rangle$ and $\langle L, S', \rightarrow' \rangle$) can be calculated co-inductively as $\bigcap \{ \underline{\leftrightarrow}_\alpha \mid \alpha < \text{card}(S \times S')^+ \}$, where

$$\underline{\leftrightarrow}_\alpha \quad = \quad \bigcap_{\beta < \alpha} (\underline{\leftrightarrow}_\beta)^{bf}$$

(e.g., Milner [**16**]). The relation $\underline{\leftrightarrow}$ is also called *bisimilarity* and is manifestly Σ^1_1 in $\rightarrow$ and $\rightarrow'$

$$s \ \underline{\leftrightarrow}\ s' \quad \text{iff} \quad (\exists R)\ R \text{ is a bisimulation relating } s \text{ to } s' \ .$$

The extensional character of bisimilarity $\underline{\leftrightarrow}$ is nicely illustrated by two simple examples. First, input/output equivalence can be recovered as bisimilarity relative to the transition predicate $\rightarrow_{at}$ described in the end of subsection 2.1[4]

$$p \text{ and } q \text{ are input/output equivalent} \quad \text{iff} \quad p \ \underline{\leftrightarrow}_{at}\ q \ .$$

This example is somewhat degenerate inasmuch as the notion of a bisimulation is of interest mainly for transition predicates $\rightharpoondown$ with long chains of transitions $s \overset{l}{\rightharpoondown} s' \overset{l'}{\rightharpoondown} s'' \overset{l''}{\rightharpoondown} s''' \cdots$. Our second example is of this kind, and furthermore makes do with a singleton label set (i.e., $|L| = 1$), in contrast to $\rightarrow_{at}$, which uses labels heavily. More specifically, a binary relation B is turned into a transition predicate $\rightarrow_B = \{(a, *, b) \mid aBb\}$, by introducing some fixed label $*$ to the pairs in B. We will henceforth speak of bisimulations and bisimilarity over a binary relation B, when we actually have in mind the transition predicate $\rightarrow_B$ formed from it. Now, defining the Mostowski collapse c_B of a well-founded binary relation B by

$$c_B(a) \quad = \quad \{c_B(b) \mid bBa\} \ ,$$

then, as pointed out in Aczel [**1**],

$$c_B(a) = c_B(a') \quad \text{iff} \quad a \underline{\leftrightarrow} a' \ ,$$

where $\underline{\leftrightarrow}$ is bisimilarity relative to B (i.e., $\rightarrow_B$). Aczel [**1**] goes on to develop a set theory supporting this equivalence, whether or not B is well-founded (so that, in particular, c_B is always well-defined).

[4]The map $\cdot_{at}$ can be arranged to respect bisimilarity (i.e., $s \ \underline{\leftrightarrow}_{at}\ t$ if $s \ \underline{\leftrightarrow}\ t$) by adjoining a fresh label $\surd$ and postulating

$$s \overset{\surd}{\rightarrow} t \quad \text{iff} \quad s = t = \surd \ .$$

Note that the last condition above is not satisfiable in $\rightsquigarrow$, but in some r.e. transition predicate with a copy in $\rightsquigarrow$.

Returning to the coinductive characterization of $\underline{\leftrightarrow}$, note that $S \times S' = \underline{\leftrightarrow}_0 \supseteq \underline{\leftrightarrow}_1 \supseteq \cdots \supseteq \underline{\leftrightarrow}_\omega \supseteq \underline{\leftrightarrow}_{\omega+1} \supseteq \cdots \supseteq \underline{\leftrightarrow}$, where $\underline{\leftrightarrow}$ may differ from $\underline{\leftrightarrow}_\omega$, as illustrated by the notorious pair

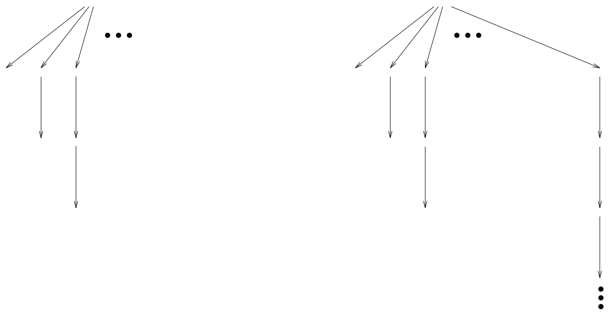

Figure 1

Confusing these trees is problematic inasmuch as the infinite branch can be interpreted as a failure of termination (or, say, an unfair merging of an infinite stream 1^∞ of 1's with 0, where 0 is forever neglected). The *Approximation Induction Principle* (AIP) of (for example) Baeten and Weijland [**2**] is (under an interpretation of equality as bisimilarity) the statement $\underline{\leftrightarrow} = \underline{\leftrightarrow}_\omega$. Just how badly AIP fails is a measure of the complexity (and hence, scope) of $\underline{\leftrightarrow}$. Nevertheless, a routine saturation argument can be employed to justify $\underline{\leftrightarrow} = \underline{\leftrightarrow}_\omega$.

Proposition 4. *For every transition predicate $\rightarrow \subseteq S \times L \times S$, there is a transition predicate $\rightarrow' \subseteq S' \times L \times S'$ with $\underline{\leftrightarrow}' = \underline{\leftrightarrow}'_\omega$ such that the first-order structure $\langle S, \{\stackrel{l}{\rightarrow}\}_{l \in L}\rangle$ is elementarily equivalent to $\langle S', \{\stackrel{l}{\rightarrow}'\}_{l \in L}\rangle$.*

Proposition 4 is an immediate consequence of the existence of recursively saturated elementary extensions (e.g., Barwise [**4**], p. 74), and the fact that a coinductive process of the sort that yields bisimilarity closes off at ω for recursively saturated models (e.g., Barwise [**4**], pp. 139, 210). At any rate, the moral of Proposition 4 is clear: first-order logic is inadequate for analyzing bisimilarity. The opposite of a saturation argument is an omitting types argument, a close companion of ω-completeness, and therefore ω-logic (or, as Girard [**11**] would put it, Π^1_1-logic). Curiously, omitting types is employed in Fernando [**9**] to bring out the first-order dependence on data, which is analyzed by reformulating transitions of the form $p \stackrel{d,d'}{\rightarrow} p'$ as $d \stackrel{p,p'}{\rightarrow} d'$. An altogether different use of omitting types is in proving the Gandy-Kreisel-Tait theorem (e.g., Barwise [**4**], p. 116) stating that the only sets a Π^1_1 theory is committed to are Δ^1_1. This fact is worth keeping in mind when establishing (as we will below) the non-Π^1_1 nature of bisimilarity relative to a certain logspace computable binary relation.

3. Pushing the analogy: semantics from syntax

Whereas termination is essential to an input/output (e.g., functional) view of computation, the question of termination takes on a different character within a more general theory of computational transitions that includes input/output transitions as special cases. Indeed, as has often been pointed out, non-termination is perfectly respectable, if not desirable, in so-called reactive systems (e.g., operating systems) that cease to be of service as soon as they terminate. The cost of admitting such immortality (formulated mathematically as non-well-foundedness) particularly in the presence of countable non-determinism is the subject of the present section.

As is implicit in Barwise, Gandy and Moschovakis [**3**], bisimilarity $\underline{\leftrightarrow}$ is Δ^1_1 in a transition predicate $\rightarrow$, provided $\rightarrow$ (or, more precisely, $\{(s,t) \mid (\exists l)\ s \xrightarrow{l} t\}$) is well-founded. It is easy to see why this should be the case. Recall that $R \subseteq S \times S$ is a bisimulation iff $R \subseteq R^{bf}$. Dually, call R a *co-bisimulation* if $R^{bf} \subseteq R$. Note that $\underline{\leftrightarrow}$ is a cobisimulation, and that there is a least cobisimulation $\sim$ which can be calculated inductively in the usual fashion, and which is a fixed point of $\cdot^{bf}$. Thus, since $\underline{\leftrightarrow}$ is the largest bisimulation, it remains to prove $\underline{\leftrightarrow}\ \subseteq \sim$, assuming $\rightarrow$ is well-founded. That is, it is enough to establish

$$(\exists s,s')\ s \underline{\leftrightarrow} s' \ \&\ s \not\sim s' \quad \text{implies}$$
$$(\exists l,t,t')\ t \underline{\leftrightarrow} t' \ \&\ t \not\sim t' \ \&\ s \xrightarrow{l} t\ ,$$

as this yields an infinite $\bigcup\{\xrightarrow{l} \mid l \in L\}$-descending sequence. But then, as $\sim$ is a cobisimulation, it follows from $s \not\sim s'$ that for some l, either (i) for some $t \xleftarrow{l} s$, $(\forall y \xleftarrow{l} s')\ t \not\sim y$, in which case $s \underline{\leftrightarrow} s'$ implies there is a $t' \xleftarrow{l} s'$ such that $t \underline{\leftrightarrow} t'$, or (ii) for some $t' \xleftarrow{l} s'$, $(\forall x \xleftarrow{l} s)\ x \not\sim t'$, and symmetrically, a $t \xleftarrow{l} s$ such that $t \underline{\leftrightarrow} t'$ follows from $s \underline{\leftrightarrow} s'$. Thus, if $\rightarrow$ is well-founded then $\underline{\leftrightarrow}$ is the unique fixed point of $\cdot^{bf}$ and $\underline{\leftrightarrow}$ is $\sim$

$$\begin{aligned} s \underline{\leftrightarrow} s' \quad &\text{iff} \quad (\exists R)\ \ Bis(R)\ \&\ Cobis(R)\ \&\ sRs' \\ &\text{iff} \quad (\forall R)\ \ Bis(R)\ \&\ Cobis(R)\ \supset\ sRs' \\ &\text{iff} \quad (\forall R)\ \ Cobis(R)\ \supset\ sRs'\ , \end{aligned}$$

where $Bis(R)$ and $Cobis(R)$ are the obvious predicates (arithmetic in $\rightarrow$) defining bisimulations and cobisimulations, respectively. Note that, in general, $s \underline{\leftrightarrow} s'$ is not equivalent to $(\exists R)\ Cobis(R)\ \&\ sRs'$ nor to $(\forall R)\ Bis(R) \supset sRs'$ (because of relations such as $S \times S$ and $=$).

Turning now to non-well-foundedness, what price must be paid for it? And what, in abstract, structural terms, do we get in return?

3.1. The challenge of non-well-foundedness. Recall that an *admissible set* (e.g., Barwise [**4**]) is a transitive set that (under the standard interpretation of $\in$) is a model of a certain weak set theory, KP. KP is characterized (among other axioms) by restricted, so-called Δ, forms of comprehension and collection. The paradigmatic example is where Δ is interpreted as (hereditarily) Δ^1_1, a principal motivation being to capture various facts about the Δ^1_1 sets on which a generalization can be based. In particular, two facts that generalize to some degree or another will interest us.

(a) The Π^1_1 sets are exactly the *inductive* sets — i.e., sets of the form $\{n \mid I_\varphi(0,n)\}$ for some elementary formula $\varphi(x,y,R)$ positive in the binary relation symbol R, where $I_\varphi(x,y)$ is the least fixed point of the operator

$$B \;\mapsto\; \{(m,n) \mid \varphi(m,n,B)\}$$

on binary relations B. Thus, the Δ^1_1 sets are exactly the sets that are both inductive and coinductive; in other words, the *hyperelementary* sets. This fact (due to Kleene and Spector) generalizes to countable sets, but may fail for uncountable sets (e.g., Moschovakis [**17**]).

(b) The set of Δ^1_1 well-orders is not Σ^1_1, implying that there are non-well-founded Δ^1_1 linear orders that have *no* Δ^1_1 descending sequences — so-called pseudo-well-orders. Defining the *well-founded part* $wf(\sqsubset)$ *of* a linear order $\sqsubset$ as the set of pairs (x,y) such that $x \sqsubset y$ and there are no $\sqsubset$-descending sequences from y, it follows that such a pseudo-well-order must have a well-founded part containing a copy of every Δ^1_1 well-order (as in the Feferman-Spector [**6**] result mentioned in section 1.2 above). Of course, there are no pseudo-well-orders if well-foundedness is hyperelementary (i.e., when the β-axiom mentioned in Barwise [**4**] holds).

Given a set A, Barwise, Gandy and Moschovakis [**3**] define

$$A^+ \;=\; \bigcap\{M \mid M \text{ is admissible and } A \in M\}$$

and go on to construct A^+ by a quotient construction similar to that of Aczel [**1**], where transitions are unlabelled and taken to be ordered pairs. It is crucial that $\underline{\leftrightarrow}$ be hyperelementary to push the argument in Aczel [**1**] through KP's restricted schemes of separation and collection. Assuming A is a non-empty transitive set closed under pairing, Barwise, Gandy and Moschovakis [**3**] encode A^+ by hyperelementary subsets of A that are well-founded trees. An inessential difference from Aczel [**1**] is that the transition predicates are unwound into trees, and subjected to the Mostowski collapse. An essential difference is that the predicates are assumed to be well-founded — an inductive condition, just as hyperelementarity is. But whereas well-foundedness makes $\underline{\leftrightarrow}$ inductive (in $\rightarrow$), hyperelementarity alone does not. To make this point, it will be sufficient to consider a linear order, which simplifies matters considerably. As usual, if $\sqsubset$ is a well-order, we will write $\mid \sqsubset \mid$ for its ordinal copy.

Proposition 5. *Let $\sqsubset$ be a linear order on ω, and for $a \in \omega$, write $wf(a)$ for the well-founded part of the order $\{(x,y) \mid x \sqsubset y \sqsubset a\}$. Let $\underline{\leftrightarrow}$ be bisimilarity relative to $\sqsubset$, with αth coinductive approximation $\underline{\leftrightarrow}_\alpha$.*

(i) *For all $a,b \in \omega$, and for every ordinal α,*

$$a \;\underline{\leftrightarrow}_\alpha\; b \quad \textit{iff} \quad |wf(a)| \cap \alpha \;=\; |wf(b)| \cap \alpha\ ,$$

whence $\underline{\leftrightarrow} \;=\; \underline{\leftrightarrow}_\beta$ *for* $\beta = |wf(\sqsubset)| + 1$, *and*

$$a \,\underline{\leftrightarrow}\, b \quad \textit{iff} \quad a = b \;\; \textit{or} \;\; \textit{neither } a \textit{ nor } b \textit{ is in } wf(\sqsubset)\ .$$

(ii) *Let* A *be an infinite set, and* $\sqsubset$ *be* A*-hyperelementary, non-well-founded but have no* A*-hyperelementary descending sequences. If* A*-hyperelementary sets admit hyperelementary well-orderings, then* $\underline{\leftrightarrow}$ *cannot be* A*-inductive.*

Proof. Prove part (i) by induction on ordinals α; the case where α is 0 or a limit is immediate. If $\alpha = \beta + 1$, then

$$\begin{array}{lll} a \,\underline{\leftrightarrow}_\alpha\, b & \text{iff} & (\forall x \sqsubset a)(\exists y \sqsubset b)\ x \,\underline{\leftrightarrow}_\beta\, y \;\; \& \\ & & (\forall y \sqsubset b)(\exists x \sqsubset a)\ x \,\underline{\leftrightarrow}_\beta\, y \qquad \text{(by definition)} \\ & \text{iff} & (\forall x \sqsubset a)(\exists y \sqsubset b)\ |wf(x)| \cap \beta \;=\; |wf(y)| \cap \beta \\ & & \& \;\; (\forall y \sqsubset b)(\exists x \sqsubset a) \\ & & \quad |wf(x)| \cap \beta \;=\; |wf(y)| \cap \beta \quad \text{(by ind. hyp.)} \\ & \text{iff} & |wf(a)| \cap \alpha \;=\; |wf(b)| \cap \alpha\ , \end{array}$$

where the last equivalence follows from the linearity of $\sqsubset$.

As for part (ii), assume towards a contradiction that $\underline{\leftrightarrow}$ were A-inductive. Then (adapting the proof of the Closure Theorem of Moschovakis [**17**], p. 33) the formula

$$\begin{array}{lll} \psi(t, n, m, T) & \text{iff} & [t = 0 \;\&\; (n, m) \in \{(x, y) \mid T(0, x, y)\}^{bf}] \text{ or} \\ & & [t = 1 \;\&\; (\exists x, y)(T(0, x, y) \;\&\; x \;\, \underline{\not\leftrightarrow}\; y)] \end{array}$$

(with $\underline{\leftrightarrow}$ equal to the 0-th section $J_\psi(0, \cdot, \cdot)$ of the largest fixed point J_ψ) defines a coinduction that takes one more step than coinduction based on $\cdot^{bf}$ (but not more than the least A-hyperelementary ordinal). Hence, $\underline{\leftrightarrow} = \underline{\leftrightarrow}_\alpha$ for some A-hyperelementary ordinal α, and there is an a ($\notin wf(\sqsubset)$) such that whenever $b \,\underline{\leftrightarrow}_\alpha\, a$, there is a descending sequence from b. Now, using an A-hyperelementary well-ordering of $\{x \mid x \,\underline{\leftrightarrow}_\alpha\, a\}$, construct an A-hyperelementary descending sequence from a, contradiction. $\dashv$

The significance of A-inductiveness is shown in part by its identification, over countable admissible sets A, with A-recursive enumerability. Setting aside the matter of inductiveness (which follows from well-foundedness) and concentrating simply on the ordinals, note that $\underline{\leftrightarrow}$ cannot equal $\underline{\leftrightarrow}_\beta$ for some ordinal β in A^+, or else β and $\beta + 1$ would be indistinguishable in A^+, which is absurd. This suggests the following elaboration of a picture found in Barwise [**4**].

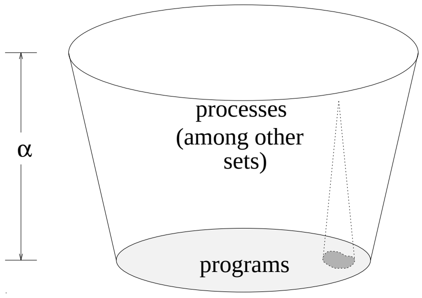

Figure 2

The idea behind Figure 2 is that a process is an equivalence class of (syntactic) programs, a minimal requirement on such an equivalence being that it be a bisimulation. The ordinal α is (among other things) a tight bound on the closure ordinal of an operator $\cdot^{bf}$ determining whether or not there is a bisimulation relating two programs. Indeed, the explicit Σ^1_1 formulation of $s \underline{\leftrightarrow} s'$, $(\exists R)$ R is a bisimulation relating s to s', can be read as

$$\text{there is a model where } s \text{ and } s' \text{ are equal ,}$$

since a bisimulation R yields a model after division: define $\rightarrow/R$ by

$$s \xrightarrow{l}/R\; t \quad \text{iff} \quad (\exists s', t')\; s'Rs \;\&\; t'Rt \;\&\; s' \xrightarrow{l} t' \;,$$

which, by the definition of a bisimulation, amounts to

$$s \xrightarrow{l}/R\; t \quad \text{iff} \quad (\exists t'Rt)\; s \xrightarrow{l} t' \;.$$

Note that $\{(s,s) \mid (\exists l,t)\; s \xrightarrow{l} t\}$ is a bisimulation between $\rightarrow/R$ and $\rightarrow$, and that $\rightarrow/R$ and $\rightarrow$ induce the same relation of bisimilarity, with, further, $(\rightarrow/R)/R' \;=\; \rightarrow/R' \;=\; (\rightarrow/R')/R$ when $R' \supseteq R$. We will be mainly concerned with $\rightarrow/\underline{\leftrightarrow}$, which will be abbreviated as $\rightarrow_\circ$. For a binary relation $>$, we will also write $>_\circ$ for $\rightarrow_\circ$ where $\rightarrow$ is $\{(n,0,m) \mid n > m\}$. Call a transition predicate $\rightarrow$ *reduced* if $\rightarrow$ is $\rightarrow_\circ$. (Note that $\rightarrow_\circ$ is reduced.) If, further, $\underline{\leftrightarrow}$ is $=$ (i.e., $a \underline{\leftrightarrow} b$ implies $a = b$), then the transition predicate $\rightarrow$ is *strongly extensional* (Aczel [**1**]). (We will confine ourselves to $\rightarrow \subseteq \omega \times \omega \times \omega$, so that the states in strongly extensional quotients are not $\underline{\leftrightarrow}$-equivalence classes, but representatives of such.)

A non-inductive notion of bisimilarity can be obtained over a transition predicate that is not only hyperarithmetic, but by Grigorieff [**13**] computable in logspace. The key is Σ^1_1 boundedness.

Theorem 6. *(i) (Fernando* [**9**]*) Bisimilarity* $\underline{\leftrightarrow}$ *over* $\succ$ *is* $\Sigma^1_1 - \Pi^1_1$.

(ii) $\succ_\circ$ *is* $\Sigma^1_1 - \Pi^1_1$.

Proof. Membership in Σ^1_1 is obvious. To see that the predicates lie outside Π^1_1, fix an $a_0 \in \mathcal{O}^* - \mathcal{O}$, and conclude from Proposition 5 (not to mention the extraction from $\mathcal{O}^*$ of Π^1_1 paths) that

$$\begin{array}{rcl}\{(b,0,c) \mid a_0 \succ b \;\&\; b \succ c \;\&\; b \in \mathcal{O}\} & = \\ \{(b,0,c) \mid a_0 \succ b \;\&\; b \succ c \;\&\; b \underline{\not\leftrightarrow} a_0\} & = \\ \{(b,0,c) \mid a_0 \succ b \;\&\; b \succ c \;\&\; b \not\succ_\circ a_0\} \end{array}$$

is well-founded with height ω_1^{CK}. ⊣

Before closing this subsection, let us remark that an infinite descending sequence in a transition system may also reflect computations over infinite streams of data — e.g. infinite precision reals.

3.2. Operational semantics reduced. The measure of complexity provided by Theorem 6 concerns a particular presentation of $\rightsquigarrow_\circ$; its robustness might be tested by trading off the complexity of transition predicate with the complexity of bisimilarity over that transition predicate. Towards that end, the following definitions will be useful. Given two transition predicates $\rightarrow$ and $\rightarrow' \subseteq \omega \times \omega \times \omega$, a binary relation R (on ω) *copies* $\rightarrow$ *into* $\rightarrow'$ if R is a bisimulation between $\rightarrow$ and $\rightarrow'$ such that

$$dom(R) \;=\; \{s \mid (\exists l, t)\; s \xrightarrow{l} t \text{ or } t \xrightarrow{l} s\}\;.$$

The predicate $\rightarrow$ *has a copy in* $\rightarrow'$ if there is a binary relation R copying $\rightarrow$ into $\rightarrow'$.

Theorem 7. *Fix an $a_0 \in \mathcal{O}^* - \mathcal{O}$, and suppose R copies $\{(b,0,c) \mid a_0 \succ b \;\&\; b \succ c\}$ into a predicate $\rightarrow$.*

(i) *If $\rightarrow$ is Σ^1_1, then bisimilarity $\underline{\leftrightarrow}$ over $\rightarrow$ is not Π^1_1.*
(ii) *If R and $\underline{\leftrightarrow}$ are Π^1_1, then $\rightarrow$ is* not *Π^1_1.*

Hence, if $\rightarrow$ is strongly extensional, then $\rightarrow$ is not Σ^1_1 nor, if R is Π^1_1, Π^1_1.

Proof. Fix a $b_0 \prec a_0$ such that $b_0 \notin \mathcal{O}$, and choose an s_0 such that $b_0 R s_0$. As $\{(b,0,c) \mid a_0 \succ b \;\&\; b \succ c\}$ is a linear order, R restricted to $\{b \in \mathcal{O} \mid a_0 \succ b\}$ must be a function (by Proposition 5), call it f. Appealing again to Proposition 5, note that

$$\begin{array}{rcl}\{(s,0,t) \mid s_0 \xrightarrow{0} s \;\&\; s \xrightarrow{0} t \;\&\; s_0 \underline{\not\leftrightarrow} s\} & = \\ \{(f(b),0,f(c)) \mid b_0 \succ b \;\&\; b \succ c \;\&\; b \in \mathcal{O}\} & = \\ \{(f(b),0,f(c)) \mid b_0 \succ b \;\&\; b \succ c \;\&\; f(b) \not\rightarrow_\circ f(b)\}\end{array}$$

is well-founded with height ω_1^{CK}, and that $\rightarrow_\circ$ is Π^1_1 if both $\underline{\leftrightarrow}$ and $\rightarrow$ are. (For the last equivalence, recall that Π^1_1 functions are Δ^1_1 — $f(x) = y$ iff $(\forall z \neq y)\; f(x) \neq z$.) ⊣

Moving in the opposite direction, it turns out that, as complex as $\rightsquigarrow_\circ$ is, that complexity need not provide a natural reduction of arithmetic transition predicates. That is to say, reducibility to $\rightsquigarrow_\circ$ becomes severely restricted if the reduction f of $\langle L, S, \rightarrow\rangle$ to $\langle \omega, \omega, \rightsquigarrow_\circ\rangle$ is required to have the form

$$f(x,y,z) \;=\; (g(x), h(y), g(z))$$

where for all $s, t \in S$ and $l \in L$,

$$s \xrightarrow{l} t \quad \text{iff} \quad g(s) \overset{h(l)}{\rightsquigarrow}_{\circ} \; g(t)$$

and, further, for every $u \in \omega$,

$$g(s) \overset{h(l)}{\rightsquigarrow}_{\circ} \; u \quad \text{implies} \quad (\exists t \in S) \; g(t) = u \quad .$$

The latter is a form of "end extension" condition, that is essential for g to be a bisimulation. The requirement in a bisimulation that h be the identity is not much of a restriction when there is only one label. Let us be careful about taking the matter of labelling for granted, however, as a great deal of complexity can be shoved over there. For example, consider the embedding

$$(\dagger) \qquad A \;\mapsto\; \{(s_0, n, \surd) \mid n \in A\}$$

of a set A into the transition predicate above with two states s_0 and $\surd$. An obvious consequence of this is

Proposition 8. *There is a Π^0_1 strongly extensional transition predicate without a copy in $\rightsquigarrow$ (or $\rightsquigarrow_\circ$).*

The embedding ($\dagger$) contrasts with the other more interesting extreme studied in Aczel [**1**] where the transition predicate is unlabelled (or, if one prefers, $|L| = 1$) and taken as $\in^{-1}$ (i.e., $\{(a, l, b) \mid b \in a\}$ for some fixed l). We will presently consider a more moderate position, to get a sense of how to pass between these two extremes. In particular, the notion of a copy will be relativized to the (computer science) notion of a trace. A trace is, in its simplest form, a sequence $(l_1, l_2, \ldots, l_n)$ of labels on successive transitions

$$s \xrightarrow{l_1} s_1 \;\&\; s_1 \xrightarrow{l_2} s_2 \;\&\; \cdots \;\&\; s_{n-1} \xrightarrow{l_n} s_n$$

where s_n is a "terminal" state. A terminal state is not meant here to be (necessarily) *any* element of the set

$$\{s \in S \;\mid\; \neg(\exists l, t) \; s \xrightarrow{l} t\} \;,$$

which is co-r.e., assuming $\rightarrow$ is r.e. Rather, we have in mind a more effective set of states, that can be marked by introducing a fresh label $\surd$ and postulating that for all states s and t,

$$s \xrightarrow{\surd} t \quad \text{iff} \quad s = t \text{ and } s \text{ is terminal.}$$

This modification to the transition predicate can*not* spoil its r.e. character, assuming the terminal states form an r.e. set. Nor can the map $\cdot_{tr}$ that sends $\rightarrow$ to the relation $\rightarrow_{tr}$ consisting of triples $(s, (l_1, l_2, \ldots, l_n), t)$ such that

$$\begin{aligned} &\surd \notin \{l_1, l_2, \ldots, l_n\} \;\&\; t \xrightarrow{\surd} t \;\& \\ &(\exists s_1, \ldots, s_{n-1}) \; s \xrightarrow{l_1} s_1 \;\&\; s_1 \xrightarrow{l_2} s_2 \;\&\; \cdots \;\&\; s_{n-1} \xrightarrow{l_n} t \;. \end{aligned}$$

(Since $\rightarrow_{tr}$ is r.e. if $\rightarrow$ is r.e., $\cdot_{tr}$ can be interpreted within $\rightsquigarrow$.) A transition predicate $\rightarrow$ is now said to *have a trace-equivalent copy in* $\rightarrow'$ if $\rightarrow_{tr}$ has a copy in $\rightarrow'_{tr}$. Observe that trace-equivalence is coarser than bisimilarity.

Moving some of the complexity away from the labelling, the argument (†) behind Proposition 8 can be used for

Proposition 8′. *(i) There is a well-founded Π^0_1 transition predicate with $|L| = 2$ that has no trace-equivalent copy in $\rightsquigarrow$.*

(ii) *There is a Π^0_1 transition predicate with $|L| = 2$ and branching bound 2 (i.e., for all n $|\{m \mid (\exists l)\ n \xrightarrow{l} m\}| \leq 2$) that has no trace-equivalent copy in $\rightsquigarrow$.*

(iii) *There is a strongly extensional Δ^0_2 transition predicate with $|L| = 2$ that has no trace-equivalent copy in $\rightsquigarrow$.*

Proof. The idea is to code a subset A of ω into a transition predicate $\rightarrow_A$ which has a copy in $\rightsquigarrow$ only if for some fixed label $l \neq \surd$, there is an s such that for all $n < \omega$,

$$\begin{aligned} n \in A \quad \text{iff} \quad & (\exists t)\ s \overset{l}{\rightsquigarrow} [t]_0 \ \&\ [t]_0 \overset{l}{\rightsquigarrow} [t]_1 \ \&\ \cdots \\ & \&\ [t]_{n-1} \overset{l}{\rightsquigarrow} [t]_n \ \&\ [t]_n \overset{\surd}{\rightsquigarrow} [t]_n \ , \end{aligned}$$

where $[t]_i$ gives the ith component of the finite sequence encoded (under some effective scheme) as t. Note that the above equivalence implies A is r.e. Now, given a binary 1-1 primitive recursive function $\langle \cdot, \cdot \rangle$ from $\omega \times \omega$ into $\omega - \{0\}$, define for (i),

$$\begin{aligned} \rightarrow_A \quad = \quad & \{(0, l, \langle n, n \rangle) \mid n \in A\} \ \cup \\ & \{(\langle n, m \rangle, l, \langle n, m-1 \rangle) \mid n \in A \text{ and } 0 < m \leq n\} \\ & \cup \{(\langle n, 0 \rangle, \surd, \langle n, 0 \rangle) \mid n \in A\} \ , \end{aligned}$$

while for (ii),

$$\begin{aligned} \rightarrow_A \quad = \quad & \{(\langle 0, n \rangle, l, \langle 0, n+1 \rangle) \mid n < \omega\} \ \cup \\ & \{(\langle 0, n \rangle, l, \langle 1, n \rangle) \mid n \in A\} \\ & \cup \{(\langle 1, n \rangle, \surd, \langle 1, n \rangle) \mid n \in A\} \ . \end{aligned}$$

Note that we cannot require $\rightarrow_A$ (for arbitrary $A \subset \omega$) to be both well-founded and finitely branching (by König) as every finite transition predicate has a copy in $\rightsquigarrow$.

As for (iii), let us reduce the transition predicate given by case (i) by choosing representatives of $\underline{\leftrightarrow}$-equivalence classes as follows. Given an m less than some element of A, let m_A be the least $x \in A$ that is $\geq m$. Note that for $m \in A$, $m_A = m$. Now, set $\rightarrow_A$ to

$$\begin{aligned} & \{(0, l, \langle n, n \rangle) \mid n \in A\} \ \cup \ \{(\langle a_0, 0 \rangle, \surd, \langle a_0, 0 \rangle)\} \ \cup \\ & \{(\langle m_A, m \rangle, l, \langle (m-1)_A, m-1 \rangle) \mid m > 0 \text{ and } m \text{ is less} \\ & \qquad\qquad\qquad\qquad \text{than some element of } A\} \end{aligned}$$

where a_0 is the least element of A. Observe that $x = m_A$ iff $x \in A \ \&\ x \geq m \ \&\ (\forall y < x)(y \in A \supset y < m)$, whence $\rightarrow_A$ is Δ^0_2. $\dashv$

Next, taking up the case $|L| = 1$, let us mention two features of a strongly extensional copy $\Rightarrow$ of $\rightsquigarrow$ that would seem to render it unsuitable for a set-theoretic characterization (as in Aczel [**1**] or Barwise, Gandy and Moschovakis [**3**]). We have in mind here, of course, a restriction of the transition predicate to one label, which we henceforth ignore. Let $V = \{a \mid (\exists b)\ a \Rightarrow b \text{ or } b \Rightarrow a\}$.

(i) A "universal" set exists — i.e., there is an a in V such that $a \Rightarrow b$ for every b in V. (This a is the copy of state 0 in the r.e. transition predicate $\{(0, n+1) \mid n \in \omega\} \cup \{(n+1, m+1) \mid n \rightsquigarrow m\}$ obtained by "rooting" $\rightsquigarrow$.)

(ii) V is not closed under very simple comprehension schemes: V contains an $x = \{x\} \cup \omega_1^{CK}$ but not $\omega_1^{CK} \subset x$. (This is an immediate consequence of Π_1^1 paths through $\mathcal{O}$ and Σ_1^1 boundedness.)

3.3. The question of strongly extensional Π_1^1 approximations. An example of a strongly extensional copy $\Rightarrow$ of $\rightsquigarrow$ is $\rightsquigarrow_\circ$ restricted to the set

$$A_0 \quad = \quad \{s \in \omega \mid (\forall s' < s)\ s' \underline{\not\leftrightarrow} s\}$$

of least $\underline{\leftrightarrow}$-representatives. That is, for all $s, l, t \in \omega$,

$$\begin{aligned} s \overset{l}{\Rightarrow} t \quad \text{iff} \quad & (\exists t' \underline{\leftrightarrow} t)\ s \overset{l}{\rightsquigarrow} t' \ \&\ (\forall s' < s)\ s' \underline{\not\leftrightarrow} s \ \ \& \\ & (\forall t' < t)\ t' \underline{\not\leftrightarrow} t \ . \end{aligned}$$

Whether $\Rightarrow$ (or any other strongly extensional copy of $\rightsquigarrow$) is Π_1^1 is not known by the present author. The problem is that the first conjunct $\rightsquigarrow_\circ$ is $\Sigma_1^1 - \Pi_1^1$. But now, just as a Π_1^1 path through $\mathcal{O}$ can be built from a relativization $\mathcal{O}^*$ of $\mathcal{O}$ to hyperarithmetic sets, we might consider a notion of pseudo-bisimilarity defined up to hyperarithmetic sets. Given a transition system $\langle L, S, \rightarrow \rangle$, with $L, S \subseteq \omega$, define the binary relation $\sim$ on S by

$$\begin{aligned} a \sim b \quad \text{iff} \quad & \text{there is a hyperarithmetic bisimulation} \\ & \text{relating } a \text{ to } b \ . \end{aligned}$$

Proposition 9. *$\sim$ is a bisimulation, and, if $\rightarrow$ is Δ_1^1, is Π_1^1.*

An obvious Π_1^1 approximation of $\Rightarrow$ is then $\{(s, l, t) \mid s \overset{l}{\rightarrow}/\!\sim t \text{ and } s, t \in A\}$, for some Δ_1^1 copy $\rightarrow$ of $\rightsquigarrow$, and some "suitable" Π_1^1 set A. More precisely, a subset A of ω is a *suitable representation* if

(S1) $(\forall s \in \omega)(\exists! a \in A)\ a \underline{\leftrightarrow} s$

(S2) $(\forall a \in A)(\forall b, l \text{ s.t. } a \overset{l}{\rightarrow} b)(\exists c \in A)\ c \underline{\leftrightarrow} b \ \&\ a \overset{l}{\rightarrow}/\!\sim\ c$.

Note that (S1) determines a function f_A, in terms of which (S2) becomes

$$(\forall a \in A)(\forall b, l \text{ s.t. } a \overset{l}{\rightarrow} b) \qquad a \overset{l}{\rightarrow}/\!\sim\ f_A(b) \ .$$

For a Π_1^1 suitable representation A, Theorem 7 implies that f_A cannot be Π_1^1. Returning to the set A_0 of least $\underline{\leftrightarrow}$-representatives, note that A_0 is Π_1^1 and (with $\rightarrow\, = \,\rightsquigarrow$) meets (S1) by a function that is not manifestly Π_1^1. Unfortunately, it is unlikely that A_0 meets (S2) in view of

Proposition 10. *For all $a, b \in \mathcal{O}^* - \mathcal{O}$, if $a \succ b$ then $a \not\sim b$ (although $a \underline{\leftrightarrow} b$).*

Proof. Otherwise, construct a hyperarithmetic descending sequence from a, appealing to

$$a \succ b \text{ and } a \sim b \quad \text{implies} \quad (\exists c \prec b)\ b \sim c\ ,$$

and repeatedly choosing the first c according to some enumeration of $\succ$. $\dashv$

That is, a suitable representation (relative to $\leadsto$) cannot contain an element a of $\mathcal{O}^* - \mathcal{O}$. This is easily repaired, however; replace a by an a' bisimilar to a with $a' \succ a'$. Thus, one might try to build successive improvements A_{n+1} of A_n, with $A = \lim_{n\to\infty} A_n$ (i.e., $s \in A$ iff $(\exists N)(\forall n > N)\ s \in A_n$). Alternatively, we might focus on beefing up $\rightarrow^0\ = \leadsto/\sim$ into a Π^1_1 set $\rightarrow_\omega = \bigcup_{n<\omega} \rightarrow^n$ such that for all $a, a' \in A_0$,

$$(\forall b, l) \qquad a \overset{l}{\leadsto} b \ \&\ b \underline{\leftrightarrow} a' \quad \supset \quad a \overset{l}{\rightarrow}_\omega a'\ ,$$

a Π^1_1 condition suggesting that a strongly extensional copy of $\leadsto$ be formed from restricting $\rightarrow_\omega$ to A_0. A combination of improvements on A_0 and $\leadsto/\sim$ might be required, but alas, the author has been unable to carry some such construction to completion.

4. Discussion

The preceding is in no small measure inspired by Feferman [**7**], which concludes with the remark

> ... I think that it can be argued that — whatever a "fundamental" theory may tell us is the basic or underlying mechanism for given technological processes or systems — it is necessary for our design and use of such to think of them at various conceptual levels and with various modular forms of organization, that is *we must think of them in structural terms*. The story of Turing's "oracle" and its significance for actual computability is but one example among many of this characteristic *modus operandi* of human intelligence.

While transition predicates constitute "but one example among many" possible formulations of "conceptual levels", the generality of that approach is promising. Furthermore, that generality can be grounded in processes that are *mechanical* inasmuch as their transitions can be programmed. Construed in ordinary recursion-theoretic terms, the transitions that can be programmed are based on process names (i.e., syntax) and may fail to be substitutive with respect to non-trivial notions of process equality (requiring, as that does, computational notions for sets — i.e., equivalence classes — rather than for natural numbers). Instead, a weaker form of substitutivity is provided by the (key) concept of a bisimulation (applied to a transition system where programs are states), from which a natural notion of process equality is then derived. The highly uncomputable character of the resulting logical abstractions should not disturb us so long as we do not confuse reasoning *about* mechanical computation with carrying out a mechanical computation. Recall how the "absolute" character of Church's thesis contrasts with the "relative" character of Gödel incompleteness (i.e., consistency). "The necessity for intuition" in Turing [**21**] cannot be "eliminated"; rather, the ineluctability of partiality leads to relative computability and various notions of consistency.

References

[1] Peter Aczel. *Non-Well-Founded Sets.* CSLI Lecture Notes Number 14, Stanford, 1988.

[2] J.C.M. Baeten and W.P. Weijland. *Process Algebra.* Cambridge Tracts in Theoretical Computer Science 18. Cambridge University Press, 1990.

[3] J. Barwise, R.O. Gandy, and Y.N. Moschovakis. The next admissible set. *Journal of Symbolic Logic*, 36(1), 1971.

[4] Jon Barwise. *Admissible Sets and Structures.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1975.

[5] D. Bjørner, A.P. Ershov, and N.D. Jones. *Partial Evaluation and Mixed Computation.* Elsevier, IFIP, 1988.

[6] Solomon Feferman. Turing in the land of 0(z). In R. Herken, editor, *The Universal Turing Machine.* Kammerer and Unverzagt, 1988.

[7] Solomon Feferman. Turing's "oracle": from absolute to relative computability and back. In the proceedings of a symposium on Structures in Mathematical Theories, San Sebastian, Spain, 1990.

[8] Tim Fernando. Provably recursive programs and program extraction. In J. Leach Albert et al, editor, *Proc. ICALP '91*, LNCS 510. Springer-Verlag, Berlin, 1991. A revision of the paper (entitled simply 'Provably recursive programs') appears as CWI Technical Report CS-R9212, Amsterdam, March 1992.

[9] Tim Fernando. Bisimulations and predicate logic. Technical Report CS-R9333, Centre for Mathematics and Computer Science (CWI), Amsterdam, May 1993. A revision has been accepted for publication in the *Journal of Symbolic Logic*.

[10] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge Tracts in Theoretical Computer Science 7. Cambridge University Press, 1989.

[11] Jean-Yves Girard. *Proof Theory and Logical Complexity*, volume 1. Bibliopolis, Napoli, 1987.

[12] Kurt Gödel. Remarks before the Princeton bicentennial conference on problems in mathematics. In S. Feferman et al, editor, *Collected Works, volume* II. Oxford University Press, 1990.

[13] Serge Grigorieff. Every recursive linear ordering has a copy in DTIME-SPACE$(n, log(n))$. *Journal of Symbolic Logic*, 55(1), 1990.

[14] Stephen Cole Kleene. On the interpretation of intuitionistic number theory. *Journal of Symbolic Logic*, 10, 1945.

[15] Georg Kreisel. On the interpretation of non-finitist proofs, I. *Journal of Symbolic Logic*, 16(4), 1951.

[16] Robin Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[17] Yiannis N. Moschovakis. *Elementary Induction on Abstract Structures.* North-Holland, Amsterdam, 1974.

[18] Yiannis N. Moschovakis. A model of concurrency with fair merge and full recursion. *Information and Computation*, 93(1), 1991.

[19] David Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proc. 5th GI Conference*, LNCS 104. Springer-Verlag, Berlin, 1981.

[20] Gordon D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[21] Alan Turing. Systems of logic based on ordinals. *Proc. London Math. Soc.*, 45, 1939.

3

Are boxes just parentheses?

December 5, 1994

Chapter 5 of Kamp and Reyle [**2**] formulates conditions of the form $\mathtt{write}(e, x, y)$, where e is an event, as

(1) $$e \;:\; \boxed{\mathtt{write}(x, y)}$$

to indicate that "the discourse referent for that event has a status different from the other discouse referents in the condition" (p. 511). States s are treated similarly, giving rise to conditions such as

(2) $$s \;:\; \boxed{\mathtt{ill}(x)}\;.$$

Recently, de Swart [**5**] has argued that in some cases, a discourse referent of type state ought to be introduced alongside negative sentences, yielding conditions of the form

$$s \;:\; \neg K\;,$$

where K is a DRS. This suggestion departs strictly from the treatment of negative sentences in Kamp and Reyle [**2**], requiring an extension of the model-theoretic framework there to specify when conditions

(3) $$s \;:\; \gamma$$

are verified, where γ is an *arbitrary* condition. De Swart also introduces a coercion operator C_{se} that takes predicates labeled by states to predicates labeled by events, as well as a complementary operator C_{es} that does the reverse (mapping event-labeled predicates into state-labeled ones). She borrows here the idea of a progressive operator PROG from page 576 of Kamp and Reyle [**2**], which applied to (1) leads to conditions of the form

(4) $$s \;:\; \boxed{\mathrm{PROG}(\mathtt{write})(x, y)}\;.$$

But what exactly are these coercion operators? Can the boxes be put to better use than as parentheses? What does it mean to verify conditions such as (3)? And what about events? The present note links these questions, by proposing that

(∗) *states label conditions, while events label DRSs.*

Under (∗),

(i) the event-labeled condition corresponding to (3) would have the form

$$e \;:\; K\;,$$

(ii) (1) would remain unchanged (at least under a certain interpretation described below in §3), but

(iii) (2) would be simplified into

$$s \; : \; \texttt{ill}(x) \; ,$$

while

(iv) the force of PROG in (4) would (under the simplest interpretation) be to unpack the box

$$s \; : \; \texttt{write}(x, y) \; .$$

1. Simple-minded coercion

A first attempt at implementing coercion is through the following systematic use of boxes —

C_{se}: draw a box around the condition to form a DRS

$$C_{se}(\gamma) \quad = \quad \boxed{\gamma}$$

C_{es}: negate the DRS, draw a box around it, and then negate once more

$$C_{es}(K) \quad = \quad \neg \, \boxed{\neg K} \; .$$

This is not the only system of coercion operators consistent with the proposal, just the first that comes to mind.[1] But what does it mean?

2. Some intuitions

On its face, the proposal $(*)$ (viz., that states label conditions, while events label DRSs) goes beyond recognizing (as in Kamp and Reyle [**2**]) the special status of events and states (relative to other discourse referents), by further speculating about the difference between events and states. But what makes conditions state-like, while events DRS-like? There is, to begin with, the rather striking parallel between

> the suggestion from de Swart [**5**] that the negation of an event is a state

and

> the fact (by stipulation in DRT) that the negation of a DRS is a condition.

Then, there is the point that DRSs, like events, involve the possibility of change, whereas conditions, like states, do not.

More precisely, recall (if necessary, consulting Muskens [4] or Fernando [**1**]) that on the one hand, a DRS K can be interpreted semantically as a binary

---

[1] Other coercion operators might actually change atomic predicates, although, as Kamp and Reyle [**2**] point out in the case of PROG (page 576), just what that modified predicate is may not be terribly clear.

relation $P(K) \subseteq S \times S$ on the set S of world-sequence pairs (M, f), (N, g), characterized by

$$(M, f)\ P(K)\ (N, g) \quad \text{iff} \quad \text{on input } (M, f), \ K \text{ returns output } (N, g)$$

(where $M = N$ and $f \subseteq g$ turn out to be necessary conditions for either side of the equivalence above). On the other hand, a condition γ amounts semantically to a subset $v(\gamma)$ of S, given by its verifying embeddings

$$(M, f) \in v(\gamma) \quad \text{iff} \quad f \text{ verifies } \gamma \text{ in } M.$$

In particular, returning to the coercion operations described in §1,

$$\begin{aligned} P(\boxed{\gamma}) &= \{((M, f), (M, f)) \mid (M, f) \in v(\gamma)\} \\ v(\neg \boxed{\neg K}) &= dom(P(K)) \ . \end{aligned}$$

3. Extending the semantic interpretation (two alternatives)

How can the model-theoretic framework in Chapter 5 be extended to treat $s : \gamma$ and $e : K$? Fix a first-order model M with additional binary predicates ϵ and ϵ', the idea being that ϵ specifies the objects of a state,[2] while ϵ' specifies the objects of an event, the significance of which is given by following definitions.

Definition 1. *An M-embedding f* verifies $s : \gamma$ *if either γ is an atomic condition*[3]*, in which case verification of $s : \gamma$ is treated as in Chapter 5 (i.e., with s added as an argument to γ), or γ is not atomic and the following holds:*

(i) $s \in dom(f)$,
(ii) *there is a subfunction $g \subseteq f$ such that for every $x \in dom(g)$, $g(x)\ \epsilon\ f(s)$, and*
(iii) $g \cup \{(s, f(s))\}$ *verifies γ.*

Similarly,

Definition 2. *An M-embedding f* verifies $e : K$ *if*

(i) $e \in dom(f)$,
(ii) *there is a subfunction $g \subseteq f$ such that for every $x \in dom(g)$, $g(x)\ \epsilon'\ f(e)$,*
(iii) *there is an M-embedding $h \supseteq g$ such that for every $x \in dom(h - g)$, $h(x)\ \epsilon'\ f(e)$, and*
(iv) *on input $g \cup \{(e, f(e))\}$, the DRS K can output $h \cup \{(e, f(e))\}$ — i.e.,*

$$(M, g \cup \{(e, f(e))\}) \quad P(K) \quad (M, h \cup \{(e, f(e))\}) \ .$$

---

[2]For example, Kamp [**3**] requires that an event e be temporally contained in a state s in order that $\mathsf{e}\ \epsilon\ \mathsf{s}$.

[3]Atomic conditions are simply atomic formulas, and, if iterated labeling $s : s' : \cdots : s^{(n)} : \gamma$ is allowed, all conditions obtained from atomic formulas by such labeling.

Note that Definitions 1 and 2 impose no restrictions on the interpretations of non-atomic conditions γ and DRSs K beyond (a) that built into their constituent atomic conditions, and (b) the requirements on embeddings involving ϵ and ϵ'. Additional restrictions would, as suggested by Kamp [**3**], need to be explictly spelled out within the conditions and DRSs. A moment's reflection shows that the result of applying Definition 2 to $e : \boxed{\gamma}$ is not quite faithful to Chapter 5. The expression $e : \boxed{\gamma}$ from Chapter 5 should instead be regarded, under Definition 2, as an abbreviation of $e : \boxed{s_e : \gamma}$, where s_e is some (canonical) projection of the event e as a state which can then label γ. The question arises as to whether (or not) the proposal $(*)$ is an empty dogma (with certain perverse consequences, such as the cluttered notation s_e for e). Certainly, a good deal of work remains to be done to defend $(*)$ convincingly against that charge.[4]

It is easy enough, however, to read, within that "empty dogma," the expression $e : \boxed{\gamma}$ from Chapter 5 as $e : K$, where K is literally $\boxed{\gamma}$, by adopting a semantic interpretation that is different from (and arguably superior to) Definitions 1 and 2. Rather than introducing binary predicates ϵ and ϵ', non-atomic conditions and DRSs can be relativized to the states and events that (respectively) label them, by requiring that all constituent atomic conditions be verified relative to the labels. In particular, for every atomic formula γ, the condition $e : \boxed{\gamma}$ would be verified by the same embeddings as in Chapter 5. The only difference is that atomic relations must now be relativized to finite sequences of events and/or states, rather than just one state or event. Such an extension would in any case be required to support iterated labeling $s : s' : \cdots : s^{(n)} : \gamma$.

To be more precise, fix a first-order model M such that for every relation symbol R, and every finite sequence α of events and/or states (in M), the model M specifies an n-ary relation, call it R^α_M, on its universe.[5]

Revised definition. *An M-embedding f* verifies *a condition γ (exactly) if $f \in v^{\langle\rangle}_M(\gamma)$, where $\langle\rangle$ is the empty sequence, and the (α-relativized verification) sets $v^\alpha_M(\gamma)$ are defined alongside (α-relativized input/output) relations $P^\alpha_M(K)$ (in the style of Fernando [**1**]) for DRSs K and conditions γ generated*

[4] The reader not inclined to accept $(*)$ may nonetheless find Definition 1 useful for specifying when $s : \gamma$ is verified, where s is an *event* or state. The revised definition below, however, might, for that purpose, be better still.

[5] The sets R^α_M can be obtained by assuming the signature of M includes an $(n+k)$-ary relation symbol R^k, for every n-ary relation symbol R and every natural number $k \geq 0$. Then, for k equal to the length of α, take

$$R^\alpha_M \quad = \quad \{\langle m_1, \ldots, m_n\rangle \mid M \models R^k[m_1, \ldots, m_n, \alpha]\} \ .$$

Of course, if the model M is closed under finite sequences of states and/or events, a single $n+1$-ary relation symbol R^1 will do.

syntactically as follows

$$\begin{array}{rcl} K & ::= & (U, \emptyset) \ \mid \ K^\frown\gamma \\ \gamma & ::= & R(x_1, \ldots, x_n) \ \mid \ \neg K \ \mid \ s : \gamma \ \mid \ e : K \end{array}$$

where U is a set of variables (and a DRS can be flattened into a pair (U, C), with C a finite set of conditions, by setting $(U, C)^\frown\gamma = (U, C \cup \{\gamma\})$). The corresponding semantic interpretations (relativized to M and α) are

$$\begin{array}{rcl} f \ P_M^\alpha(U, \emptyset) \ g & \text{iff} & f \subseteq g \text{ and} \\ & & dom(g) = dom(f) \cup U \\ f \in v_M^\alpha(R(x_1, \ldots, x_n)) & \text{iff} & R_M^\alpha(f(x_1), \ldots, f(x_n)) \\ f \ P_M^\alpha(K^\frown\gamma) \ g & \text{iff} & f \ P_M^\alpha(K) \ g \text{ and } g \in v_M^\alpha(\gamma) \\ f \in v_M^\alpha(\neg K) & \text{iff} & f \notin dom(P_M^\alpha(K)) \\ f \in v_M^\alpha(s : \gamma) & \text{iff} & s \in dom(f) \text{ and } f \in v_M^{\alpha^\wedge f(s)}(\gamma) \\ f \in v_M^\alpha(e : K) & \text{iff} & e \in dom(f) \text{ and} \\ & & f \in dom(P_M^{\alpha^\wedge f(e)}(K)) \end{array}$$

where $\alpha^\wedge \mathsf{a}$ is the sequence α, with a tagged on at the end.

Observe that the revised definition extends ordinary DRT in a conservative manner, and that the sequence superscript α can be ignored except for the conditions $s : \gamma$ and $e : K$. It differs from Definitions 1 and 2 in, for instance, relativizing γ to e in conditions such as $e : \boxed{\gamma}$ (as in Chapter 5), and pushing these relativizations through more complicated constructions such as $s : (e : K)$. The price paid is that suitable axioms should (according to one's favorite theory of states and events) be imposed relating R_M^α to R_M^β when (for example) α is a subsequence of β.

4. Discussion

Recall that event-labeled conditions in de Swart [**5**] are of the form $e : \gamma$ (or $e : \boxed{\gamma}$, with the box used parenthetically, as in Chapter 5 of Kamp and Reyle [**2**]). The extension described in the previous section raises the question as to whether there are natural languages (perhaps with more complicated tense systems than French) for which conditions of the form $e : K$ are useful, where K is a DRS (U, C) with a non-empty set U of discourse markers. Even if no such languages exist, however, it should be noted that de Swart [**5**] employs conditions of the form $s : \gamma$, with full-fledged DRSs built inside γ, suggesting that such event-labeled conditions may nevertheless be useful in analyzing how the $s : \gamma$'s can be formed from $e : K$'s. Such an analysis would presumably require coercion operations between states and events (and not just between conditions and DRSs), for which there are obvious candidates (in line with §1). As §3 stands, however, the characterization of verification of $s : \gamma$ is essentially independent of that of $e : K$.

It is true enough that the notion of a condition is, in principle, dispensable, as a condition can be understood as a special (pathological) kind of DRS (whose output can never differ from its input). In practice, however, the distinction between DRSs and conditions is a very convenient one. Furthermore, the possibility that there is more to the distinction between states and events than in that between conditions and DRSs is no argument against the proposal $(*)$, construed modestly.

References

[1] Tim Fernando. What is a DRS? In R. Cooper and J. Groenendijk, editors, *Integrating Semantic Theories II*. Dyana deliverable R2.1.B, September 1994. To be presented in a workshop on Computational Semantics at Tilburg (December 1994).

[2] H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht, 1993.

[3] Hans Kamp. Personal communication, December 1994.

[4] Reinhard Muskens. A compositional discourse representation theory. In *Proc. Ninth Amsterdam Colloquium*. ILLC, University of Amsterdam, 1994.

[5] Henriëtte de Swart. Negation and temporal structure. Hand-out for a talk in Tübingen, November 1994.

4

Computational foundations for dynamic accounts of presuppositions

1. Introduction

It is one thing to understand a natural language, quite another matter to spell out what it is that is understood (let alone just how that understanding comes about). It is considerably simpler to devise an artificial language of *formulas*, and to define a *semantic* function $[\![\cdot]\!]$ specifying the *meaning* $[\![\varphi]\!]$ of a formula φ. Of course, the plausibility of such an analysis rests on the extent to which the meaning of a natural language expression e can be characterized as $[\![\varphi_e]\!]$, for some (preferably "natural") association of formulas φ_e with natural language expressions e. The present paper is concerned, more concretely, with analyses of presuppositions by formulas subjected to so-called dynamic semantics, under which the meaning of a formula φ is equated with its *context change potential* (CCP) — i.e., a binary relation $[\![\varphi]\!]$ on (the set of) *contexts*, or *states*, s and s' specifying the input/output behavior of φ

$$(1) \qquad s[\![\varphi]\!]s' \quad \text{iff} \quad \text{on input } s,\ \varphi \text{ can return output } s'\ .$$

A paradigmatic example of dynamic semantics is *Discourse Representation Theory* (DRT, Kamp and Reyle [**13**]), central to which is an algorithm constructing *discourse representation structures* (DRS's) from syntactic analyses of natural language expressions. That algorithm is studied in a much simplified form below, with both DRS's and syntactic analyses of natural language expressions replaced by formulas akin to those in first-order logic, following the spirit (but not the letter) of *Dynamic Predicate Logic* (DPL, Groenendijk and Stokhof [**8**]). Although DPL departs in significant respects from DRT, much of what follows applies to both DRT and DPL, hence the notation DRT/DPL. By contrast, various analyses of presuppositions based on line (1) above build on semantic structures going beyond that of DRT/DPL, giving rise to the question of the adequacy of DRT/DPL. That question is investigated at length below, alongside the more fundamental problem of computing CCP's (in accordance with Church's thesis).

To understand what is going on here, it is useful to ask two simple questions, fleshing out line (1)

(**Q1**) What is a state s?

(**Q2**) When does φ define a transition from s
— i.e., when is s in $dom([\![\varphi]\!])$?

At its most elementary level, DRT/DPL provides the following answers:

> a state s is a partial function $f : Var \rightharpoondown |M|$ (called an "M-embedding" in DRT, and assumed to be total in DPL) from a fixed set Var of variables to the universe $|M|$ of some first-order model M,

and

> $s \in dom(\llbracket\varphi\rrbracket)$ exactly if φ is true (or better — in view of section 5 below —, satisfiable) in s.

Accordingly, the negation $\neg\varphi$ of a formula φ is analyzed in DRT/DPL as follows

$$s\llbracket\neg\varphi\rrbracket s' \quad \text{iff} \quad s = s' \text{ and } s \notin dom(\llbracket\varphi\rrbracket) \ .$$

Observe that DRT/DPL negation $\neg$ is "semantic" insofar as $\llbracket\neg\varphi\rrbracket$ depends only on $\llbracket\varphi\rrbracket$; that is, if $\llbracket\varphi\rrbracket = \llbracket\psi\rrbracket$ then $\llbracket\neg\varphi\rrbracket = \llbracket\neg\psi\rrbracket$. But is the notion of negation used to identify presuppositions semantic (in that same sense)? What is troublesome about presuppositions is that they concern not only *what* information is conveyed, but also *how* it is presented, a point driven home to the present author by E. Keenan through the following pair of sentences:

`John is the one linguist who came to the party.`

`John is the only linguist who came to the party.`

Presuppositions aside, these two sentences would seem to express the same claim (supported, as they are, by exactly the same contexts). But now, to uncover their presuppositions, form their negations by changing "`is`" to "`is not`", and the semantic equivalence breaks down. Evidently, neither this syntactic negation operation nor the extraction of presuppositions can be interpreted semantically.

Alternatively, negation can be "semanticized" (compositionally) by enriching the semantic interpretation of a formula to carry its negation. That is, the syntax/semantics distinction between *how* information is conveyed ($\approx$ syntax) and *what* information is presented ($\approx$ semantics) can be blurred by turning a bit of the "how" into "what." Repackaging an embedding $f : Var \rightharpoondown |M|$ as a *world-sequence pair* (M, f), Heim [**11**] and Beaver [**1**] pass to a notion of state — commonly referred to as an *information state* — given by a set of world-sequence pairs, allowing not only transitions to the empty set, in cases of falsity, but also the possibility that no output is returned. Staying at the elementary level where states are world-sequence pairs — or wsp's, for short —, van Eijck [**3**] introduces transitions to an "error state", called ϵ, in cases of presupposition failure. In terms of questions (**Q1**) and (**Q2**) above, the situation can be described as follows

	(**Q1**)	(**Q2**)
DRT/DPL	wsp (M, f)	satisfiability
Heim/Beaver	set of wsp's	presupposition success
van Eijck	wsp or ϵ	satisfiability or presup failure

The "positive" content (+) of a formula can be distinguished from its "negative" content (−) according to

	(+)	(−)
Heim/Beaver	transition to non-empty set	transition to $\emptyset$
van Eijck	transition to non-ϵ-state	no transition

An explicit separation between positive and negative components is provided by the transition/anti-transition pairs $([\![\varphi]\!]^+, [\![\varphi]\!]^-)$ of Krahmer [**15**], where ideas from ordinary 3-valued logic are adapted to DPL. Now, the present work proposes further to restore the syntax-semantics distinction by defining translations pos and neg between formulas such that $[\![\varphi]\!]^+ = [\![\mathsf{pos}(\varphi)]\!]$ and $[\![\varphi]\!]^- = [\![\mathsf{neg}(\varphi)]\!]$ under the usual DRT/DPL interpretation $[\![\cdot]\!]$, and characterizing the presuppositions $\mathsf{pres}(\varphi)$ of a formula φ as $\mathsf{pos}(\varphi) \vee \mathsf{neg}(\varphi)$. That is, rather than semanticizing presuppositions by changing the answers of DRT/DPL to (**Q1**) and (**Q2**), Keenan's intuition is developed via syntactic manipulations that keep the DRT/DPL semantic framework largely intact. Thus, van Eijck's introduction of an error state into the semantic framework can be avoided by "syntacticizing" φ-transitions to ϵ as $\neg\mathsf{pres}(\varphi)$. It is important here not to confuse DRT/DPL negation $\neg$ with syntactic translation*s* neg used to test presuppositions (of which there might be a multitude, in contrast to the single semantic notion $\neg$ of negation). A common semantic basis would facilitate a comparison of different accounts of presuppositions, including van der Sandt [**18**]. A reduction to DRT, in particular, is of both scientific and practical interest, in view of the amount of extant work using, analyzing and implementing DRT. There is also the broader and, as it turns out, related question of computing CCP's, which is taken up in section 2. The central thrust of the present work is to *formulate dynamic accounts of presuppositions within a* computational semantic *framework by introducing on top of that framework translations that are naturally regarded as* pragmatic *constructs.*[1] Such translations are studied in section 3, and their connection with pragmatic approaches to presuppositions considered briefly in section 4, where the main problem is put to be that of choosing an appropriate translation.

Turning finally to the logical motivation for adopting dynamic semantics, it must be conceded that there is, at the surface, nothing particularly dynamic about characterizing presupposition as the disjunction of positive and negative components. That is not to say, however, that the shift from truth to change (as the basic semantic concept) is without consequence. In particular, that shift suggests presentations of $\mathsf{pos}(\varphi)$ as presupposition-assertion pairs $(\mathsf{pres}(\varphi), \mathsf{as}(\varphi))$, complementing the reduction of $\mathsf{pres}(\varphi)$ to $\mathsf{pos}(\varphi) \vee \mathsf{neg}(\varphi)$. These presentations are related in section 5 to the broader conception of discourse interpretation as a "dynamic" construction of contexts, rather than a

[1] That is, the object of the present paper is to supply computational foundations (in the sense described in section 2) for CCP models of presuppositions, and not (so much) to expand their linguistic coverage (insofar as such coverage is deemed independent of computational considerations).

"static" evaluation of truth against a fixed context (a conception that, in turn, supports an anaphoric analysis of presuppositions; e.g., van der Sandt [**18**]).

2. Computing context change potentials

To start out, let us fix a signature (or vocabulary) L, a countable set *Var* of variables, and a sufficiently large set $\mathcal{M}$ of L-models (e.g., those whose universe is a set of ordinals less than the cardinality of the set of L-formulas). Let WSP be the set of wsp's (M, f) where $M \in \mathcal{M}$ and f is a partial function from *Var* to the universe of M. Let us also fix a set Φ of formulas — for example (but not necessarily) DRS's — that will be interpreted by binary relations called CCP's. To avoid confusing the different CCP's that will be considered, let us write $P(\varphi)$ for the CCP of $\varphi \in \Phi$ relative to wsp's,

$$P(\varphi) \quad \subseteq \quad \mathsf{WSP} \times \mathsf{WSP} \ .$$

Let us assume also that

$$dom(P(\varphi)) \quad = \quad \{(M, f) \in \mathsf{WSP} \ : \ \varphi \text{ is satisfiable in } (M, f)\} \ ,$$

for some notion of "satisfiability" that can be linked, for example, to Tarski's definition of satisfaction $\models$ as follows. A first-order L-formula χ is a *characteristic formula of* φ if for every wsp (M, f) and every L-model M_0,

$$\begin{array}{lll} (M_0, \emptyset) \ P(\varphi) \ (M, f) & \text{iff} & M = M_0 \text{ and} \\ & & dom(f) = FVAR(\chi) \text{ and} \\ & & M \models \chi[f] \end{array}$$

where $FVAR(\chi)$ is the set of free variables of χ. Now, to try out the computational demands of P, let $L = \{\dot{+}, \dot{\times}, \dot{0}, \dot{1}\}$ be a language for arithmetic, and $\mathbf{N}$ be its standard model $\langle \{0, 1, 2, \dots\}, +, \times, 0, 1\rangle$. Note that every first-order L-sentence (without a free variable) is the characteristic formula of some DRS φ.[2] It follows immediately that for the collection Φ of DRSs, the set

$$\{\varphi \in \Phi \ : \ (\mathbf{N}, \emptyset) \in dom(P(\varphi))\}$$

is non-arithmetic — i.e., computationally hopeless. That is to say, the notion from *Quantified Dynamic Logic* (see, e.g., Harel [**10**], where particular attention is given to the model $\mathbf{N}$) of computing over a fixed first-order model cannot be transferred to DRT/DPL. The culprit is the DRT/DPL notion of negation $\neg$, which is interpreted by P as a test for the complement of the halting problem

$$P(\neg\varphi) \quad = \quad \{(s, s) \ : \ s \in \mathsf{WSP} - dom(P(\varphi))\} \ .$$

(In dynamic logic, $\neg\varphi$ can be equated with $[\varphi]\bot?$, where $[\varphi]$ is the box modality labeled by φ, and $\bot$ is some unsatisfiable statement; but to say that $[\varphi]\bot?$ is a program is to deny that every program can be computed.)

[2] This is worked out in detail in Fernando [**4**], which expands considerably on the present section, disputing widespread claims that dynamic semantics is "computational" because its formulas are programs [sic] of dynamic logic.

To interpret the formulas in a computationally meaningful manner, we must pass from a fixed (albeit arbitrary) choice of a model to a sufficiently large collection of models, moving from a computationally intractable notion of satisfaction (relative say to **N**) to a notion of semantic validity (involving say countable models of Peano arithmetic) amenable (by Gödel's completeness theorem) to syntactic manipulations. This passage is effected by the classic "subset construction" from automata theory (e.g., Hopcroft and Ullman [**12**]) that determinizes non-deterministic (finite) automata. An operation, call it D, can be extracted from that construction that (more generally) turns a binary relation R on any set S to a function $D[R]$ defined on subsets σ of S by

$$D[R](\sigma) \quad = \quad \{s' \ : \ (\exists s \in \sigma) \ sRs'\} \ .$$

Rather than studying arbitrary sets of wsp's, let us concentrate on certain "accessible" sets, starting with the *initial* information state

$$\sigma_\circ \quad = \quad \{(M, \emptyset) \ : \ M \in \mathcal{M}\} \ .$$

Observe that $\sigma_\circ$ has zero information content insofar as *no* model in $\mathcal{M}$ has been eliminated, and *no* variable has been initialized. The set *Acc* of *accessible* information states is then the smallest set Σ of information states including $\sigma_\circ$, and closed under the rule

$$\frac{\sigma \in \Sigma}{D[P(\varphi)](\sigma) \in \Sigma} \quad \varphi \in \Phi \ , \ D[P(\varphi)](\sigma) \neq \emptyset \ .$$

Having quantified out the choice of a model in an an accessible information state, we are now in a position to characterize, in purely syntactic terms, the restriction of $D[P(\varphi)]$ to *Acc*, for every $\varphi \in \Phi$. Let $\Phi_\perp$ be a set of formulas that are said to be *absurd*, and let & be a binary connective on Φ that *merges* formulas. For every $\varphi \in \Phi$, define the CCP $\xrightarrow{\varphi} \subseteq \Phi \times \Phi$ by

$$\psi \xrightarrow{\varphi} \psi' \quad \text{iff} \quad \psi' = \psi \& \varphi \text{ and } \psi' \notin \Phi_\perp \ .$$

(Note that each CCP $\xrightarrow{\varphi}$ is a partial function.) Call a formula φ *trace equivalent* to an information state σ, denoted $\varphi \sim \sigma$, if for every finite sequence $\varphi_1, \ldots, \varphi_n$ of formulas,

$$\begin{array}{lll} \varphi \in dom(\xrightarrow{\varphi_1} \circ \cdots \circ \xrightarrow{\varphi_n}) & \text{iff} & D[P(\varphi_n)](\cdots(D[P(\varphi_1)](\sigma))\cdots) \\ & & \neq \emptyset \ . \end{array}$$

Also, for every $\varphi \in \Phi$, let σ_φ abbreviate $D[P(\varphi)](\sigma_\circ)$.

Equivalence Lemma. *Assume that*

$$\Phi_\perp \quad = \quad \{\varphi \in \Phi \ : \ \sigma_\varphi = \emptyset\} \ ,$$

and that P interprets & as relational composition $\circ$ — for all $\varphi, \varphi' \in \Phi$,

$$P(\varphi \& \varphi') \quad = \quad P(\varphi) \circ P(\varphi') \ .$$

Then for every $\varphi \in \Phi$, $\varphi \sim \sigma_\varphi$.

Because $\xrightarrow{\varphi}$ and $D[P(\varphi)]$ (restricted to non-empty information states) are deterministic relations (for every $\varphi \in \Phi$), trace equivalence is the same as the equivalence induced (maximally) by the notion of a *bisimulation* (Park [**16**]). This is of some interest in that, in general, the former is commonly accepted as the coarsest notion of equivalence on states, whereas the latter is taken to be the finest. For both notions of equivalence, one must be careful about transformations that produce transitions when before there were none — e.g., $D[R]$-transitions to $\emptyset$. Hence, $D[P(\varphi)]$ is restricted to non-empty information states in the definition of trace equivalence.

Confining our attention to states in the CCP's $\xrightarrow{\varphi}$, define an equivalence relation $\underline{\leftrightarrow}$ on Φ by

$$\varphi \,\underline{\leftrightarrow}\, \psi \quad \text{iff} \quad (\forall \varphi' \in \Phi)\ (\varphi \& \varphi' \in \Phi_\perp \text{ iff } \psi \& \varphi' \in \Phi_\perp)$$

(which corresponds to bisimulation equivalence over the $\xrightarrow{\varphi}$'s).

Equivalence Theorem. *If, in addition to the assumptions from the previous lemma, every formula has a characteristic formula, then not only is $\varphi \sim \sigma_\varphi$ for every $\varphi \in \Phi$, but also*

(i) *for all φ and $\varphi' \in \Phi$,*

$$\varphi \,\underline{\leftrightarrow}\, \varphi' \quad \textit{iff} \quad \sigma_\varphi = \sigma_{\varphi'} \ ,$$

and

(ii) *both $\underline{\leftrightarrow}$ and $\Phi_\perp$ are recursively enumerable.*

Part (i) of the theorem says that the restriction of $D[P(\varphi)]$'s to non-empty accessible information states is a *strongly extensional quotient* of the CCP's $\xrightarrow{\varphi}$ (meaning roughly that it abstracts out all syntactic details in $\xrightarrow{\varphi}$). Formulating an l-transition from φ to ψ as a triple (φ, l, ψ), part (ii) provides a recursively enumerable reformulation of $\xrightarrow{\varphi}$-transitions as the set

$$\{(\psi, \varphi, \psi \& \varphi) \ : \ \psi, \varphi \in \Phi\} \ \cup \ \{(\varphi, \perp, \varphi) \ : \ \varphi \in \Phi_\perp\}$$

of transitions, bisimulation equivalence over which is given (also) by $\underline{\leftrightarrow}$. In other words, choosing contexts (on which CCP's operate) to be information states, as opposed to formulas, involves a trade-off between the canonicity specified by part (i), and the recursive enumerability in part (ii). Part (ii) is proved by appealing to Gödel's completeness theorem to conclude that

$$\begin{array}{lll} \varphi \,\underline{\leftrightarrow}\, \varphi' & \text{iff} & \vdash (\forall \overline{x})\ \chi_\varphi \equiv \chi_{\varphi'} \\ \varphi \in \Phi_\perp & \text{iff} & \vdash \neg\exists\overline{x}\chi_\varphi \end{array}$$

for all φ and $\varphi' \in \Phi$, with characteristic formulas χ_φ and $\chi_{\varphi'}$, respectively.

3. Translating transitions and anti-transitions

Turning to presuppositions, let us consider the notion that the presuppositions $\mathsf{pres}(\varphi)$ of a formula φ are determined by the domain of its CCP

$[\![\varphi]\!]$

(2) $$\mathsf{pres}(\varphi) \quad \approx \quad dom([\![\varphi]\!]) \ .$$

Now what could $[\![\varphi]\!]$ be? Certainly, it cannot be $P(\varphi)$, as the domain of $P(\varphi)$ is too small, consisting only of wsp's at which φ is satisfiable. Intuitively, the presuppositions of φ can be met also at states where φ is unsatisfiable. Applying the subset construction D to $P(\varphi)$, it is tempting to say that the presuppositions of φ can hold at an information state σ for which $D[P(\varphi)](\sigma) = \emptyset$. Of course, the problem now in identifying $D[P(\varphi)]$ with $[\![\varphi]\!]$ in (2) is that the domain of $D[P(\varphi)]$ is too big, being (as it is) the full power set of WSP. Accordingly, Heim [**11**] and Beaver [**1**] revise $D[P(\varphi)]$ into a partial function, thereby allowing for presupposition failure, but also leaving unexplained the intuition (on which they build) that

an information state is the set of wsp's possible under it

(as partiality blocks the reduction via D of CCP's to transitions between wsp's). Be that as it may, note that a transition from an information state σ to the empty set is a "negative" transition inasmuch as it arises from the failure of any wsp in σ to return an output — failures that can be put positively as successful applications of $\neg$ to the wsp's in σ. Of course, this cannot be the whole story of transitions between wsp's, as an explanation of presupposition failure is missing. Enter van Eijck [**3**]'s error state ϵ, from which presumably the Heim/Beaver CCP's between information states can be reconstructed along the lines of D. I have yet to see such a reconstruction worked out rigorously, nor am I particularly keen about carrying it out myself, as I do not see that much can be gained in explaining an error through an error. (Are errors meant to be introduced or to be eliminated?) It is far preferable, I believe, to provide an analysis of presupposition errors with*out* building an error state into the stock of semantic primitives, retaining instead the distinction (going back to Keenan's forementioned insight) between *what* information is presented and *how* it is conveyed.

Toward that end, I propose that the characterization (2) of $\mathsf{pres}(\varphi)$ be revised to

$$\mathsf{pres}(\varphi) \quad \approx \quad \mathsf{pos}(\varphi) \vee \mathsf{neg}(\varphi)$$

for certain functions pos and neg on formulas that are subjected to the usual semantics P (which, in particular, interprets disjunction $\vee$ as

$$P(\psi \vee \psi') \quad = \quad \{(s,s) \ : \ s \in dom(P(\psi) \cup P(\psi'))\}$$

for all ψ and $\psi' \in \Phi$).[3] The translations pos and neg (not to be confused with $\neg$) are not meant to be carved into stone in the same way that P (or $\neg$) is, although the reader is certainly free to do as she pleases. Putting off to

[3] Whereas the introduction of a translation neg has been motivated above by the need to consider "negative" transitions, the appearance of pos (at this point) may seem a bit mysterious. Suffice it to say that for reasons of symmetry that should become clear in the example below, it is useful to have a translation pos that differs from the identity function.

the next section what exactly I mean by not carving pos and neg into stone, let me concede, for now, that she can pass to the transition/anti-transition pairs $[\![\varphi]\!]^+, [\![\varphi]\!]^-$ of Krahmer [**15**], by composing the semantics P with the translations pos and neg, respectively. That is to say, she can ride roughshod over the what/how (semantics/syntax) distinction that I have taken pains to maintain, by semanticizing the whole business. Evidently, there is no point in being too dogmatic about a distinction that is so easily obliterated. On the other hand, the matter can be turned on its head by asking if transition/anti-transition pairs $[\![\varphi]\!]^+, [\![\varphi]\!]^-$ admit a decomposition as $P(\mathsf{pos}(\varphi)), P(\mathsf{neg}(\varphi))$.

With this question in mind, note that the following (semantic) operations on pairs (a, b) are basic to the conception of the meaning of a formula as a positive/negative pair

$$\begin{aligned} \mathsf{POS}(a,b) &= a \\ \mathsf{NEG}(a,b) &= b \\ \sim(a,b) &= (b,a) \ . \end{aligned}$$

The present proposal is to formulate these operations on $sem(\varphi) = ([\![\varphi]\!]^+, [\![\varphi]\!]^-)$ through translations pos, neg, and $-$ (respectively) such that

$$\begin{aligned} \mathsf{POS}(sem(\varphi)) &= P(\mathsf{pos}(\varphi)) \\ \mathsf{NEG}(sem(\varphi)) &= P(\mathsf{neg}(\varphi)) \\ \sim sem(\varphi) &= sem(-\varphi) \ , \end{aligned}$$

preserving the DRT/DPL system P of CCP's. Only the translations pos and $-$ need be specified, if it is assumed (as it will) that neg is the composition of the two translations

$$\mathsf{neg}(\varphi) \quad = \quad \mathsf{pos}(-\varphi) \ .$$

The literal (i.e., syntactic) equality

$$--\varphi \quad = \quad \varphi$$

can then be arranged by defining a function $\hat{\cdot}$ mapping n-ary relation symbols R to n-ary relation symbols $\hat{R}$, and n-ary logical connectives α to n-ary logical connectives $\hat{\alpha}$ such that

$$\hat{\hat{R}} \quad = \quad R$$

and

$$\hat{\hat{\alpha}} \quad = \quad \alpha \ .$$

To construct such a map $\hat{\cdot}$, it may be necessary to throw in fresh relation symbols and logical connectives, and stipulate that

$$-R(\bar{t}) \quad = \quad \hat{R}(\bar{t})$$

and

$$-\alpha(\varphi_1, \ldots, \varphi_n) \quad = \quad \hat{\alpha}(\tilde{\varphi}_1, \ldots, \tilde{\varphi}_n) \ ,$$

where $\tilde{\varphi}_i$ is either φ_i or $-\varphi_i$ (simplifying the notation somewhat, by suppressing the subscripts α and i on the tilde $\tilde{\cdot}$). The choice between setting $\tilde{\varphi}_i$ to φ_i or to $-\varphi_i$ depends on whether the ith argument occurs "negatively" or "positively," as will be illustrated presently.

Example. A familiar practice in 3-valued logic is to stipulate that conjunction $\wedge$ and disjunction $\vee$ form a dual pair according to De Morgan's laws

$$\begin{aligned} -(\varphi \wedge \psi) &= -\varphi \ \vee \ -\psi \\ -(\varphi \vee \psi) &= -\varphi \ \wedge \ -\psi \ . \end{aligned}$$

Note that "dynamic" conjunction & (interpreted by relational composition) should be distinguished from $\wedge$, for which let us agree that

$$\mathsf{pos}(\varphi \wedge \psi) \quad = \quad \neg(\neg\mathsf{pos}(\varphi) \vee \neg\mathsf{pos}(\psi)) \ .$$

Assuming that P interprets implication $\supset$ by[4]

$$\begin{aligned} P(\varphi \supset \psi) \;=\; \{(s,s) \ : \ & s \in \mathsf{WSP} \text{ and} \\ & \{s' : s \ P(\varphi) \ s'\} \subseteq dom(P(\psi))\} \ , \end{aligned}$$

define

$$\begin{aligned} -(\varphi \& \psi) &= \varphi \ \supset \ -\psi \\ -(\varphi \supset \psi) &= \varphi \ \& \ -\psi \end{aligned}$$

(noting that in $\varphi \supset \psi$, φ occurs negatively, while ψ occurs positively), and

$$\begin{aligned} \mathsf{pos}(\varphi \& \psi) &= \mathsf{pos}(\varphi) \ \& \ \mathsf{pos}(\psi) \\ \mathsf{pos}(\varphi \vee \psi) &= \mathsf{pos}(\varphi) \ \vee \ \mathsf{pos}(\psi) \\ \mathsf{pos}(\varphi \supset \psi) &= \mathsf{pres}(\varphi) \ \& \ (\mathsf{pos}(\varphi) \supset \mathsf{pos}(\psi)) \ . \end{aligned}$$

The preceding is largely consistent with section 6 of Krahmer [**15**] (devised in collaboration with R. Muskens), which also treats random assignments $x :=?$, a form of negation, and a certain presupposition construct $\varphi_{\langle\pi\rangle}$, all of which can be brought within the present framework.[5]

---

[4]The reader familiar with dynamic logic might regard formulas as [sic] programs and rewrite

$$\begin{array}{ccc} \varphi \& \psi & \text{as} & \varphi ; \psi \\ \varphi \vee \psi & \text{as} & \neg\neg(\varphi + \psi) \\ \varphi \supset \psi & \text{as} & \neg(\varphi ; \neg\psi) \end{array}$$

where, as previously noted, $\neg\varphi$ is the test $[\varphi]\bot?$. Although the connectives $\vee$ and $\supset$ can be defined from & and $\neg$ (which can, in turn, be given as $\cdot \supset \bot$), certain "dualities" that are presently of interest are best brought out by taking $\vee$ and $\supset$ also to be primitive.

[5]A minor point of departure is that $[\![\varphi \supset \psi]\!]^-$ in section 6 of Krahmer [**15**] is the static projection $\neg\neg\mathsf{neg}(\varphi \supset \psi)$ of the present interpretation. Otherwise, put in terms of our present approach, the proposal there is that

$$\begin{array}{rcll} \mathsf{pos}(A) & = & A & \text{for atomic formulas } A \\ \mathsf{pos}(x :=?) & = & \exists x \ x = x & \\ \mathsf{pos}(\varphi_{\langle\pi\rangle}) & = & \mathsf{pos}(\pi) \ \& \ \mathsf{pos}(\varphi) \ . & \end{array}$$

4. Choosing translations: a pragmatic turn

Characterizing the presuppositions $\mathsf{pres}(\varphi)$ of a formula φ as $\mathsf{pos}(\varphi) \vee \mathsf{neg}(\varphi)$, for translations pos and neg obtained by extending the extension/anti-extension pairs familiar from 3-valued logic to logical connectives may well strike the reader as an exercise in a *semantic* approach to presuppositions. (A "semantic" approach is understood here to be one based on Strawson's reduction of presuppositions to conditions for truth or falsity — i.e., *bivalence*.) This impression is *in*correct on two counts.

(i) The semantic values assigned to formulas are *not* truth values. Indeed, reference to truth has, up to this point, been carefully avoided (except for the use of the term "true" in the introduction, which was quickly changed to "satisfiable").

(ii) Were I compelled to describe the approach to presuppositions I intend to push as either semantic or pragmatic, I would choose the latter label.

Let us take up these two points in turn.

With regard to (i), note that although ideas from 3-valued logic are employed in the previous section, the following assignment of truth values need not yield a 3-valued logic

$$\begin{array}{rcl} \varphi \text{ is TRUE at } s & \text{iff} & s \in dom(P(\mathsf{pos}(\varphi))) \\ \varphi \text{ is FALSE at } s & \text{iff} & s \in dom(P(\mathsf{neg}(\varphi))) \\ \varphi \text{ is UNDEFINED at } s & & \text{otherwise .} \end{array}$$

Not only does this definition fail to rule out the possibility of a fourth truth value (in case φ is both TRUE and FALSE at s), it can be ill-behaved even over the two ordinary truth values, as illustrated by the following case in DPL. Let M be a first-order model that interprets a unary relation symbol R by a disjoint extension/anti-extension pair such that neither set is empty, and let f be an embedding into M that maps the variable x to the anti-extension of R. Then $R(x)$ is FALSE at f, $\exists x\ R(x)$ is TRUE at f, but $(\exists x\ R(x))\ \&\ R(x)$ is TRUE at f, which is to say that a FALSE formula can become TRUE if (pre-)conjoined with a TRUE formula. (Of course, there will also be cases where a

---

Krahmer's negation can be defined away through the translation $-$ by introducing constructs dual to random assignment

$$\begin{array}{rcl} -(x :=?) & = & ? =: x \\ -(? =: x) & = & x :=? \ , \end{array}$$

and to the presupposition operator

$$\begin{array}{rcl} -(\varphi_{\langle\pi\rangle}) & = & {}_{[\pi]} - \varphi \\ -({}_{[\pi]}\varphi) & = & (-\varphi)_{\langle\pi\rangle} \ , \end{array}$$

and subject to

$$\begin{array}{rcl} \mathsf{pos}(? =: x) & = & \perp \qquad (\text{where } P(\perp) = \emptyset) \\ \mathsf{pos}({}_{[\pi]}\varphi) & = & (\mathsf{pos}(\pi) \supset \mathsf{pos}(\varphi))\ \&\ \mathsf{pos}(\pi)\ \&\ \mathsf{pos}(\varphi) \ . \end{array}$$

TRUE formula conjoined with a FALSE formula is FALSE.) The simple lesson here is that formulas are interpreted semantically as binary relations, and some thought needs to be invested to derive a natural notion of truth from a relational interpretation. We will take up this challenge in the next section, but first, onto point (ii) and the pragmatic character of presuppositions.

I have already mentioned that the translations pos and neg should not be accorded the same status as P or $\neg$. Following up on Keenan's insight, P and $\neg$ concern *what* information is presented, whereas pos and neg have to do with *how* it is conveyed. True, this distinction can be rubbed out by the compositions $P \circ \mathsf{pos}$ and $P \circ \mathsf{neg}$, semanticizing everything to transition/anti-transition pairs. But I fail to see what is achieved by mixing up *what* with *how*, particularly when firm answers can be supplied to the former, by taking a softer stance toward the latter. Take, for example, a conditional *if A then B*. It is not at all clear how to characterize its presuppositions uniformly for all A's and B's. Although it is commonly taken for granted in semantic approaches to presupposition (including dynamic ones) that the conditional should inherit the presuppositions of its antecedent A, there are troublesome sentences such as

```
If Erwin realizes the cat is dead,
      then the cat is dead
```

that would suggest otherwise.[6] And the situation does not get any easier with the presuppositions of the consequent B, the study of which branches off to "strong" or "weak" presuppositions (e.g., Krahmer [**15**]), and "global" or "local" or "intermediate accomodation" (e.g., van der Sandt [**18**]).

Conceptually, the problem is that, as Karttunen [**14**] puts it, "ordinary discourse is not always fully explicit"

> People do make leaps and shortcuts by using sentences whose presuppositions are not satisfied in the conversational *context*. This is the rule rather than the exception, and we should not base our notion of presupposition on the false premiss that it does not or should not happen. But granting that *ordinary discourse is not always fully explicit* in the above sense, I think we can maintain that a *sentence is always taken to be an increment to a context that satisfies its presuppositions.* [p. 191, italics added]

That is, a natural language expression is often ambiguous between different formulations. The most plausible formulation may depend on any number of considerations, including the "conversational context." Now, a discourse-based account of natural language, such as DRT, differs from a purely sentence-based one by incorporating the conversational context within the representation of the discourse. In particular, building on the system of CCP's $\xrightarrow{\varphi}$ described in section 2, the conversational context can be identified with a

[6] I am indebted to Robert van Rooy for bringing such examples (well-known to proponents of pragmatic approaches to presupposition) to my attention.

formula that not only can be fed as an input to a CCP, but itself defines a CCP (making it both computationally amenable and computationally active). Thus, Beaver [**1**]'s picture (intended to model global accomodation) of a single formula that takes as input a preference ordering on information states can be "dualized" into a picture of a single input fed to a preference ordering on formulas. The latter preference ordering reflects ambiguity, and gives rise, in turn, to the sort of inputs Beaver considers. Moreover, for some DRS's φ and φ', merging & is commutative

$$\varphi \mathbin{\&} \varphi' \quad = \quad \varphi' \mathbin{\&} \varphi \ .$$

To be more precise, a DRS can be understood as a pair (U, C) consisting of a set U of variables (or discourse markers) and a set C of *conditions* (that are generated simultaneoulsy with DRS's). The CCP $P(U, C)$ of a DRS (U, C) is given on wsp's (M, f) and (N, g) by

$$\begin{array}{lll} (M,f)\ P(U,C)\ (N,g) & \text{iff} & M = N \text{ and } f \subseteq g \text{ and} \\ & & dom(g) = dom(f) \cup U \text{ and} \\ & & \text{each } \gamma \in C \text{ is verified by } (M,g)\ , \end{array}$$

for some notion of *verification* defined simultaneously with P. Now, the point is that, modulo pesky variable conflicts that can often be overcome,[7]

$$(3) \qquad P(U,C) \circ P(U',C') \quad \approx \quad P(U \cup U', C \cup C')\ .$$

In cases where (3) can be established, the commutativity of $\cup$ allows the input DRS (U, C) in (3) to be swapped with the updating DRS (U', C').

5. Dynamics and disjunction reconsidered

Looking more closely at Karttunen [**14**]'s claim that a statement can always be "taken to be an increment to a context that satisfies its presuppositions," two dimensions of information growth can be discerned in the CCP of a DRS (U, C):

(i) an *expansive* potential given by U (initializing variables in U), and
(ii) an *eliminative* potential given by C (eliminating contexts that fail to verify a condition in C).

Now, suppose we decomposed the positive transitions $\mathsf{pos}(\varphi)$ of φ into a presuppositional component $\mathsf{pres}(\varphi)$ followed by the remnant of $\mathsf{pos}(\varphi)$, call it $\mathsf{as}(\varphi)$ (for assertion),

$$\mathsf{pos}(\varphi) \quad = \quad \mathsf{pres}(\varphi) \mathbin{\&} \mathsf{as}(\varphi)\ .$$

Assuming $\mathsf{pos}(\varphi)$ is the DRS (U, C), a simple-minded parallel to the normal form

$$P(U,C) \quad = \quad P(U,\emptyset) \circ P(\emptyset, C)$$

[7] A crucial advantage here of DRT over DPL is that in DRT a variable x is assigned a random value iff x is uninitialized.

would shove all expansive growth $P(U,\emptyset)$ to the presuppositional component $\mathsf{pres}(\varphi)$, leaving the non-presuppositional component $\mathsf{as}(\varphi)$ a purely eliminative character (i.e., making Karttunen [**14**]'s "increment to a context that satisfies its presuppositions" exclusively eliminative). In such a case, $P(\neg\neg\mathsf{as}(\varphi)) = P(\mathsf{as}(\varphi))$ and we might expect to reduce the negation translation neg to DRT/DPL $\neg$ as follows

$$\mathsf{neg}(\varphi) \quad = \quad \mathsf{pres}(\varphi) \ \& \ \neg\mathsf{as}(\varphi) \ .$$

But if $\mathsf{pres}(\varphi)$ is to have expansive potential, the characterization $\mathsf{pres}(\varphi) = \mathsf{pos}(\varphi) \vee \mathsf{neg}(\varphi)$ must be modified, as a disjunction $\psi \vee \psi'$ can never expand a wsp:

$$s \ P(\psi \vee \psi') \ s' \quad \text{implies} \quad s = s' \ .$$

A simple solution is to replace $\vee$ by non-deterministic choice $+$, interpreted as union,

$$s \ P(\psi + \psi') \ s' \quad \text{iff} \quad s \ P(\psi) \ s' \text{ or } s \ P(\psi') \ s' \ ,$$

reformulating $\mathsf{pres}(\varphi)$ as $\mathsf{pos}(\varphi) + \mathsf{neg}(\varphi)$, and setting $\mathsf{as}(\varphi)$ to $\mathsf{pres}(\varphi) \supset \mathsf{pos}(\varphi)$. This time, however, $\mathsf{pres}(\varphi)$ goes too far, trivializing the non-presuppositional component $\mathsf{as}(\varphi)$. The appropriate disjunction connecting $\mathsf{pos}(\varphi)$ with $\mathsf{neg}(\varphi)$ to yield $\mathsf{pres}(\varphi)$ probably lies somewhere between $\vee$ and $+$.

The decomposition of $P(U,C)$ into an expansive component $P(U,\emptyset)$ followed by an eliminative part $P(\emptyset,C)$ runs parallel also to a Henkin-style proof of Gödel's completeness theorem, in which a theory is *expanded* with Henkin witnesses, before it is turned *eliminatively* to a maximally consistent set. This suggests a view of discourse interpretation as a construction of contexts, with truth derived from change by the equivalence

(4) $\qquad \varphi$ is true at s iff φ holds at every completion of s .

The equivalence (4) is bound to remind the reader familiar with van Fraassen [**7**] of *supervaluations*. The approach I have in mind, however, concentrates on gaps arising from uninitialized variables, rather than those between non-exhaustive extensions and anti-extensions (of, for example, presuppositional verbs such as `regret` — which arguably ought not to be closed).[8] To establish (4), it is not enough to equate the completion of a wsp (M,f) with some wsp (M,g), where g is an arbitrary total function extending f. All existential truths in M must be witnessed by g, and, to take an extreme case, a constant function g would witness very few such truths. In the terminology of Boolean-valued semantics, a first-order model cannot be formed simply from an ultrafilter over a Lindenbaum algebra of first-order formulas; the ultrafilter must be *generic*. For (M,g) to be a completion of (M,f), g should not only

[8] Although I cannot claim to understand the matter fully, it seems to me that the move made in this section from, on the one hand, $\mathsf{pres}(\varphi) = \mathsf{pos}(\varphi) \vee \mathsf{neg}(\varphi)$ and extension/anti-extension pairs to, on the other hand, $\mathsf{pos}(\varphi) = \mathsf{pres}(\varphi) \ \& \ \mathsf{as}(\varphi)$ and anaphoric/existential complications is very much in the spirit of the 2-dimensional approach to presuppositions described in van Rooy [**17**].

extend f, but also be defined, when possible, on variables with conditions attached to them. Such conditions make these variables similar to Henkin witnesses, or to Hilbert's ϵ-terms. A basic difference here, however, is that these variables must remain uninitialized if the conditions cannot be met. In this respect, they are closer to restricted parameters of situation theory (e.g., Cooper [**2**]). The precise details are somewhat involved, and form, on their own, an entire paper, the results of which I have space only to summarize. Relative to the partial order $\sqsubseteq$ given on wsp's (M, f) and (N, g) by

$$(M, f) \sqsubseteq (N, g) \quad \text{iff} \quad M = N \text{ and } f \subseteq g \ ,$$

truth is defined by a so-called double intuitionistic negation translation

$$\varphi \text{ is } \textit{true} \text{ at } s \quad \text{iff} \quad (\forall s' \sqsupseteq s)(\exists s'' \sqsupseteq s')\ s'' \in \mathit{dom}(P(\varphi)) \ .$$

In addition to making (4) precise, Fernando [**5**] establishes the following results.

(i) Recalling that *acceptance* in Veltman [**19**] is given by the set $\mathit{fix}(P(\varphi))$ of fixed points of $P(\varphi)$, whereas *acceptability* amounts to $\mathit{dom}(P(\varphi))$, the double negation translation makes acceptance acceptable insofar as

$$(\forall s' \sqsupseteq s)(\exists s'' \sqsupseteq s')\ s'' \in \mathit{fix}(P(\varphi)) \quad \text{iff}$$
$$(\forall s' \sqsupseteq s)(\exists s'' \sqsupseteq s')\ s'' \in \mathit{dom}(P(\varphi)) \ .$$

(ii) The resulting notion of truth is not only compositional, but also Boolean-valued.

(iii) The double intuitionistic negation translation can be applied to information states, under the so-called *Smyth pre-order* $\sqsubseteq_{Sm}$,

$$U \sqsubseteq_{Sm} U' \quad \text{iff} \quad (\forall s' \in U')(\exists s \in U)\ s \sqsubseteq s' \ ,$$

so that

$$(\forall U' \sqsupseteq_{Sm} U)(\exists U'' \sqsupseteq_{Sm} U')\ U'' \in \mathit{dom}(D[P(\varphi)]) \quad \text{iff}$$
$$(\forall s \in U)\ \varphi \text{ is true at } s \ .$$

6. Conclusion

Having equated the presuppositions $\mathsf{pres}(\varphi)$ of a formula φ with a disjunction $\mathsf{pos}(\varphi)$ **or** $\mathsf{neg}(\varphi)$, we have left open, for pragmatic analysis, just what pos, neg and **or** are. The following (minimal) assumptions have been studied above

$$\begin{aligned} \mathsf{neg}(\varphi) &= \mathsf{pos}(-\varphi) \\ \mathit{dom}(P(\varphi \textbf{ or } \psi)) &= \mathit{dom}(P(\varphi) \cup P(\psi)) \ , \end{aligned}$$

where $-$ is a function swapping extensions with anti-extensions, and P is the DRT/DPL semantic map interpreting a formula φ as a CCP $P(\varphi)$ between world-sequence pairs. The problem of computing $P(\varphi)$ was taken up in section 2, while guidelines for defining $-$ were suggested in section 3. The last two sections examined notions of truth and context, complementing the reduction of $\mathsf{pres}(\varphi)$ to $\mathsf{pos}(\varphi)$ **or** $\mathsf{neg}(\varphi)$ with a decomposition of $\mathsf{pos}(\varphi)$ into presuppositional and non-presuppositional components, $\mathsf{pres}(\varphi)$ & $\mathsf{as}(\varphi)$. A comparison of that decomposition with the initialize-and-then-test interpretation of a DRS suggests a "dynamic" re-interpretation of **or**, giving $\mathsf{pres}(\varphi)$ expansive potential (i.e., allowing $P(\mathsf{pres}(\varphi))$ to return an output different from its input). In particular, if the non-presuppositional remnant $\mathsf{as}(\varphi)$ is purely eliminative — i.e., $P(\neg\neg\mathsf{as}(\varphi)) = P(\mathsf{as}(\varphi))$ —, then we can expect to reduce $\mathsf{neg}(\varphi)$ to $\mathsf{pres}(\varphi)$ & $\neg\mathsf{as}(\varphi)$. But, as with pos and neg, it is not at all clear what *the* right notion of disjunction is. This is hardly surprising, given the various notions of presuppositions and proposals to accomodate them.

Analyzing presuppositions is no doubt a huge problem, a basic question concerning which is how to divide labor on it in a fruitful manner. Rather than espouse a single notion of presupposition, different notions can be formulated above through pragmatic translations between formulas that are eventually given a DRT/DPL semantics P. The scope of this semantics can be expanded by introducing abstract entities into the worlds. Examples of such entities include proposition-like objects of `regret`, an "external" analysis of which (Veltman [**19**]) is linked to presuppositions in Beaver [**1**]. I close with some words about these developments (i.e., Veltman [**19**], Groenendijk, Stokhof and Veltman [**9**], Beaver [**1**]), as these pose serious problems for the computational foundations supplied above by questioning the adequacy of CCP's $P(\varphi)$ defined on wsp's for analyzing $\mathsf{might}\ \varphi$ and presuppositions. Squaring the intuition $\mathsf{pres}(\varphi) \approx dom([\![\varphi]\!])$ with a semantics based on P has already been taken up in section 3 (the trouble with existential quantification that Heim [**11**] has with "a fat man pushing his bicycle" being an artifact of defining CCP's on information states, rather than on wsp's). As for $\mathsf{might}\ \varphi$, the solution I propose is to introduce formulas as objects in models — providing, as it were, an *internal* notion of formula, in contrast to the *external* (metatheoretic) formulas on which Veltman [**19**] defines the semantics of $\mathsf{might}\ \varphi$. A fixed distinguished variable $\hat{x}$ ranging over internal formulas is designated to keep track of updates, using programs of the form[9]

$$\begin{aligned} \mathtt{record}(\varphi) \quad = \quad & \textit{if } \hat{x} \textit{ is uninitialized then } \hat{x} := \varphi \\ & \textit{else } \hat{x} := \hat{x}\&\varphi \ . \end{aligned}$$

The translation pos is defined in the obvious way by inserting $\mathtt{record}(\varphi)$, the crucial clause being for formulas $\mathsf{might}\ \varphi$, which are analyzed through a consistency check on $\hat{x}$. Adding a unary relation symbol Con to the underlying signature (or vocabulary) of the models in wsp's, set

$$\mathsf{pos}(\mathsf{might}\ \varphi) \quad = \quad \neg\neg(\mathtt{record}(\varphi)\ \&\ \mathsf{Con}(\hat{x}))$$

[9]That is,

$$P(\mathtt{record}(\varphi)) \quad = \quad \{((M,f),(M,f\&\varphi)) \ : \ (M,f) \in \mathsf{WSP}\}$$

with $f\&\varphi$ equal to f except on $\hat{x}$, where

$$f\&\varphi(\hat{x}) \ = \ \begin{cases} \varphi & \text{if } \hat{x} \notin dom(f) \\ f(\hat{x})\&\varphi & \text{otherwise.} \end{cases}$$

where the double DRT/DPL negation prefix suppresses the dynamic effects of $\texttt{record}(\varphi)$. The translation neg is defined as before to be the composition of pos with $-$, where $-$ is extended to $\mathsf{might}\ \varphi$ by introducing a dual connective must

$$\begin{aligned} -\mathsf{might}\ \varphi &= \mathsf{must}(-\varphi) \\ -\mathsf{must}\ \varphi &= \mathsf{might}(-\varphi)\ , \end{aligned}$$

with $\mathsf{pos}(\mathsf{must}\ \varphi)$ obtained by pushing one of the negations in $\mathsf{pos}(\mathsf{might}\ \varphi)$ through $\texttt{record}(\varphi)$

$$\mathsf{pos}(\mathsf{must}\ \varphi) \ = \ \neg(\texttt{record}(-\varphi)\ \&\ \mathsf{Con}(\hat{x}))\ .$$

Now, returning to the predicate Con behind might, one can show that if we insist that Con always be interpreted in the external sense corresponding to Veltman [**19**], then the set $\Phi_\perp$ of absurd formulas (as defined in section 2) cannot be recursively enumerable. Indeed iterated applications of might (alongside quantification) would reach arbitrary heights of the arithmetic hierarchy, leading to ω-logic (Fernando [**6**]).

Acknowledgments. My thanks to D. Beaver, S. Berman, H. Kamp, E. Keenan, E. Krahmer, P. Krause and R. van Rooy for helpful discussions on this subject.

References

[1] David Ian Beaver. What comes first in dynamic semantics. Dissertation, University of Edinburgh, 1994.

[2] Robin Cooper. Situation theoretic discourse representation theory. Manuscript, HCRC, Edinburgh, 1992.

[3] Jan van Eijck. Presuppositions and dynamic logic. In M. Kanazawa, C. Piñon, and H. de Swart, editors, *Papers from the Second CSLI Workshop on Logic, Language and Computation, June 1993.* CSLI Lecture Notes, to appear.

[4] Tim Fernando. What is a DRS? In R. Cooper and J. Groenendijk, editors, *Integrating Semantic Theories II.* Dyana deliverable R2.1.B, 1994. Presented in a Computational Semantics workshop at Tilburg (December 1994).

[5] Tim Fernando. A persistent notion of truth in dynamic semantics. In D. Westerståhl and J. Seligman, editors, *Language, Logic and Computation: The 1994 Moraga Proceedings.* CSLI, Stanford, To appear.

[6] Tim Fernando. It might change. A reply to Groenendijk, Stokhof and Veltman's "This might be it", Unfinished manuscript.

[7] B.C. van Fraassen. Presupposition, implication and self-reference. *Journal of Philosophy*, 65, 1968.

[8] J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14, 1991.

[9] J. Groenendijk, M. Stokhof, and F. Veltman. This might be it. In D. Westerståhl and J. Seligman, editors, *Language, Logic and Computation: The 1994 Moraga Proceedings.* CSLI, Stanford, To appear. Also appears in Dyana deliverable R2.1.B, *Integrating Semantic Theories II*, (R. Cooper and J. Groenendijk, editors), 1994.

[10] David Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2. Dordrecht: Reidel, 1984.

[11] Irene Heim. On the projection problem for presuppositions. In M. Barlow, D.P. Flickinger, and M.T. Westcoat, editors, *Proc. West Coast Conference on Formal Linguistics*, volume 2. Stanford Linguistics Association, California, 1983.

[12] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Language and Computation*. Addison-Wesley, 1979.

[13] H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht, 1993.

[14] Lauri Karttunen. Presupposition and linguistic context. *Theoretical Linguistics*, pages 181–194, 1973.

[15] Emiel Krahmer. Partiality and dynamics: theory and applications. In *Proc. Ninth Amsterdam Colloquium*. ILLC, University of Amsterdam, 1994.

[16] David Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proc. 5th GI Conference*, LNCS 104. Springer-Verlag, Berlin, 1981.

[17] Robert van Rooy. A two-dimensional account of presuppositions in quantified contexts, 1995. Manuscript, University of Stuttgart.

[18] Rob A. van der Sandt. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9(4), 1992.

[19] Frank Veltman. Defaults in update semantics. In H. Kamp, editor, *Conditionals, Defaults and Belief Revision*. Edinburgh, Dyana deliverable R2.5.A, 1990.

5

A persistent notion of truth in dynamic semantics

ABSTRACT. For a certain interpretation of first-order formulas A as binary (input/output) relations $[\![A]\!]$ on a set S of states (specifying state transitions induced by A), a notion $|A| \subseteq S$ of truth for A is investigated, arising from a so-called double (intuitionistic) negation translation of the domain of $[\![A]\!]$ (or of, equivalently, as it turns out, the fixed points of $[\![A]\!]$). A global, Boolean-valued analysis is presented alongside a local, three-valued non-compositional approximation of it. Complications with existence and identity are exposed, and suitable generic models constructed. The analysis is carried over to a determinization of transitions between "disjunctive" sets of output states.

1. INTRODUCTION

The essential idea behind "dynamic semantics" (e.g., [**14**]) is that the semantic content of a formula (or statement) lies not so much in its truth relative to a state (or context) but in the change in state that it induces (that change being the reason the statement is *used*). That is, over a given set S of states, a formula A is interpreted as a binary relation $[\![A]\!] \subseteq S \times S$ describing transitions between states s and s'

$$s[\![A]\!]s' \quad \text{iff} \quad \text{on input } s,\ A \text{ can output } s' \ .$$

By contrast, let us understand the truth value $|A|$ of a formula A to be the subset $\{s \in S : A \text{ is true at } s\}$ of S, given some definition of

$$(*) \quad A \text{ is true at } s \ .$$

The problem addressed by the present work is to define a notion $(*)$ of truth, given a relational interpretation $[\![A]\!]$ of formulas A.

1.1. Related work. Two obvious candidates for $(*)$ mentioned in [**4**] amount essentially to what are called acceptance and acceptability in [**19**]

(i) A is *accepted at* s if $s[\![A]\!]s$ (i.e., $s \in fix([\![A]\!])$)
(ii) A is *acceptable at* s if there is some state s' for which $s[\![A]\!]s'$ (i.e., $s \in dom([\![A]\!])$).

Neither definition mentions any notion of partiality on states, measuring say information content — a feature that is particularly prominent in [**19**], where a state is formed from a set of worlds. If the partiality of a state s is understood to refer to the possibility of a formula A as well as its negation being (separately) acceptable in s, then adopting (ii) for $(*)$ yields an incoherent notion of truth. Option (i) would, on the other hand, mean that asserting a truth can have *no* effect on the initial state, thereby building a certain logical omniscience into states, as well as trivializing the informational significance of truths.

Proceeding more concretely, consider, for example, *Dynamic Predicate Logic* (DPL, [**7**]), where a state is a function from a set of variables to the universe of some first-order model. It is easy to see (under definitions reviewed in §2.1 below) that over a first-order model M where the unary relation symbol R has a non-empty denotation R^M, and a function f that *fails* to map the variable x to an element of R^M,

$$\begin{array}{ll} R(x) & \text{is neither accepted nor acceptable at } f \ , \\ \exists x\ R(x) & \text{is not accepted but is acceptable at } f \ , \end{array}$$

and

$$(\exists x\ R(x))\ \&\ R(x) \quad \text{is not accepted but is acceptable at } f \ .$$

Hence, if the formula $\exists x R(x)$ ought to be true at f, then stipulating that A is true at s precisely when A is accepted at s is too strong to account for the dynamic effects of quantification. As for identifying truth with acceptability, it is slightly embarrassing to assert that the conjunction of a true formula with an untrue formula is true. It is only slightly embarrassing in view of the somewhat novel "dynamic" interpretation of conjunction as relational composition. Even so, however, when breaking new ground, there is every reason to pause before throwing out hard-won fruits of past labor, among which one might count a Boolean-valued notion of truth.

Bringing into the picture a pre-order $\sqsubseteq$ on the set S of states (comparing information content), a natural intuition is that if $s \sqsubseteq s'$ then every formula true at s is also true at s'. That is, the truth set $|A|$ of a formula A ought to be persistent, where a set $U \subseteq S$ of states is understood to be *persistent* if for all $s \in U$, $s' \sqsupseteq s$ implies $s' \in U$. As there is no reason to expect that $fix([\![A]\!])$ or $dom([\![A]\!])$ should be persistent relative to natural measures $\sqsubseteq$ of information content (a point confirmed in section 4 below), a bit more work needs to be put into the definition of truth. And also, as it turns out, into the relational interpretation $[\![A]\!]$ of formulas A, which will be defined below to be more faithful to [**14**], extended with witness constructs $\epsilon x : A$ for explicit quantification $\exists x\ A$.

1.2. Summary of present work. Building on the idea that a state $s \in S$ is *encompassed by* a set $U \subseteq S$ of states if

$$(\forall s' \sqsupseteq s)\ (\exists s'' \sqsupseteq s') \qquad s'' \in U \ ,$$

a formula A is defined to be *true at* s if s is encompassed by $fix([\![A]\!])$ — i.e.,

$$(\forall s' \sqsupseteq s)\ (\exists s'' \sqsupseteq s') \qquad s''[\![A]\!]s'' \ .$$

Then the truth set $|A| = \{s \in S : A \text{ is true at } s\}$ is manifestly persistent, and is, in fact, a persistent negation applied twice. That is, $|A|$ is the image

$j(fix([\![A]\!]))$ of $fix([\![A]\!])$ under the the so-called *double negation translation*[1] j that maps subsets U of S into persistent subsets of S according to

$$j(U) \quad = \quad \{s \in S \ : \ s \text{ is encompassed by } U\} \ .$$

A certain interpretation $[\![A]\!]$ of first-order formulas A as binary relations on finite functions from a set *Var* of variables to the universe $|M|$ of a first-order model M is defined supporting the following facts.

(i) Although acceptance is, as it stands, an inappropriate notion of truth for the dynamic effects introduced to handle anaphora, j makes acceptance acceptable in the sense that

$$j(fix([\![A]\!])) \quad = \quad j(dom([\![A]\!])) \ .$$

(ii) $|A|$ is compositional (relative to the logical connectives), and is moreover Boolean-valued (e.g., $|A \& B| = |A| \wedge |B|$).

(iii) The truth of A at s is determined by the behavior of A at completions of s, or more precisely, the "generic models" of s

$$A \text{ is true at } s \quad \text{iff} \quad \text{every generic model of } s \text{ satisfies } A \ .$$

(iv) $|A|$ supports truth gaps implicated in presupposition failure.

(v) Under a passage to a richer notion of ("information") state (compatible with [**19**]), j is well-behaved.

The relational interpretation $[\![A]\!]$ is essentially the part of [**14**] that was meant to be reformulated by DPL, plus certain constructs introduced to handle complications with existence and identity. These constructs are reminiscent of Hilbert's epsilon terms as well as restricted parameters in situation theory.

2. Preliminaries

The linguistic motivation for the present work can be traced back to *Discourse Representation Theory* (DRT, [**14**]). Although DRT covers a good deal of ground (see, for example, [**13**]), only a small part of it need concern us here. That part is henceforth referred to as DRT/DPL, having been analyzed in DPL, modulo certain simplifications that unfortunately constitute significant differences from DRT. Nonetheless, it is instructive to reverse the

[1] "Negation" here refers to Kripke's semantic interpretation of the intuitionistic negation $\overline{A}$ of a formula A under a forcing relation $\Vdash$ between states s partially ordered by $\leq$ and formulas

$$s \Vdash \overline{A} \quad \text{iff} \quad \text{not } (\exists s' \geq s) \ s' \Vdash A \ .$$

The double negation translation $\overline{\overline{A}}$ of a formula A then yields a so-called *weak forcing* relation $\Vdash^w$ (e.g., [**15**])

$$\begin{aligned} s \Vdash^w A \quad & \text{iff} \quad s \Vdash \overline{\overline{A}} \\ & \text{iff} \quad (\forall s' \geq s)(\exists s'' \geq s') \ s'' \Vdash A \end{aligned}$$

typically subject to Boolean logic. In the present context, the terms "weak acceptance" and "weak acceptability" are somewhat inappropriate because neither acceptance nor acceptability is persistent, as pointed out in section 4, and thus neither implies truth.

historical order, and present DPL (in its simplicity) before DRT/DPL. Towards that end, fix a signature **L** with equality, and a countable set *Var* of variables.

2.1. Dynamic Predicate Logic. At the heart of DPL is a translation $\cdot^{DPL}$ of first-order **L**-formulas with variables from *Var* into programs from (quantified) dynamic logic (see, for instance, [**9**]) according to

$$\begin{array}{rcll} A^{DPL} & = & A? & \text{for atomic formulas } A \\ (\neg A)^{DPL} & = & \neg(A^{DPL}) & \\ (A \& B)^{DPL} & = & A^{DPL} \ ; \ B^{DPL} & \\ (\exists x\ A)^{DPL} & = & x := ?\ ;\ A^{DPL}\ , & \end{array}$$

where the programs p (in the right hand side) are interpreted relative to an **L**-model M as binary relations $\rho(p)$ on functions $(f, g, h, \ldots)$ from *Var* to the universe $|M|$ of M as follows

$$\begin{array}{rcl} f\ \rho(A?)\ g & \text{iff} & f = g \text{ and } M \models A[f] \\ f\ \rho(x := ?)\ g & \text{iff} & (\forall x' \neq x)\ f(x') = g(x') \\ f\ \rho(p;q)\ g & \text{iff} & (\exists h)\ f\ \rho(p)\ h \ \text{ and } \ h\ \rho(q)\ g \\ f\ \rho(\neg p)\ g & \text{iff} & f = g \text{ and } f \notin \mathit{dom}(\rho(p))\ . \end{array}$$

The reader familiar with dynamic logic may code $\neg p$ up as the test $([p]\bot)?$, where $[p]$ is the universal (box) modality labelled by the program p, and $\bot$ is some contradictory formula. As in classical logic, disjunction, implication and universal quantification can be defined in DPL from negation, conjunction and existential quantification

$$\begin{array}{rcl} A \vee B & \equiv & \neg(\neg A\ \&\ \neg B) \\ A \supset B & \equiv & \neg(A\ \&\ \neg B) \\ \forall x\ A & \equiv & \neg\exists x\ \neg A\ , \end{array}$$

where $A \equiv B$ means $\rho(A^{DPL}) = \rho(B^{DPL})$.

2.2. DRT/DPL. A consequence of the use of total functions as inputs and outputs in dynamic logic is that the value of a variable can be destroyed by a program interpreting quantification. One can argue, however, that there is no compelling reason for such destruction of information, and that, indeed, the use of total functions in dynamic logic obscures the expansive growth of information implicit in the introduction of discourse markers in DRT. Without getting into the details of DRT, let us switch from total functions to finite functions, and call a function from a finite subset of *Var* to $|M|$ an (M-)*substitution*. (We will often say "substitution" without mentioning M, with the understanding that M is fixed in the background.) The interpretations $\rho(p)$ are characterized as before (i.e., $A?$ as a test for A, sequential composition $p;q$ as relational composition $\rho(p) \circ \rho(q)$, and negation $\neg p$ as a test for the complement of the domain of $\rho(p)$), except that the relation $\rho(x :=?)$ is no longer symmetric:

$$f \ \rho(x :=?) \ g \quad \text{iff} \quad \begin{array}{l} dom(g) = dom(f) \cup \{x\} \text{ and} \\ (\forall x' \in dom(f) - \{x\}) \ f(x') = g(x') \end{array}$$

for all substitutions f and g. Now, to protect a pre-existent binding on a variable $x \in Var$, let us define a *guarded assignment* $x := *$ to behave exactly like $x :=?$ on input substitutions where x is uninitialized, and to do nothing elsewhere. That is, let $x := *$ abbreviate

$$x = x? \quad + \quad \neg(x = x?) \ ; \ x :=? \ ,$$

where $+$ is non-deterministic choice, and interpreted as union:

$$f \ \rho(p + q) \ g \quad \text{iff} \quad f \ \rho(p) \ g \ \text{ or } \ f \ \rho(q) \ g$$

for all substitutions f and g. Observe that $\neg\neg$ reduces $+$ to DPL disjunction

$$\rho(\neg\neg(p + q)) \quad = \quad \rho(\neg(\neg p; \neg q)) \ ,$$

or, put in another way, $+$ provides a "dynamic" alternative to DPL disjunction. Note also that because a substitution need not be defined on a variable occurring in a formula A, the program $A? + \overline{A}?$ (or $\neg(\neg A?; \neg\overline{A}?)$), where $\overline{A}$ is interpreted as the complement of A, may fail to return an output. That is, bivalence may break down because of variables undefined at the input state.

The precise relationship between the modification of DPL above and DRT is described in [**6**]. As detailed there, guarded assignments provide an implicit treatment of existential quantification, in the sense that the fragment of DRT covered by DPL is given by the following translation $\cdot^\bullet$ of quantifier-free first-order formulas

$$\begin{array}{rcl} (R(\overline{t}))^\bullet & = & \overline{x} := * \ ; \ R(\overline{t})? \quad \text{for all relation} \\ & & \qquad\qquad\qquad \text{symbols } R \text{ including } = \\ (A \& B)^\bullet & = & A^\bullet; B^\bullet \\ (\neg A)^\bullet & = & \neg(A^\bullet) \end{array}$$

where the list $\overline{x} = x_1, \ldots, x_n$ in the first line for $\cdot^\bullet$ above consists of all variables occurring in $\overline{t}$, and $\overline{x} := *$ is $x_1 := *; \ldots; x_n := *$. An explicit treatment of quantification will be presented in the next section that extends this translation conservatively.

3. An interpretation of formulas as programs

To overcome complications with identity and existence, it is useful to introduce terms that witness existential formulas $\exists x \ A$. Such terms can be viewed as "labelled variables" $y_{A,x}$ or as epsilon terms $\epsilon x : A$, similar to those of Hilbert mentioned in [**12**] (except that here such terms will not always be defined, thus blocking the treatment of universal quantification intended by Hilbert). (These terms can also be viewed situation-theoretically as restricted parameters; see [**5**].) We will use the notation $y_{A,x}$ and $\epsilon x : A$ interchangably.

3.1. A language. To avoid pesky variable clashes, let us be careful to partition the set Var of variables into two disjoint sets X and Y, where X is a countable set of "unlabelled variables" given at the outset (along with a fixed signature $\mathbf{L}$) and the set Y of fresh "labelled" variables is defined simultaneously with the set $\mathbf{L}(Var)$ of formulas as follows.

(i) If R is an n-ary relation symbol of $\mathbf{L}$ (including $=$), and $\overline{t}$ is a list of n $\mathbf{L}$-terms with variables from $X \cup Y$, then $R(\overline{t})$ belongs to $\mathbf{L}(Var)$.
(ii) If $x \in X$, and A and B belong to $\mathbf{L}(Var)$, then $A\&B$, $A \vee B$, $\neg A$, and $\exists x\ A$ belong to $\mathbf{L}(Var)$.
(iii) If $x \in X$ and $A \in \mathbf{L}(Var)$, then the fresh variable $y_{A,x}$ (= epsilon term $\epsilon x\!:\!A$) belongs to Y.

Observe that $\mathbf{L}(Var)$ is *not* closed under $\supset$ or $\forall$. The expressions $A \supset B$ and $\forall x A$ can, if the reader insists, be understood as abbreviations of $\neg(A\&\neg B)$ and $\neg\exists x \neg A$, respectively.

3.2. A translation. Next, modifying the DRT/DPL translation slightly, define a translation $\cdot^u$ from formulas $A \in \mathbf{L}(Var)$ into programs by

$$
\begin{array}{lrcl}
(1) & (R(\overline{t}))^u & = & \overline{x} := * \;;\; R(\overline{t})? \\
 & (A\&B)^u & = & A^u ; B^u \\
 & (A \vee B)^u & = & A^u + B^u \\
 & (\neg A)^u & = & \neg(A^u) \\
 & (\exists x\ A)^u & = & y_{A,x} := * \;;\; A[y_{A,x}/x]^u \;; \\
(2) & & & \mathit{always}(A[y_{A,x}/x]) \;,
\end{array}
$$

where the list $\overline{x}$ in (1) consists of all variables in X occurring in $\overline{t}$, the witness variable $y_{A,x}$ in (2) belongs to Y, and the program $\mathit{always}(A[y_{A,x}/x])$ will be described shortly. Observe that variables in X occuring freely in a formula A are automatically initialized in A^u (in accordance with (1)), whereas other variables (in Y) are initialized only when they are bound (according to (2)). That is, the choice between using unlabelled and labelled variables (to represent discourse markers) in a formula depends on how much one wishes to fuss over variable initializations.[2]

[2]For example, the donkey sentence

`If a farmer owns a donkey then he beats it`

can be translated as

$$\mathtt{farmer}(x)\ \&\ \mathtt{own}(x,x')\ \&\ \mathtt{donkey}(x') \quad \supset \quad \mathtt{beat}(x,x')$$

(because of effects on initializing unlabelled variables reminiscent of [**17**]), or (among other possibilities) as

$$(\exists x)(\exists x')(\mathtt{farmer}(x)\ \&\ \mathtt{own}(x,x')\ \&\ \mathtt{donkey}(x')) \quad \supset \quad \mathtt{beat}(\epsilon x\!:\!\exists x' A, \epsilon x'\!:\!A')\ ,$$

To insure that a labelled variable $y_{A,x}$ is initialized only if it will always witness the formula, a program $always(A[y_{A,x}/x])$ is appended[3] to the translation of $\exists x A$. (For example, if A were $v = v \supset R(x)$ — i.e., $\neg(v = v \& \neg R(x))$ —, where $v \in Y$, then any object would witness $\exists x A$ at a state f where $v \notin dom(f)$.) Note that a witness to A can be spoiled only if variables mentioned in A^u are given new values. Fortunately, there are only finitely many such variables. This allows us to construct $always(A)$ as

$$always(A) \quad = \quad \neg\ (in(A)\ ;\ \neg(A^u))\ ,$$

where the program $in(A)$ is the non-deterministic choice

$$\sum In(A)$$

of the set $In(A)$ of all "legal" initializations that have some bearing on A^u. Just what the set $In(A)$ is, we turn to next. The reader is warned that the description of $In(A)$ is annoyingly technical, and is perhaps best skipped on a first reading. (Afterall, once all free variables in A which can be initialized are initialized, the test $always(A[y_{A,x}/x])$ added to (2) can be ignored.) With this in mind, the definition of $In(A)$ is indented below to mark it from the rest of the text.

> For $v \in Y$, let us write W_v for the formula that v should witness
>
> $$W_v \quad = \quad C[v/x] \qquad \text{where } v \text{ is } y_{C,x}\ .$$
>
> Then, given an $A \in \mathbf{L}(Var)$, define the sets $Y_A \subset Y$ and $Fmla_A \subset \mathbf{L}(Var)$ inductively by
>
> > if $v \in Y$ occurs in A or in some $B \in Fmla_A$ then v belongs to Y_A and W_v belongs to $Fmla_A$.
>
> **Proposition**. *For every $A \in \mathbf{L}(Var)$, $A \notin Fmla_A$ and the sets $Fmla_A$ and Y_A are finite. Moreover, for every $B \in Fmla_A$, $Fmla_B \subset Fmla_A$ and $Y_B \subseteq Y_A$.*
>
> **Proof**. That $A \notin Fmla_A$ follows from the inductive construction of Var and $\mathbf{L}(Var)$. The inclusions $Fmla_B \subset Fmla_A$ and $Y_B \subseteq Y_A$ for $B \in Fmla_A$ are easy consequences of the definitions of $Fmla_{(\cdot)}$ and $Y_{(\cdot)}$. An application of König's Lemma to the derivation trees of $Fmla_A$ and Y_A yields the finiteness of the sets $Fmla_A$ and Y_A. $\dashv$

---

where A is $\texttt{farmer}(x)$ & $\texttt{own}(x, x')$ & $\texttt{donkey}(x')$, A' is $A[(\epsilon x : \exists x' A)/x]$, and $B \supset C$ is understood as an abbreviation of $\neg(B \& \neg C)$. Observe that we lose the DPL equivalences

$$\begin{aligned}(\exists x A)\ \&\ B \ &\equiv\ \exists x(A\ \&\ B)\\ (\exists x A)\ \supset\ B \ &\equiv\ \forall x(A\ \supset\ B)\end{aligned}$$

because of variable labelling, although $\exists x$ still introduces a variable available for reference outside $\exists$'s usual scope.

[3]An alternative might be to initialize all variables in A that can be initialized before choosing a witness for $\exists x A$

$$(\exists x\ A)^u \quad = \quad init(A)\ ;\ y_{A,x} := *\ ;\ A[y_{A,x}/x]^u$$

(for some suitable program $init(A)$) but this possibility will not be pursued here.

Next, let X_A be

$$\{x \in X \ : \ x \text{ occurs freely in } A \text{ or in some } B \in Fmla_A\} ,$$

and finally let $In(A)$ be the (finite) set of programs of the form

$$\begin{gathered} x_1 := * \ ; \ x_2 := * \ ; \ \cdots \ ; \ x_m := * \ ; \\ v_1 := * \ ; \ v_2 := * \ ; \ \cdots \ ; \ v_n := * \ ; \\ (W_{v_1})^u \ ; \ (W_{v_2})^u \ ; \ \cdots \ ; \ (W_{v_n})^u \end{gathered}$$

where $x_1, \ldots, x_m$ is some list of elements from X_A (that does not necessarily include every $x \in X_A$), and (similarly) $v_1, \ldots, v_n$ is some list of elements from Y_A. (The sequence $(W_{v_1})^u \ ; \ (W_{v_2})^u \ ; \cdots \ ; \ (W_{v_n})^u$ is appended to verify that the initializations of labelled variables are "legal".) The astute reader concerned about the inductive character of our definition of $\cdot^u$ may draw from the previous proposition the

Corollary. *The binary relation $\succ$ on $\mathbf{L}(Var)$ given by*

$$A \succ B \quad \textit{iff} \quad B \in Fmla_A$$

is well-founded (and transitive), and for every $A \in \mathbf{L}(Var)$, the set $\{B : A \succ B\}$ is finite.

Having defined the translation $\cdot^u$, let us isolate the set S of *states* that will interest us

$$S \quad = \quad \{f \ : \ \text{for some } A \in \mathbf{L}(Var), \ \emptyset \ \rho(A^u) \ f\} \ .$$

For every $A \in \mathbf{L}(Var)$, define

$$[\![A]\!] \quad = \quad \rho(A^u) \text{ restricted to } S \ .$$

Notice that for every $A \in \mathbf{L}(Var)$, the program A^u is *guarded* in that all occurrences of random assignments in it are within guarded assignments.

Proposition 1 (Monotonicity). *For every guarded program p, and for all substitutions f and g, if $f \ \rho(p) \ g$ then $f \subseteq g$. In particular, for all $A \in \mathbf{L}(Var)$, $f[\![A]\!]g$ implies $f \subseteq g$.*

By contrast, under DPL, the output state of $[\![\exists x A]\!]$ may destroy a variable binding in the input state.

4. Truth of formulas *qua* programs

Let $\sqsubseteq$ be the restriction of $\subseteq$ to S. A natural property to require of a definition of "A is true at f" is that truth persist through $\sqsubseteq$-larger states g. Neither acceptance nor acceptability is persistent; indeed

$$f[\![A]\!]f \text{ and } f \sqsubseteq g \quad \text{need not imply} \quad g \in dom([\![A]\!]) \ ,$$

since a variable may be assigned a value at g that precludes the acceptability of A. (For example, take A to be $\neg x = x$.) To secure persistence, let us recall the double negation translation j mentioned in the introduction.

Definition. A formula A of $\mathbf{L}(Var)$ is *true at* a state $f \in S$ if $f \in j(fix([\![A]\!]))$ — i.e.,

$$(\forall g \sqsupseteq f)\ (\exists h \sqsupseteq g) \qquad h[\![A]\!]h\ .$$

Proposition 2 (Persistence). *If A is true at f and $f \sqsubseteq g$ then A is true at g.*

Proposition 2 is immediate from the transitivity of $\sqsupseteq$. The next lemma may seem rather technical, but will nevertheless prove important, particularly as $fix([\![A]\!])$ is not necessarily persistent.

Lemma 3 (Eventual stability). *Given an $A \in \mathbf{L}(Var)$ and a $g \in S$, there is a $\hat{g} \sqsupseteq g$ such that*

$$(\exists h \sqsupseteq \hat{g})\ h \in dom([\![A]\!]) \quad \textit{implies} \quad (\forall h \sqsupseteq \hat{g})\ h[\![A]\!]h\ .$$

Proof. The idea is to choose $\hat{g}$ such that $g[\![A']\!]\hat{g}$, where A' is some tautology that translates (under $\cdot^u$) to a program initializing all variables in A^u which can be initialized (given input g). Towards this end, recall the definitions of $Fmla_A$, Y_A and X_A from §3.2, and choose (by the inductive construction of $\mathbf{L}(Var)$ along with Var) an enumeration $B_0, \ldots, B_n$ of $Fmla_A$ such that for all i and j with $0 \leq i < j \leq n$,

if B_j is W_v then v does not occur in B_i.

Take A' to be

$$\bigwedge_{0 \leq j \leq m} x_j = x_j \quad \& \quad \bigwedge_{0 \leq i \leq n} (B_i \vee \neg B_i)\ ,$$

where $x_0, \ldots, x_m$ is an (any) enumeration of the set X_A of all variables in X occurring freely in A or in some $B \in Fmla_A$.[4] Let $Var_A = Y_A \cup X_A$ and F be the set of substitutions in S with domain contained in Var_A

$$F \;=\; \{f \in S\ :\ dom(f) \subseteq Var_A\}\ .$$

Observe that for every $\hat{g}$ such that $g[\![A']\!]\hat{g}$, the restriction of $\hat{g}$ to Var_A is $\sqsubseteq$-maximal among all substitutions in F, yielding the desired property. $\dashv$

We can now provide three alternatives U to $fix([\![A]\!])$ satisfying $j(U) = j(fix([\![A]\!]))$.

Theorem 4. *Given an $A \in \mathbf{L}(Var)$ and an $f \in S$, the following are equivalent.*

(i) *A is true at f.*
(ii) $f \in j(dom([\![A]\!]))$.
(iii) $f \in j(\{h \in S : (\forall f' \sqsupseteq h)\ f'[\![A]\!]f'\})$.
(iv) $f \in j(\{h \in S : (\forall f' \sqsupseteq h)\ f' \in dom([\![A]\!])\})$.

[4] This proof exploits the "dynamic" interpretation of disjunction as union, $[\![A \vee B]\!] = [\![A]\!] \cup [\![B]\!]$. If disjunction $A \vee B$ is instead reduced "statically" to $\neg(\neg A \& \neg B)$, then the formula A' in the proof above will have to be replaced by one of the 2^{n+1} combinations of negated or unnegated B_i's that is satisfiable (i.e., returns an output).

Proof. Clearly, (i) implies (ii), and (iii) implies (i). To see that (ii) implies (iii), assume (ii) and suppose $g \sqsupseteq f$. Appealing to Lemma 3, pick a $\hat{g} \sqsupseteq g$ such that (by (ii)) $(\forall f' \sqsupseteq \hat{g})\ f'[\![A]\!]f'$, thus validating (iii). Finally, note that (iii) implies (iv), while (iv) implies (ii). $\dashv$

As $dom([\![A]\!]) = dom([\![\neg\neg A]\!])$, Theorem 4 yields

Corollary 5 (Static projection). *For every formula A and state f, A is true at f iff for some B such that $[\![B]\!] = [\![\neg\neg A]\!]$, B is true at f.*

As $\neg\neg p$ never returns an output state different from the input, Corollary 5 raises the question as to whether or not the truth of a formula at a state f can be predicted from information about the truth of its (proper) parts at f. As the double negation translation j involves the totality of all states, it is hardly surprisingly that the answer, as the next subsection shows, is negative. Nevertheless, this exercise will prove useful warm-up for the global analysis that then follows.

4.1. A local three-valued analysis of truth. The present subsection is concerned with the question of predicting the truth of a formula A at a state f from information about the truth at the same state f of (proper) subformulas of A. Complementing the notion of truth, a notion of falsehood will be considered that builds on certain harmless assumptions, yielding a negation map $\sim$. Let $\mathbf{L}$ be a signature such that a bijective map $\hat{\cdot}$ on the relation symbols in $\mathbf{L}$ can be defined assigning to every R its "complement" $\hat{R}$ (also in $\mathbf{L}$) with the same arity, such that $\hat{\hat{\cdot}}$ is the identity on the relation symbols of $\mathbf{L}$. (A simple doubling of $\mathbf{L}$ will suffice.) An $\mathbf{L}$-model M is understood always to satisfy

$$(\dagger) \qquad R^M \subseteq |M|^n - \hat{R}^M$$

for every n-ary relation symbol R in $\mathbf{L}$, which is easy enough to express in the first-order language of $\mathbf{L}$. Next, let $\sim$ be the map on $\mathbf{L}(\mathit{Var})$ defined recursively by

$$\begin{array}{rcl} \sim R(\bar{t}) & = & \hat{R}(\bar{t}) \\ \sim(A \& B) & = & \sim A \vee \sim B \\ \sim(A \vee B) & = & \sim A \ \& \sim B \\ \sim\neg A & = & A \\ \sim(\exists x\ A) & = & \neg\exists x\ A \end{array}$$

(e.g., [**16**], except that the clause for $\exists$ is modified, based on an identification of $\forall x A$ with $\neg\exists x \neg A$, as $\mathbf{L}(\mathit{Var})$ is not closed under $\forall$). Because of ($\dagger$) above,

Lemma 6. *If $f[\![A]\!]f$ then not $f[\![\sim A]\!]f$.*

We can now assign a truth value $|A|_f$ to a formula $A \in \mathbf{L}(\mathit{Var})$ at a state f from the 3-element set $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. Define

$|A|_f = \mathbf{t}$ if A is true at f,
A is false at f, $|A|_f = \mathbf{f}$, if $|\sim A| = \mathbf{t}$,

and

A is undetermined at f, $|A|_f = \mathbf{u}$, otherwise.

That is, $|A|_f = \mathbf{u}$ iff

$$(\exists g \sqsupseteq f)(\forall h \sqsupseteq g) \text{ not } h[\![A]\!]h \quad \text{and}$$
$$(\exists g \sqsupseteq f)(\forall h \sqsupseteq g) \text{ not } h[\![\sim A]\!]h \ .$$

A formula may be neither true nor false at a state not only because a variable may be uninitialized at that state, but also because the inclusion $\subseteq$ in (†) may be proper. Moreover, from the preceding lemma,

Corollary 7 (3-valuedness). *A cannot be both true and false at the same state.*

Although the truth values $\mathbf{t}$ and $\mathbf{f}$ persist, $\mathbf{u}$ need not. Holding a state f fixed, however, the definitions above yield the following truth tables, where $A \supset B$ is taken as an abbreviation of $\neg(A\&\neg B)$.

A	B	$\&$	$\sim$	$\vee$	$\supset$
t	**t**	**t**	**f**	**t**	**t**
t	**f**	**f**		**t**	**f**
t	**u**	**u**		**t**	**u**
f	**t**	**f**	**t**	**t**	**t**
f	**f**	**f**		**f**	**t**
f	**u**	**f**		**u**	**t**
u	**t**	**u**	**u**	**t**	**t**
u	**f**	**f**		**u**	$\neq$ **f**
u	**u**	$\neq$ **t**		$\neq$ **f**	$\neq$ **f**

The truth tables are strong Kleene except on certain entries where one of the subformulas has an undetermined truth value and no truth value can be predicted for the formula as a whole. For instance, if $|A|_f = |B|_f = \mathbf{u}$, then $|A \supset B|_f$ can be $\mathbf{t}$, as with

$$|x = 1 \supset x = 1|_{\emptyset} \ ,$$

or $\mathbf{u}$, as with

$$|x = 1 \supset x = 0|_{\emptyset} \ .$$

Observe that $|A \supset B|_f$ can never be $\mathbf{f}$, or else $\sim(A \supset B)$ would be true at f; i.e., $|A \ \&\neg B|_f = \mathbf{t}$, which is absurd since $|A|_f = \mathbf{u}$.

4.2. A global Boolean-valued analysis. The non-compositionality in §4.1 can be repaired through an analysis of truth that is global in that the relativization of the truth value $|A|_f$ to a state f is dropped.[5] Instead, we define

$$|A| \quad = \quad \{f \in S \ : \ A \text{ is true at } f\} \quad (= \ j(fix([\![A]\!]))) \ ,$$

with the intuition (if the reader likes) that the truth value $\mathbf{u}$ breaks up into many different values lying (properly) between $\emptyset$ and S.

[5]An alternative would be to keep the truth value local, but to set $|A|_f = \{g \in S : f[\![A]\!]g\} \in \text{Power}(S)$. There is then, however, no real difference with the dynamic interpretation $[\![A]\!]$, the compositionality of which carries over to $|A|$.

Rather than proceeding directly to $|A|$, let us try more slowly to relate $j(fix(\llbracket A \rrbracket))$ to standard constructions in algebraic semantics. Towards this end, define

$$\begin{aligned} \Omega &= \{U \subseteq S \;:\; (\forall f \in U)(\forall g \sqsupseteq f)\; g \in U\} \\ \Omega_j &= \{j(U) \;:\; U \subseteq S\}\;, \end{aligned}$$

the first set Ω consisting of subsets of S persistent with respect to $\sqsubseteq$, and the second set Ω_j of j-sets.

Theorem (Folklore).

(a) $\langle \Omega, \cap, \cup, \rightarrow, \emptyset \rangle$ *is a complete Heyting algebra, where*

$$U \rightarrow U' \;=\; \{f \in S \;:\; (\forall g \sqsupseteq f)\;\; \textit{if } g \in U \textit{ then } g \in U'\}$$

for $U, U' \in \Omega$, and least upper bounds are given by union, and greatest lower bounds by intersection.

(b) $\langle \Omega_j, \cap, \sqcup, c, S, \emptyset \rangle$ *is a complete Boolean algebra, where*

$$\begin{aligned} U \sqcup U' &= j(U \cup U') \\ c(U) &= \{f \in S \;:\; (\forall g \sqsupseteq f)\; g \notin U\} \end{aligned}$$

for U and $U' \in \Omega_j$, and

$$\begin{aligned} \coprod_{i \in I} U_i &= j(\bigcup_{i \in I} U_i) \\ \prod_{i \in I} U_i &= j(\bigcap_{i \in I} U_i) \end{aligned}$$

for $\{U_i \;:\; i \in I\} \subseteq \Omega_j$.

Next, to provide an algebraic semantics for $\mathbf{L}$-sentences relative to a fixed $\mathbf{L}$-model M, it is customary to extend the signature $\mathbf{L}$ to a signature $\mathbf{L}_M$ by throwing into $\mathbf{L}$ constants naming elements in $|M|$ (identifying $m \in |M|$ with its name). Such a step will not be required here, although $\mathbf{L}$ can, if the reader wishes, be assumed to have names for all objects in $|M|$. In any case, we will interpret not only $\mathbf{L}$-sentences (without free variables), but also arbitrary formulas from $\mathbf{L}(Var)$ (possibly with free variables from Var). Or, put another way, the variables in Var will be treated as constants, which, as we will see, will serve as the usual witnesses.

Another point the reader familiar with algebraic semantics might raise is that the double negation translation j is commonly applied to a Heyting algebra — for example, the Heyting algebra Ω at hand. Although neither $fix(\llbracket A \rrbracket)$ nor $dom(\llbracket A \rrbracket)$ is necessarily in Ω, Theorem 4 suggests defining a (persistent) forcing relation $\Vdash$ between $f \in S$ and $A \in \mathbf{L}(Var)$ by either

$$f \Vdash A \quad \text{iff} \quad (\forall g \sqsupseteq f)\; g \in dom(\llbracket A \rrbracket)\;,$$

or (non-equivalently)

$$f \Vdash A \quad \text{iff} \quad (\forall g \sqsupseteq f)\; g \llbracket A \rrbracket g\;.$$

Unfortunately, neither alternative provides (for instance) a decomposition of $\{f : f \Vdash A \vee B\}$ as the union (the join in the Heyting algebra Ω) of $\{f : f \Vdash A\}$ and of $\{f : f \Vdash B\}$. (Consider the case where A is $(\epsilon x : C) = (\epsilon x : C)$ and B is $\neg A$.) Some Heyting algebra other than Ω is called for; but the only natural one the present author has found is the Boolean algebra Ω_j of j-sets!

A final point before presenting our Boolean-valued analysis of truth concerns our treatment of quantification. Following our coding of guarded assignments in terms of equality, the range of quantification is restricted to objects m that "exist" in the sense that $m = m$. Also, note that for every family $\mathcal{U} \subseteq \Omega_j$, if $\bigcup \mathcal{U} \in \Omega_j$ (i.e., $\bigcup \mathcal{U} = j(\bigcup \mathcal{U})$), then $\bigcup \mathcal{U} = \coprod \mathcal{U}$.

Theorem 8 (Boolean-valued semantics). *For all $A, B \in \mathbf{L}(Var)$,*

$$
\begin{aligned}
|A \& B| &= |A| \cap |B| \\
|A \vee B| &= |A| \sqcup |B| \quad (= j(|A| \cup |B|)) \\
|\neg A| &= c(|A|) \\
&= j(S - |A|) \\
|\exists x\ A| &= |(\epsilon x : A) = (\epsilon x : A)| \\
&= \bigcup \{|v = v\ \&\ A[v/x]| \ :\ v \in Var\} \\
&= \bigcup \{|A[m/x]| \ :\ m \in |M|\} \qquad \textit{if } \mathbf{L} \textit{ names all} \\
& \qquad\qquad\qquad\qquad\qquad\qquad\quad \textit{objects in } M.
\end{aligned}
$$

Proof. Use monotonicity (Proposition 1) and the various characterizations of $|A|$ provided by Theorem 4. For example, to see that $f \in |A \& B|$ implies $f \in |B|$, fix a $g \sqsupseteq f$ and conclude (as required by Theorem 4(ii)) that there must be some $h \sqsupseteq g$ in $dom([\![B]\!])$ since $f \in |A \& B|$. Assuming that $|A \& B| = |A| \cap |B|$ has been established, then we can appeal to static projection (Corollary 5) to reduce $|A \vee B| = |A| \sqcup |B|$ to $|\neg A| = c(|A|)$, since $[\![\neg\neg(A \vee B)]\!] = [\![\neg(\neg A \& \neg B)]\!]$. By definition, $|\neg A| = j(S - dom([\![A]\!]))$, and so by eventual stability (Lemma 3), $|\neg A| = \{f \in S : (\forall g \sqsupseteq f)\ g \notin |A|\}$. Note that eventual stability also yields $j(S - dom([\![A]\!])) = j(S - |A|)$.

Turning finally to quantification, these equations are consequences of the careful bookkeeping and coding that has gone into labelling, as well as our restriction of S to the substitutions accessible from $\emptyset$. The point is that witnesses are persistent in the sense that for all $f \in S$,

$$(\epsilon x : A) \in dom(f) \quad \text{implies} \quad f \in dom([\![\exists x\ A]\!]) \ .$$

Note the restriction in quantification to $v = v$ is required since quantified formulas translate to programs that always initialize the (translations of) variables bound by quantifiers. (Consider, for example, $\exists x\ \neg x = x$.) $\dashv$

Theorem 8 provides some evidence that our notion of truth is natural. A further argument is presented next, consisting of a construction along the partial order $\sqsubseteq$ on S, that will again force us to face the complications with existence and identity above.

4.3. Constructing generic models from states. To understand just what the double negation translation j is, we might ask what constraints on information growth from f are implied by $f \in j(U)$? Call a sequence $\{f_\alpha : \alpha < \kappa\} \subseteq S$ (for some cardinal κ) a *maximal $\sqsubseteq$-chain from f* if

(i) $f \sqsubseteq f_0$,
(ii) for all $\alpha < \kappa$ and $\beta < \alpha$, $f_\beta \sqsubseteq f_\alpha$, and
(iii) there is *no* $g \in S$ such that for all $\alpha < \kappa$, $f_\alpha \sqsubset g$.

Now, it is true enough that if every maximal $\sqsubseteq$-chain from f intersects U, then $f \in j(U)$. On the other hand, the converse need not hold: given a maximal $\sqsubseteq$-chain $f = f_0 \sqsubset f_1 \sqsubset f_2 \sqsubset \cdots$, the maximal chain $\{f_{2i} : i \in \omega\}$ does not intersect $U = \{f_{2i+1} : i \in \omega\}$. For a more pleasing fit with information growth, we must consider not only the partial order $\sqsubseteq$ on S, but also the formulas A in $\mathbf{L}(Var)$. In particular, we will adapt the theory of generic models described in [**15**] (going back to [**18**]).

It will be convenient to define the basic notions from [**15**] in our setting as follows. A subset U of S is a *generic set* if

(i) for all $f, f' \in U$, there is a $g \in U$ such that $f \sqsubseteq g$ and $f' \sqsubseteq g$, and
(ii) for every $A \in \mathbf{L}(Var)$, there is an $f \in U$ such that $f \in |A| \cup |\neg A|$.

Defining the *theory $Th(U)$ of U* to be

$$Th(U) \quad = \quad \{A \in \mathbf{L}(Var) \ : \ (\exists f \in U)\ f \in |A|\} \ ,$$

observe that condition (i) (in the definition of a generic set) is a consistency criterion, while (ii) is a completeness criterion. Next, given a signature $\mathbf{L}$, define an *$\mathbf{L}$-free model* to be an $\mathbf{L}$-model N, except that equality is construed as a non-logical symbol $\approx$ that is interpreted by N as some restriction of true equality

$$\approx^N \quad = \quad \{(n,n) \ : \ n \in dom(\approx^N)\} \ .$$

The term "free" here should be construed not in the usual algebraic sense but rather as in "free logic" (e.g. [**3**]), the intuition being that $dom(\approx^N)$ consists of the objects that exist. The notion of an $\mathbf{L}$-free model N *satisfying* an $\mathbf{L}$-sentence A, $N \models A$, is defined as in the familiar Tarskian manner (with $\approx$ regarded as a non-logical symbol), except that quantification is restricted to the domain of $\approx^N$

$$\begin{aligned} N \models \forall x A \quad &\text{iff} \quad (\forall n \in dom(\approx^N))\ N \models A[n/x] \\ N \models \exists x A \quad &\text{iff} \quad (\exists n \in dom(\approx^N))\ N \models A[n/x] \ . \end{aligned}$$

Now, treating Var as a set of constants, a *generic model for a* generic set U is an $\mathbf{L}(Var)$-free model N such that all its objects are named in Var

$$|N| \quad = \quad \{v^N \ : \ v \in Var\} \ ,$$

and every $\mathbf{L}(Var)$-sentence in $Th(U)$ is satisfied by N. If, moreover, $f \in U$, then N is said to be a *generic model for* f.

Theorem 9 (Generic Model Theorem). *Every $f \in S$ has a generic model; i.e., for every $f \in S$, there is a generic model for f.*

Proof. The proof is similar to pp. 101 and 102 of [**15**] except that (i) a generic set is constructed in a different way below (to avoid requiring that $\mathbf{L}(Var)$ be countable), (ii) conjunction, rather than disjunction, is taken to be primitive (since $\vee$ cannot be interpreted as union), and (iii) quantification must be restricted to objects n such that $n \approx n$. (Note that j turns DRT/DPL negation into intuitionistic negation.) More precisely, Theorem 9 is an immediate consequence of the following two lemmas (applied in sequence).

Lemma. *Every $f \in S$ belongs to a generic set.*

Proof. From an enumeration $\{A_\alpha : \alpha < \kappa\}$ of $\mathbf{L}(Var)$,construct a generic set $U = \{f_\alpha : \alpha < \kappa\}$ by taking f_0 to be f, and choosing f_α as follows, for $0 < \alpha < \kappa$. Following the notation of the proof of Lemma 3, let Var_α be the (finite) set of all variables occuring in some $B \in Fmla_{A_\alpha}$. Let g be the substitution

$$\{(v,m) \in Var_\alpha \times |M| \ : \ (\exists\beta < \alpha)\ v \in dom(f_\beta) \text{ and } f_\beta(v) = m\}$$

and pick a $\hat{g}$ such that $g[\![A'_\alpha]\!]\hat{g}$, where A'_α is the A' in the proof of Lemma 3, with A replaced by A_α. Set f_α to $\hat{g}$. $\dashv$

Lemma. *Every generic set has a generic model.*

Proof. Given a generic set U, define a (partial equivalence) relation $\equiv$ on Var by

$$v \equiv v' \quad \text{iff} \quad v \approx v' \in Th(U)\ ,$$

and form an $\mathbf{L}(Var)$-free model N by taking as its universe $|N|$ the set $\{v^\equiv : v \in Var\}$ of (partial equivalence) classes $v^\equiv = \{v' \in Var : v \equiv v'\}$ for $v \in Var$. (Note that $\emptyset \in |N|$.) Interpret the k-ary relation symbols R of $\mathbf{L}$ (inlcuding $\approx$) and the l-ary function (including constant) symbols F of $\mathbf{L} \cup Var$ according to

$$\begin{array}{rcl} R^N(v_1^\equiv, \ldots, v_k^\equiv) & \text{iff} & R(v_1,\ldots,v_k) \in Th(U) \\ F^N(v_1^\equiv, \ldots, v_l^\equiv) & = & y_{x=F(v_1,\ldots,v_l),x}{}^\equiv \ \text{for some} \\ & & \qquad x \in X - \{v_1,\ldots,v_l\}\ . \end{array}$$

These are legitimate definitions (i.e., independent of the choice of representatives v of $v^\equiv$), since $Th(U)$ respects the laws of equality restricted to $dom(\approx^N)$ and the witnessing on variables. Notice that every term τ has a denotation τ^N, and that $\tau^N = \emptyset$ precisely if $\tau \approx \tau \notin Th(U)$. Luckily, $\emptyset \not\approx^N \emptyset$, and we have no way of expressing equality in $\mathbf{L}(Var)$ except by its approximation $\approx$.) The expected equivalence

$$N \models A \quad \text{iff} \quad A \in Th(U)$$

can be established by a routine induction on A, appealing to the consistency criterion (i) of a generic set. $\dashv$

$\dashv$

Corollary 10. *Let $A \in \mathbf{L}(Var)$ and $f \in S$. A is true at f (i.e., $f \in |A|$) iff A is satisfied by every generic model for f.*

Proof. As in pp. 102 and 103 of [**15**], the forward direction is trivial, while the converse follows from Theorem 9. $\dashv$

Lastly, let us remark that the underlying $\mathbf{L}$-model M (relative to which $[\![\cdot]\!]$ and S are defined) is an elementary extension of N°, for every generic model N (based on M), where $\cdot^\circ$ is the obvious map from $\mathbf{L}(Var)$-free models to $\mathbf{L}$-models (obtained by throwing out all "non-existent" objects). Conversely, every countable elementary substructure of M can be obtained as such.

5. Discussion

A basic objection that might be raised against the double negation translation j on which our notion of truth rests is that it abstracts away the temporary instability of a formula (as Lemma 3 and ultimately Theorem 8 suggest). How is the doctrine that

the meaning of a formula is the change it induces

served by trivializing that very change? But, ideologies aside, if we refrain from reducing meaning to truth, and instead agree that truth is a "static" abstraction of meaning, then the objection loses much of its force. The question becomes, quite innocently, what does our notion of truth fail to abstract away? One answer is the partiality of $\mathbf{L}(Var)$. For although a good deal of effort was put above into developing the Boolean-valued character of the notion of truth proposed, the formally bivalent (i.e. 2-valued) interpretation in §4.3 is achieved only at the cost of a non-logical treatment of equality and a somewhat non-standard interpretation of quantification. Furthermore, as demonstrated in §4.1, one can define translations that have the effect of re-interpreting the logical connectives (for instance, negation[6]) to support some partiality. Let us conclude by briefly considering these points in relation to presuppositions, and turning to a richer notion of state similar to that in [**19**] (concerning which an additional sense in which the double negation translation is persistent will be established).

[6]Observe, however, that even though a negation $\sim$ different from DRT/DPL negation $\neg$ is introduced in §4.1, DRT/DPL negation is still useful for $\forall$ and presumably implication.

5.1. Truth gaps. With regard to truth gaps implicated in presupposition failure, two sources of partiality then remain:

(i) the partiality of the substitutions f, particularly relative to labelled variables (or epsilon terms), some of which cannot be initialized, and
(ii) the possibility of defining negation translations based on extension/anti-extension pairs that need not be full Boolean complements.

Comparing DRT/DPL negation $\neg$ with the negation translation $\sim$ employed in §4.1 to define falsehood, note that whereas $A \vee \neg A$ is always true, $A \vee \sim A$ may fail to be true even if for every k-ary $R \in \mathbf{L}$, $R^M = |M|^k - \hat{R}^M$ (in which case, $R^N = dom(\approx^N)^k - \hat{R}^N$ holds for every generic model N based on M). There remains the possibility of an epsilon term that can never be initialized — for example, $\epsilon x :$ wife(x,John), where John has no wife (in M). Such truth gaps provide natural opportunities for giving accounts of simple (existential) cases of presupposition failure. Interpreting $\hat{R}$ to be less than the full complement of the interpretation of R offers further possibilities.

5.2. Information states. The analysis above is predicated on a notion of state that is somewhat impoverished, next to [**10**]'s set of "sequence-world-pairs", commonly used in accounts of presuppositions. (This is a non-trivial matter inasmuch as our notion of truth is conditioned by it.) A sequence-world-pair is simply a substitution f, coupled with the first-order model M that structures the range of f. (It becomes important to keep track of M, when a state is built from different M's.) For a sufficiently large set $\mathcal{M}$ of $\mathbf{L}$-models, let

$$\mathcal{S} \;=\; \{(f, M) \mid M \in \mathcal{M} \text{ and } f \text{ is an } M\text{-substitution}\}\ .$$

The richer notion of a state as a subset of $\mathcal{S}$ can be obtained from the interpretation $[\![\cdot]\!]$ above, as in the classic construction of deterministic finite automata from non-deterministic ones (e.g. [**11**]). More precisely, given an $A \in \mathbf{L}(Var)$, lift the binary relation

$$[A] \;=\; \{((f, M), (g, M)) \mid M \in \mathcal{M} \text{ and } f[\![A]\!]_M g\}$$

on $\mathcal{S}$ to the binary relation $\mathcal{D}[A]$ on $\mathsf{Pow}_+(\mathcal{S})$ $(= \mathsf{Pow}(\mathcal{S}) - \{\emptyset\})$, given by

$$\mathcal{D}[A] \;=\; \{(U, V) \in \mathsf{Pow}_+(\mathcal{S}) \times \mathsf{Pow}_+(\mathcal{S}) \mid V = \{t \mid (\exists s \in U) s[A]t\}\}.$$

(Let us agree to exclude the empty set from the field of $\mathcal{D}[A]$, as that arises from the impossibility of an $[A]$-transition.) This lifting is feasible because all programs involved above are so-called distributive (i.e., definable at the level of single substitutions). We will attend later to the non-distributive programs $\mathsf{might}\ A$ of [**19**]. As $\mathcal{D}$ internalizes the non-determinism of $[\![A]\!]$ in states (from $\mathsf{Pow}_+(\mathcal{S})$), it is natural to lift a pre-order $\leq$ on $\mathcal{S}$ comparing information content to its so-called Smyth pre-order

$$\leq_{\mathcal{D}} \;=\; \{(U, V) \in \mathsf{Pow}_+(\mathcal{S}) \times \mathsf{Pow}_+(\mathcal{S}) \mid (\forall t \in V)\ (\exists s \in U)\ s \leq t\}\ .$$

Next, let us write J for the double negation translation $j^{\sqsubseteq_{\mathcal{D}}}$ defined relative to the Smyth pre-order $\sqsubseteq_{\mathcal{D}}$ of

$$\{((f, M), (g, M)) \quad | \quad M \in \mathcal{M} \text{ and } f \sqsubseteq_M g\} ,$$

where $\sqsubseteq_M$ is the partial order $\sqsubseteq$ used to define truth in section 3 for a fixed **L**-model M. Finally, define an $A \in \mathbf{L}(Var)$ to be *true at* a state $U \in \mathsf{Pow}_+(\mathcal{S})$ if $U \in J(fix(\mathcal{D}[A]))$.

Theorem 11. Given an $A \in \mathbf{L}(Var)$ and a $U \in \mathsf{Pow}_+(\mathcal{S})$, the following are equivalent.

(i) A is true at U.
(ii) $U \in J(dom(\mathcal{D}[A]))$.
(iii) For every $s \in U$, A is true at $\{s\}$.
(iv) For every $s \in U$, A is true at s (according to section 4).

Proof. Argue that (i) implies (ii), (ii) implies (iii), (iii) implies (iv), and finally (iv) implies (i). The first implication is immediate. The others hold because every $V \in \mathsf{Pow}_+(\mathcal{S})$ is $\sqsubseteq_{\mathcal{D}}$-dominated by some singleton (even though every singleton is also dominated by a non-singleton) and by appealing to Theorem 4. The universal quantifier in J leads to the universal (rather than existential) force in (iii) and (iv). (Observe that the theorem is quite robust under revisions of the interpretation $[\![A]\!]_M$.) $\dashv$

A final remark (which the present author plans to expand upon elsewhere) concerns [**19**]'s program construct might, two features of which would appear to be at odds with the analysis above — viz., non-persistence and non-distributivity. Just how useful our persistent notion of truth is in analyzing the non-persistence of might remains to be seen. As for the second feature, in the same way that modality can be treated syntactically (an approach advocated in [**1**] for propositional attitudes), a distributive analysis of might is possible, so long as updates are added to the underlying first-order model (exactly as sets are treated in generalized Henkin models for higher-order logic). The introduction of abstract objects into first-order models is argued and investigated at length in [**2**]. In the case of might, the alternative syntactic approach promises to be of interest, given complications with quantification defined at the richer level of $\mathsf{Pow}_+(\mathcal{S})$-states (addressed in [**8**]) — complications that the present author suspects vanish in a distributive setting.

Acknowledgments. Although the present paper was completed in Stuttgart, much of the work was carried out while I was a visiting researcher at ITK, Tilburg, where I profited from discussions with Emiel Krahmer. I also thank David Israel and Fernando Pereira for comments following my talk on this subject at the fourth conference on situation theory and its applications.

References

[1] N. Asher and H. Kamp. Self-reference, attitudes, and paradox. In G. Chierchia, B.H. Partee, and R. Turner, editors, *Properties, Types and Meaning*, volume 1. Kluwer Academic Publishers, Dordrecht, 1989.

[2] Nicholas Asher. *Reference to Abstract Objects in Discourse.* Kluwer Academic Publishers, Dordrecht, 1993.

[3] Ermanno Bencivenga. Free logics. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 3. Reidel, Dordrecht, 1986.

[4] Johan van Benthem. Logic and the flow of information. In D. Prawitz et al., editor, *Proc. 9th International Congress of Logic, Methodology and Philosophy of Science.* North-Holland, Amsterdam, 1991.

[5] Robin Cooper. Situation theoretic discourse representation theory. Manuscript, HCRC, Edinburgh, 1992.

[6] Tim Fernando. What is a DRS? In R. Cooper and J. Groenendijk, editors, *Integrating Semantic Theories II.* Dyana deliverable R2.1.B, 1994. Presented in a Computational Semantics workshop at Tilburg, December 1994.

[7] J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14, 1991.

[8] J. Groenendijk, M. Stokhof, and F. Veltman. This might be it. In J. Seligman and D. Westerståhl, editors, *Logic, Language and Computation*, volume 1. CSLI Lecture Notes Number 58, Stanford, 1996. Also appears in Dyana deliverable R2.1.B, *Integrating Semantic Theories II*, (R. Cooper and J. Groenendijk, editors), 1994.

[9] David Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2. Reidel, Dordrecht, 1984.

[10] Irene Heim. On the projection problem for presuppositions. In M. Barlow, D.P. Flickinger, and M.T. Westcoat, editors, *Proc. West Coast Conference on Formal Linguistics*, volume 2. Stanford Linguistics Association, California, 1983.

[11] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Language and Computation.* Addison-Wesley, 1979.

[12] David Israel. The very idea of dynamic semantics. In P. Dekker and M. Stokhof, editors, *Proc. Ninth Amsterdam Colloquium.* ILLC, University of Amsterdam, 1994.

[13] H. Kamp and U. Reyle. *From Discourse to Logic.* Kluwer Academic Publishers, Dordrecht, 1993.

[14] Hans Kamp. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof, editors, *Formal Methods in the Study of Language.* Mathematical Centre Tracts 135, Amsterdam, 1981.

[15] H. Jerome Keisler. Forcing and the omitting types theorem. In M. Morley, editor, *Studies in Model Theory.* The Mathematical Association of America, 1973.

[16] David Nelson. Constructible falsity. *J. Symbolic Logic*, 14, 1949.

[17] P. Pagin and D. Westerståhl. Predicate logic with flexibly binding operators and natural language semantics. *J. Logic, Language and Information*, 2(2), 1993.

[18] H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics.* Warsaw, 1963.

[19] Frank Veltman. Defaults in update semantics. In H. Kamp, editor, *Conditionals, Defaults and Belief Revision.* Edinburgh, Dyana deliverable R2.5.A, 1990.

6

Ambiguity under changing contexts

Abstract. Notions of disambiguation supporting a compositional interpretation of ambiguous expressions and reflecting intuitions about how sentences combine in discourse are investigated. Expressions are analyzed both inductively by breaking them apart, and co-inductively by embedding them within larger contexts.

1. Introduction

It is not clear how to distinguish ambiguous from unambiguous expressions without appealing to some process of disambiguation that both separates and links them. Nor is it clear why one would bother with expressions — be they ambiguous or not — if they effected *no* change in context (typically identified, for simplicity and concreteness, with the "common ground"). Insofar as disambiguation is one of the effects that asserting an expression can have, it is natural to study ambiguity within (so-called) "dynamic semantics" (e.g., Heim [**10**] and Kamp and Reyle [**12**]), central to which is a binary "merge" operation $^\wedge$ on formulas φ and ψ such that $\varphi^\wedge\psi$ represents the context obtained by asserting the second argument ψ at a(n initial) context represented by φ.[1] The present paper is an attempt to provide an analysis of expressions given at the level of sentences (or formulas) and sequences of sentences (i.e., discourse), respecting the two aforementioned intuitions,

> **Ambiguity as incomplete interpretation**: ambiguous expressions require disambiguation for an unambiguous interpretation

and

> **Semantics as change**: the meaning of a formula is the change in context it induces.

The aim is to develop these intuitions in accordance with the principle of compositionality (viz., the meaning of a whole is a function of the meaning of its parts) and with a view to reasoning with partial disambiguations.

1.1. Outline of the paper. Towards that end, various notions of disambiguation are considered below. The simplest, introduced in §2.1, is a binary relation on expressions that picks out an unambiguous formula for an ambiguous expression assembled from unambiguous ones through a binary logical connective $\bullet$. (The idea is to analyze an expression e that is ambiguous between two unambiguous formulas φ and φ' as $\varphi \bullet \varphi'$, rather than say, the

[1] The notation $^\wedge$ is meant to suggest a conjunction $\wedge$ lifted (!) to a dynamic setting, which — contrary to what the alternative notation $+$ (adopted by some) might imply — need not be commutative any more so than string concatenation.

unambiguous disjunction $\varphi \vee \varphi'$.) More generally, a disambiguation can be viewed as a mapping from an ambiguous expression to an unambiguous one. This notion is employed in §2.2 to extend a relation of logical consequence between unambiguous formulas to ambiguous expressions. All this presupposes, however, a unique choice of a disambiguation, whereas ambiguity arises (in practice) precisely when identifying *the* appropriate disambiguation (for a particular context) is problematic. Still, in introducing a notion of disambiguation, we have made some progress; to address ambiguity, our analyses of $\bullet$ and of ambiguous consequence relations need only be lifted from a single disambiguation to a family of disambiguations (say, through quantification). A family of disambiguations may, in turn, be viewed as a *partial* disambiguation, the semantics for which §2.3 begins to explore.

For a compositional semantic interpretation of ambiguous expressions, it turns out to be useful to build into the notion of a disambiguation a specification not only of *what* unambiguous formulas an expression can be reduced to, but also of *how* — say, by merging with other expressions — that proceeds. Accordingly, disambiguations are characterized in §3 (the main section of the present paper) as incremental processes, stretching forward and backward in time. Finally, §4 discusses how the various pictures of disambiguation complement each other and fit within dynamic semantics.

1.2. A few examples (for orientation). As the sequel is fairly abstract, some examples might be useful to keep in mind. Suppose the sentence

(A1) `Josef loves squash`

has exactly two unambiguous interpretations, given by

(U1) `Josef loves the sport of squash`

and

(U2) `Josef loves the vegetable squash.`

(For the sake of the argument, let us take the words `Josef` and `loves` to be unambiguous — implausible though that may be.) A "cooperative" speaker who utters (A1) is assumed to mean either (U1) or (U2), but not (U1) $\vee$ (U2). §2 below analyzes (A1) as (U1) $\bullet$ (U2), a formal (unambiguous) interpretation of which requires a choice — i.e., a disambiguation — between (U1) and (U2). Just what disambiguation is appropriate may be clear from the context. For instance, the choice (U1) is natural if the utterance of (A1) is preceded by

(U3) `Many people are fond of indoor sports.`

That is, (U1), as opposed to (U2), can be said to be a consequence of the discourse

(A2) `Many people are fond of indoor sports.`
`Josef loves squash.`

The proper disambiguation may also become evident further on in the discourse. In particular, (A1) presumably means (U2) within the two-sentence discourse formed by following (A1) up with

(A3) `He eats it raw.`

Moreover, in this case the disambiguation is *mutual*; (A1) also disambiguates the pronouns `he` and `it` in (A3). This gives rise to the complication involving acyclicity mentioned in §3, and, in addition, motivates an *underspecified representation* of ambiguity. In isolation, (A3) is infinitely ambiguous inasmuch as the pronouns `he` and `it` might mean any number of things. Rather than constructing (A3) from an infinitary version of $\bullet$, it is simpler to treat (A3) as an underspecified representation in its own right, alongside expressions such as (U1) $\bullet$ (U2). Finally, observe that ambiguities may stay unresolved indefinitely

(A4) `... Different things appeal to different people. Josef loves squash. Simone can't stand it. ...`

2. The necessity of contextual disambiguation

Let Φ be a set of formulas that are unambiguous insofar as each $\varphi \in \Phi$ has a well-defined semantic interpretation $[\![\varphi]\!]$. Suppose e is an expression that intuitively is ambiguous between two formulas φ and φ' in Φ. Perhaps the most simple-minded way of analyzing e is to equate it with the disjunction $\varphi \vee \varphi'$, under the assumption that there is a fixed binary operation $F_\vee$ (such as set union) relative to which $\varphi_1 \vee \varphi_2$ is to be interpreted as $F_\vee([\![\varphi_1]\!], [\![\varphi_2]\!])$, for all φ_1 and $\varphi_2 \in \Phi$. But then where in this picture does disambiguating information enter? If the interpretation of $\varphi \vee \varphi'$ is determined *unambiguously* by the interpretations $[\![\varphi]\!]$ and $[\![\varphi']\!]$ of the disjuncts φ and φ' respectively, then does reducing e to $\varphi \vee \varphi'$ not abstract away the ambiguity in e? It is plausible enough that ambiguous expressions must somehow be converted into unambiguous formulas. What is disturbing about this disjunctive analysis, however, is that it carries out the conversion, with*out* calling for any disambiguating information. This objection applies (as pointed out by an anonymous referee) to any fixed (unambiguous) combination of φ and φ', not just disjunction. Disjunction has attracted, however, by far the greatest attention (and criticism) in the literature (see, for example, van Deemter [**4**], Poesio [**14**] and other articles in that volume), suggesting consequence relations between ambiguous expressions that will (among other things) be considered in this section.

2.1. Choosing between unambiguous formulas. Bringing disambiguations into the picture, consider a binary logical connective $\bullet$, interpreted relative to a binary relation $\prec$ on formulas that chooses between its arguments φ and φ' according say, to[2]

$$[\![\varphi \bullet \varphi']\!]_\prec \;=\; \begin{cases} [\![\varphi]\!] & \text{if } \varphi \prec \varphi' \\ [\![\varphi']\!] & \text{otherwise.} \end{cases} \tag{1}$$

[2] A similar "meta-or" connective is proposed in p.234 of van Deemter [**4**], the novelty of the present approach being to make the disambiguation $\prec$ explicit.

The plain intuition in (1) is that $\varphi \prec \varphi'$ means: choose φ over φ'. In the sense that (1) forces a choice between $[\![\varphi]\!]$ and $[\![\varphi']\!]$, the relations $\prec$ appropriate for (1) represent *total* disambiguations. We will consider partial disambiguations shortly, but it will prove instructive to first look carefully at (1), and to formulate it more generally as follows.

Definition 1. *Let $\Phi^\bullet$ be the set of expressions obtained from Φ by adding a closure condition for $\bullet$ to those for Φ. Given a binary relation $\prec$ on $\Phi^\bullet$, define the function $\cdot_\prec : \Phi^\bullet \to \Phi$ inductively by*

$$
\begin{array}{rcll}
\varphi_\prec & = & \varphi & \text{for } \varphi \in \Phi \\
(e \bullet e')_\prec & = & \left\{ \begin{array}{ll} e_\prec & \text{if } e \prec e' \\ e'_\prec & \text{otherwise} \end{array} \right. & \\
\gamma(e, \ldots, e')_\prec & = & \gamma(e_\prec, \ldots, e'_\prec) & \text{for each connective } \gamma \text{ in } \Phi \ .
\end{array} \tag{2}
$$

Given Definition 1, a function $[\![\cdot]\!]$ interpreting Φ can be extended to a function $[\![\cdot]\!]_\prec$ interpreting $\Phi^\bullet$ by composing $\cdot_\prec$ with $[\![\cdot]\!]$,

$$\text{for every } e \in \Phi^\bullet, \quad [\![e]\!]_\prec \ = \ [\![e_\prec]\!] \ ,$$

thereby generalizing line (1) smoothly to

$$[\![e \bullet e']\!]_\prec \ = \ \left\{ \begin{array}{ll} [\![e]\!]_\prec & \text{if } e \prec e' \\ [\![e']\!]_\prec & \text{otherwise} \end{array} \right.$$

for all e and $e' \in \Phi^\bullet$ (and similarly for other logical connectives such as $\vee$, $[\![e \vee e']\!]_\prec = F_\vee([\![e]\!]_\prec, [\![e']\!]_\prec)$). Now, to isolate the essential contribution that $\prec$ makes to $[\![\cdot]\!]_\prec$, define the binary relation $\prec^\bullet$ on $\Phi^\bullet$ by

$$e \prec^\bullet e' \quad \text{iff} \quad [\![e]\!]_\prec \neq [\![e']\!]_\prec \text{ and } [\![e \bullet e']\!]_\prec = [\![e]\!]_\prec$$

for all $e, e' \in \Phi^\bullet$. It is a routine matter to prove

Proposition 1. *Let $\prec$ and $\prec'$ be binary relations on $\Phi^\bullet$. Then*

(i) $[\![\cdot]\!]_\prec = [\![\cdot]\!]_{\prec^\bullet}$

(ii) $[\![\cdot]\!]_\prec = [\![\cdot]\!]_{\prec'}$ *iff* $\prec^\bullet = \prec'^\bullet$

(iii) $\prec^\bullet$ *is an irreflexive relation contained in* $\prec$

and

(iv) *if*

$$[\![e \bullet e']\!]_\prec \ = \ [\![e' \bullet e]\!]_\prec \tag{3}$$

holds for all e and $e' \in \Phi^\bullet$, then $\prec^\bullet$ is asymmetric (i.e., $e \prec^\bullet e'$ implies not $e' \prec^\bullet e$).

Note that the commutative principle (3) is faithful to the intuition underlying (1) that $\prec$ specifies a choice between φ and φ' to disambiguate the expression $\varphi \bullet \varphi'$. Faithful too is the associative principle

$$[\![(e \bullet e') \bullet e'']\!]_\prec \ = \ [\![e \bullet (e' \bullet e'')]\!]_\prec \tag{4}$$

which holds for every transitive relation $\prec$. Unfortunately, (4) does not seem sufficient to guarantee that $\prec^\bullet$ is transitive; it is easy enough to see that

$e \prec^\bullet e'$ and $e' \prec^\bullet e''$ imply, in the presence of (4), that $[\![e \bullet e'']\!]_\prec = [\![e]\!]_\prec$; but what is lacking to conclude $e \prec^\bullet e''$ is that $[\![e]\!]_\prec \neq [\![e'']\!]_\prec$. Accordingly, let us define (given $\prec$) a slightly different relation $\preceq^\bullet$ on $\Phi^\bullet$ by

$$e \preceq^\bullet e' \quad \text{iff} \quad [\![e \bullet e']\!]_\prec = [\![e]\!]_\prec$$

for all $e, e' \in \Phi^\bullet$.

Proposition 2. *Let $\prec$ and $\prec'$ be binary relations on $\Phi^\bullet$. Then*

(i) $[\![\cdot]\!]_\prec = [\![\cdot]\!]_{\preceq^\bullet}$
(ii) $[\![\cdot]\!]_\prec = [\![\cdot]\!]_{\prec'}$ *iff* $\preceq^\bullet = \preceq'^\bullet$
(iii) $\preceq^\bullet$ *is a reflexive relation contained in* $\prec \;\cup\; \{(e, e') \in \Phi^\bullet \times \Phi^\bullet : [\![e]\!]_\prec = [\![e']\!]_\prec\}$.
(iv) *if* (3) *holds then*

$$e \preceq^\bullet e' \text{ and } e' \preceq^\bullet e \quad \text{imply} \quad [\![e]\!]_\prec = [\![e']\!]_\prec$$

and

(v) *if* (4) *holds for all* e, e' *and* $e'' \in \Phi^\bullet$, *then* $\preceq^\bullet$ *is transitive.*

The modifications in Proposition 2 to parts (iii) and (iv) of Proposition 1 are somewhat minor. What is perhaps more troubling is that missing from Proposition 2 is an assertion stating the existence of a relation $\leq$ on the range of $[\![\cdot]\!]$ such that

$$\preceq^\bullet \;=\; \{(e, e') \in \Phi^\bullet \times \Phi^\bullet \;:\; [\![e]\!]_\prec \leq [\![e']\!]_\prec\} \tag{5}$$

(or, in Proposition 1, some relation $<$ similarly characterizing $\prec^\bullet$). Further conditions on $\prec$ need to be made in order to guarantee the existence of a relation $\leq$ satisfying (5). This matter will be taken up in §2.3, but for now note that (5) is, in general, implausible, given line (2). Indeed, the syntactic and total character of disambiguation in (2) suggests that it is natural to assume that $\prec$ is a linear order on $\Phi^\bullet$ (inasmuch as the linearity of $\prec$ corresponds to the totality of disambiguation). Such an assumption can, under (3) and (4), be made without loss of generality, since an arbitrary binary relation $\prec$ satisfying (3) and (4) can be turned into a linear order inducing the same interpretation $[\![\cdot]\!]_\prec$, by linearizing $\preceq^\bullet$. Conversely, a linear order $\prec$ validates (3) and (4), whereas (3) fails for a (non-linear) partial order that contains neither (φ, φ') nor (φ', φ), for some φ and $\varphi' \in \Phi$ such that $[\![\varphi]\!] \neq [\![\varphi']\!]$.

Whether or not we agree that the "natural" relations $\prec$ relative to which (1) interprets $\bullet$ are linear orders, the crucial point is that $\bullet$ requires some binary relation $\prec$ for its interpretation, whereas an unambiguous connective such as $\vee$ does not (except possibly in interpreting disjuncts constructed with $\bullet$). To see why this is so "crucial," let us consider how entailment might be defined between ambiguous expressions.

2.2. A generalization and some consequences. Suppose the set Φ of unambiguous formulas is equipped with a binary relation $\vdash$ on Φ of logical consequence. (E.g., $\varphi \vdash \varphi'$ iff $[\![\varphi \vee \varphi']\!] = [\![\varphi']\!]$.) A natural question to ask is how to lift $\vdash$ to some set $E \supset \Phi$ of possibly ambiguous expressions. (E.g.,

$E = \Phi^\bullet$, although this identification is not necessary.) A number of proposals in the literature (e.g., van Deemter [**4**], Reyle [**15**]) are based on a function $\hat{\delta} : E \to \mathsf{Pow}(\Phi)$ specifying the set $\hat{\delta}(e)$ of unambiguous formulas to which an expression e can be disambiguated. (E.g., $\hat{\delta}(\varphi \bullet \varphi') = \{\varphi, \varphi'\}$.) The relation $\vdash$ can then be extended to E relative to a choice of quantifiers $Q, Q' \in \{\forall, \exists\}$ by stipulating

$$(6) \qquad e \vdash_{QQ'} e' \quad \text{iff} \quad (Q\varphi \in \hat{\delta}(e))(Q'\varphi' \in \hat{\delta}(e'))\ \varphi \vdash \varphi'$$

for all $e, e' \in E$. In particular, the relation $\vdash_{\forall\exists}$ arises from a disjunctive analysis of an expression e as $\bigvee \hat{\delta}(e)$

$$\bigvee \hat{\delta}(e) \vdash \bigvee \hat{\delta}(e') \quad \text{iff} \quad (\forall\varphi \in \hat{\delta}(e))(\exists\varphi' \in \hat{\delta}(e'))\ \varphi \vdash \varphi' \ .$$

As noted in van Deemter [**4**] (p. 207), the schemes $\vdash_{QQ'}$ carry none of the "connections between interpretive choices" argued (also in Reyle [**16**]) to be essential to disambiguation — by which is meant that in (6), the choice of $\varphi \in \hat{\delta}(e)$ need not constrain the choice of $\varphi' \in \hat{\delta}(e')$ in the following sense. Recalling Definition 1 from §2.1, we might (instead) extend $\vdash$ to $\Phi^\bullet$ relative to a binary relation $\prec$ by stipulating that for all $e, e' \in \Phi^\bullet$,

$$e \vdash^{\prec} e' \quad \text{iff} \quad e_{\prec} \vdash e'_{\prec} \ ,$$

where, in contrast to (6), the choices $e_\prec \in \hat{\delta}(e)$ and $e'_\prec \in \hat{\delta}(e')$ are "connected" by $\prec$. Now, it is true enough that $\vdash^\prec$ is hardly an alternative to $\vdash_{\forall\exists}$ since (the choice of a particular) $\prec$ resolves away ambiguity. On the other hand, ambiguity can easily be re-introduced by collecting together a family $\mathcal{O}$ of relations $\prec$ and eliminating the dependence on a particular $\prec$ as follows. For all $e, e' \in \Phi^\bullet$, form the set

$$\{\prec \in \mathcal{O} \ : \ e \vdash^{\prec} e'\}$$

and check if that set is a sufficiently large subset of $\mathcal{O}$. Just what counts as "sufficiently large" depends on how loose or strict we want our ambiguous consequence relation to be. More precisely, the idea is to collect these sufficiently large sets in a family q of subsets of $\mathcal{O}$, and to extend $\vdash$ to $\Phi^\bullet$ relative to q by defining

$$(7) \qquad e \vdash^{q} e' \quad \text{iff} \quad \{\prec \in \mathcal{O} \ : \ e \vdash^{\prec} e'\} \in q$$

for all $e, e' \in \Phi^\bullet$. The difference between (6) and (7) is not only that the "interpretive choices" of the antecedent e and succedent e' are "connected" in (7) by $\prec$, but also that Q and Q' in $\vdash_{QQ'}$ have been absorbed into one parameter q in $\vdash^q$. Is there a way of deriving the relations $\vdash_{QQ'}$ (for whatever they are worth) using (7)? The answer is yes, provided we generalize our notion of disambiguations $\prec$ to make them dependent on context.

The notion of context dependence involved here can, in fact, be motivated by the discourse (A2) in §1.2, the point being that

(∗) within (A2), (U3) provides a context for which the natural reading of (U1) • (U2) is (U1).

To formalize $(*)$, let us choose the binary relation interpreting $\bullet$ according to the relevant context φ. Writing $\prec^{\varphi}$ for that relation, $(*)$ can then be encoded by stipulating

$$\text{(U1)} \quad \prec^{\text{(U3)}} \quad \text{(U2)}\ .$$

More generally, without assuming that $E = \Phi^{\bullet}$, consider a function $\delta : (\Phi \times E) \to \Phi$ supporting the intuition that for every $\varphi \in \Phi$ and every $e \in E$,

$$\delta(\varphi, e) \quad \approx \quad \text{an unambiguous formula to which } e \text{ can be disambiguated, given the "context" } \varphi\ ,$$

so that, in particular, for every $\varphi' \in \Phi$,

$$\delta(\varphi, \varphi') \quad = \quad \varphi'\ .$$

For the specific case of $E = \Phi^{\bullet}$, such a function δ might (recalling Definition 1) be given by

(8) $$\delta(\varphi, e) \quad = \quad e_{\prec^{\varphi}}$$

for an appropriate function mapping formulas $\varphi \in \Phi$ to binary relations $\prec^{\varphi}$ on $\Phi^{\bullet}$.

Definition 2 (Part 1). *Given a binary relation $\vdash$ on Φ, and a function $\delta : (\Phi \times E) \to \Phi$, let $\vdash^{\delta}$ be the binary relation on E defined by*

(9) $$e \vdash^{\delta} e' \quad \textit{iff} \quad \delta(\top, e) \vdash \delta(\delta(\top, e), e')$$

where $\top$ is a tautology (i.e., for all $\varphi \in \Phi$, $\varphi \vdash \top$).

The idea behind (9) is to disambiguate the succedent e' relative to the context $\delta(\top, e)$ induced by the disambiguation of the antecedent. The antecedent is assumed, for the sake of simplicity, to be disambiguated relative to the vacuous context $\top$. Such an assumption is harmless, as disambiguation relative to a context ψ can be carried out instead through a function $\delta' : (\Phi \times E) \to \Phi$ related to δ by $\delta'(\varphi, e) = \delta(\psi \wedge \varphi, e)$ for all $\varphi \in \Phi$ and $e \in E$. ($\wedge$ is the merge operation mentioned in the introduction.)

Having extended $\vdash$ to E relative to a fixed function $\delta : (\Phi \times E) \to E$, let us introduce some variation, by fixing a set Δ of such functions and a family $\mathcal{D}$ of ("sufficiently large") subsets of Δ, passing from $\vdash^{\delta}$ to $\vdash^{\mathcal{D}}$ as in (7).

Definition 2 (Part 2). *Given a set $\Delta \subseteq (\Phi \times E) \to \Phi$ of functions from $\Phi \times E$ to Φ, and a family $\mathcal{D}$ of subsets of Δ, let $\vdash^{\mathcal{D}}$ be the binary relation on E (extending the binary relation $\vdash$ on Φ) satisfying*

(10) $$e \vdash^{\mathcal{D}} e' \quad \textit{iff} \quad \{\delta \in \Delta \ :\ e \vdash^{\delta} e'\} \ \in \ \mathcal{D}$$

for all $e, e' \in E$.

It is now a simple exercise to formulate the aforementioned notions $\vdash_{QQ'}$ as $\vdash^{\mathcal{D}}$.

Proposition 3. *Given a function $\hat{\delta} : E \to \mathsf{Pow}(\Phi)$, let Δ be the set of (all) functions $\delta : (\Phi \times E) \to \Phi$ such that for all $\varphi \in \Phi$ and $e \in E$, $\delta(\varphi, e) \in \hat{\delta}(e)$.*

Then the relations $\vdash_{QQ'}$ *and* $\vdash^{\mathcal{D}}$*, specified (respectively) by* (6) *and* (10)*, coincide, assuming that*

(i) *for* $Q = Q' = \forall$, $\mathcal{D}$ *is* $\{\Delta\}$

(ii) *for* $Q = Q' = \exists$, $\mathcal{D}$ *is* $\{D \subseteq \Delta : D \neq \emptyset\}$

(iii) *for* $Q = \forall$ *and* $Q' = \exists$, $\mathcal{D}$ *is the set of all sets* $D \subseteq \Delta$ *such that*

$$(\forall e \in E) \qquad \hat{\delta}(e) \;=\; \{\delta(\top, e) \;:\; \delta \in D\}$$

and

(iv) *for* $Q = \exists$ *and* $Q' = \forall$, $\mathcal{D}$ *consists of all sets* $D \subseteq \Delta$ *such that for all* $e, e' \in E$*, there is a* $\delta \in D$ *such that*

$$\hat{\delta}(e') \;=\; \{\delta'(\delta'(\top, e), e') \;:\; \delta' \in D \text{ and } \delta'(\top, e) = \delta(\top, e)\}\ .$$

Observe that in Proposition 3, $\hat{\delta}$ requires only of each $\delta \in \Delta$ that $\delta(\varphi, e) \in \hat{\delta}(e)$, imposing no constraints whatsoever on φ. This reflects the absence of "connections between interpretive choices" in $\vdash_{QQ'}$. The context-dependence of the disambiguations in $\mathcal{D}$ is employed in Proposition 3 only as a technical device to reduce the two parameters Q and Q' to a single parameter $\mathcal{D}$ (whose elements δ have an extra argument for contexts).

In accordance with the intuition that $\mathcal{D}$ consists of the "sufficiently large" subsets of Δ, it is natural to require of the family $\mathcal{D}$ in Definition 2 (Part 2) that

(C1) $\Delta \in \mathcal{D}$

(since $e \vdash^{\mathcal{D}} e'$ ought to hold if for every $\delta \in \Delta$, $e \vdash^{\delta} e'$) and

(C2) for all subsets D_1 and D_2 of Δ, if $D_1 \in \mathcal{D}$ and $D_1 \subseteq D_2$, then $D_2 \in \mathcal{D}$

(since $e_2 \vdash^{\mathcal{D}} e_2'$ should follow if $e_1 \vdash^{\mathcal{D}} e_1'$ and whenever $e_1 \vdash^{\delta} e_1'$, $e_2 \vdash^{\delta} e_2'$). Note that for each of the four cases in Proposition 3, $\mathcal{D}$ meets conditions (C1) and (C2).

Reflecting further on Definition 2, observe that each disambiguation $\delta \in \Delta$ is total insofar as the range of δ is Φ. A notion of partial disambiguation is, however, implicit in $\mathsf{Pow}(\Delta)$; that is, a subset D of Δ can be regarded as a partial disambiguation (under the common reduction of a partial object to the set of its completions). Indeed, $\mathcal{D}$ can be thought of simply as the set of partial disambiguations D covering enough of Δ such that for all $e, e' \in E$, if for every $\delta \in D$, $e \vdash^{\delta} e'$, then $e \vdash^{\mathcal{D}} e'$. Given that, in practice, disambiguation is partial, it makes sense to step one level up in abstraction and reconsider our extensions of $[\![\cdot]\!]$ to $\Phi^{\bullet}$, by passing to a set $\mathcal{O}$ of binary relations $\prec$ on $\Phi^{\bullet}$. As it turns out, this leads us back to the issue (broached in line (5) in §2.1) of compositional semantics.

2.3. Partial disambiguations and semantic interpretations. The basic problem in generalizing (1) to a notion of partial (rather than total) disambiguation, given by a set $\mathcal{O}$ of binary relations on $\Phi^{\bullet}$, is indicated by the question mark "?" in

$$[\![\varphi \bullet \varphi']\!]_{\mathcal{O}} \;=\; \begin{cases} [\![\varphi]\!] & \text{if for every } \prec \text{ in } \mathcal{O},\ \varphi \prec \varphi' \\ [\![\varphi']\!] & \text{if for every } \prec \text{ in } \mathcal{O},\ \varphi' \prec \varphi \\ ? & \text{otherwise.} \end{cases}$$

One way to attack "?" is to weaken "every" to some quantifier Q (such as "most")[3]

$$[\![\varphi \bullet \varphi']\!]_{\mathcal{O}} \;=\; \begin{cases} [\![\varphi]\!] & \text{if for Q } \prec \text{ in } \mathcal{O},\ \varphi \prec \varphi' \\ [\![\varphi']\!] & \text{if for Q } \prec \text{ in } \mathcal{O},\ \varphi' \prec \varphi \\ ? & \text{otherwise.} \end{cases}$$

While this may narrow the gap, it will not (in general) close it, especially if we insist that the choice between φ and φ' be "fair" in the sense that $[\![\varphi \bullet \varphi']\!]_{\mathcal{O}} = [\![\varphi' \bullet \varphi]\!]_{\mathcal{O}}$. Of course, rather than choosing between φ and φ', we might simply fill the gap with say, $[\![\varphi \vee \varphi']\!]$ or even $[\![\varphi \wedge \varphi']\!]$. To see why this might not be so far-fetched, let us reconsider $[\![\varphi \bullet \varphi']\!]_{\prec}$ and line (1).

The introduction in (1) of the relation $\prec$ is broadly reminiscent of the addition of an accessibility relation in Kripke's semantics for modal logic, an essential difference being that $\prec$ is a relation on formulas, while the accessibility relation is one on worlds. (In this sense, the interpretation of $\bullet$ is just an intensional variation on Kripke's semantics.) This difference raises the question of analyzing $\bullet$ compositionally under $[\![\cdot]\!]_{\prec}$; that is,

is there a function $F_\bullet$ such that for all e and $e' \in \Phi^\bullet$,

$$[\![e \bullet e']\!]_{\prec} = F_\bullet([\![e]\!]_{\prec}, [\![e']\!]_{\prec}) \;?$$

Let us proceed carefully, and spell out an assumption that we have been making all along; the unambiguous nature of formulas in Φ is construed as supporting

$$\text{Assumption A:} \qquad [\![\varphi]\!]_{\prec} = [\![\varphi]\!] \quad \text{for every } \varphi \in \Phi\ .$$

Confining our attention to unambiguous formulas φ, φ', ψ and ψ' ($\in \Phi$), and agreeing that

$$\varphi \equiv_0 \psi \quad \text{means} \quad [\![\varphi]\!] = [\![\psi]\!]\ ,$$

a necessary condition for the existence of $F_\bullet$ is that

$$\varphi \equiv_0 \psi \ \text{ and } \ \varphi' \equiv_0 \psi' \quad \text{imply} \quad [\![\varphi \bullet \varphi']\!]_{\prec} = [\![\psi \bullet \psi']\!]_{\prec}\ .$$

This, in turn, suggests (under Assumption A) *closing* $\prec$ as follows

(11) $$\varphi \prec \varphi' \ \text{ and } \ \varphi \equiv_0 \psi \not\equiv_0 \varphi' \equiv_0 \psi' \quad \text{imply} \quad \psi \prec \psi'\ .$$

But if $\psi' \prec \psi$ (as it well could), then (11) would mean that $[\![\varphi \bullet \varphi']\!]_{\prec} \neq [\![\varphi' \bullet \varphi]\!]_{\prec}$, which is a rather high price to pay for compositionality. The problem is that the conjunction $\varphi \prec \varphi'$ and $\varphi' \prec \varphi$ is compatible with (1) only as long as $\varphi \equiv_0 \varphi'$, which is the case with linear orders (strict or otherwise). As the

[3] How the choice of Q here relates to the parameter q in $\vdash^q$ (from §2.2) is an obvious question not investigated in the present paper.

closure condition suggested by (11) may take us outside the realm of linear orders, it is appropriate to reconsider (1). To the extent that $\prec$ is intended to disambiguate *totally*, our modification of (1) should still amount to composing some syntactic function $\cdot_{\prec} : \Phi^{\bullet} \to \Phi$ (that disambiguates totally, in that its range is Φ) with $[\![\cdot]\!]$. Isolating line (2) of Definition 1, let us weaken it to

$$(e \bullet e')_{\prec} \quad = \quad \begin{cases} e_{\prec} & \text{if } e \prec e' \text{ but not } e' \prec e \\ e'_{\prec} & \text{if } e' \prec e \text{ but not } e \prec e' \\ ? & \text{otherwise} \end{cases} ,$$

writing, as before, "?" to indicate doubt. Casting aside such doubts, suppose

$$(e \bullet e')_{\prec} \quad = \quad \begin{cases} e_{\prec} \vee e'_{\prec} & \text{if neither } e \prec e' \text{ nor } e' \prec e \\ e_{\prec} \wedge e'_{\prec} & \text{if both } e' \prec e \text{ and } e \prec e' \end{cases} ,$$

modeling (as it were) indifference by disjunction, and conflicts in choices by conjunction. Now, such schemes may well seem unnecessarily complicated, designed for poorly chosen total disambiguations that we would be better off without. But even if we start with "good" linear orders $\prec$ (suited for the simple scheme (1)), the question of defining $[\![\cdot]\!]_{\mathcal{O}}$ relative to a set $\mathcal{O}$ of such linear orders leads us back to such complications, if $\mathcal{O}$ is reduced either to $\bigcap \mathcal{O}$ or $\bigcup \mathcal{O}$ (the trouble being that neither the intersection nor the union of a set $\mathcal{O}$ of linear orders can be expected to be a linear order). Some motivations for and consequences of collapsing a set of linear orders on $\Phi^{\bullet}$ to a binary relation on $\Phi^{\bullet}$ are detailed in Fernando [**6**], where an arbitrary binary relation R on $\Phi^{\bullet}$ is modified in a minimal way to support a compositional analysis of $\bullet$ under a revised interpretation $[\![\cdot]\!]_R$ of (1) along the lines described above — i.e., using $\vee$ and $\wedge$.

But returning to the question of $[\![\cdot]\!]_{\mathcal{O}}$, does resolving "?" through $\vee$ or $\wedge$ (or any unambiguous construct, for that matter) do justice to the intuition that $\mathcal{O}$ is a *partial* disambiguation? What is so troubling here is that within the scheme sketched above, $[\![\cdot]\!]_{\mathcal{O}}$ is the composition $[\![\cdot_{\mathcal{O}}]\!]$ of some function $\cdot_{\mathcal{O}} : \Phi^{\bullet} \to \Phi$ with $[\![\cdot]\!]$ (just as, in §2.1, $[\![\cdot]\!]$ is extended to $\Phi^{\bullet}$ by pre-composing it with the function $\cdot_{\prec}$ specified in Definition 1). The fact that the range of $\cdot_{\mathcal{O}}$ is Φ betrays the total, rather than partial, character of the disambiguation effected. The obvious alternative is to let the range of $\cdot_{\mathcal{O}}$ spill over into $\Phi^{\bullet}$, as in the following variation on Definition 1.

Definition 3. *Given a set $\mathcal{O}$ of binary relations on $\Phi^{\bullet}$, let $\cdot_{\mathcal{O}}$ be the function on $\Phi^{\bullet}$ given inductively by*

$$\varphi_{\mathcal{O}} \quad = \quad \varphi \qquad \textit{for } \varphi \in \Phi$$

$$(12) \qquad (e \bullet e')_{\mathcal{O}} \quad = \quad \begin{cases} e_{\mathcal{O}} & \textit{if } Q \prec \textit{ in } \mathcal{O},\ e \prec e' \\ e'_{\mathcal{O}} & \textit{if } Q \prec \textit{ in } \mathcal{O},\ e' \prec e \\ e_{\mathcal{O}} \bullet e'_{\mathcal{O}} & \textit{otherwise.} \end{cases}$$

$$\gamma(e, \ldots, e')_{\mathcal{O}} \quad = \quad \gamma(e_{\mathcal{O}}, \ldots, e'_{\mathcal{O}}) \quad \textit{for every connective/quantifier } \gamma \textit{ in } \Phi \ .$$

For definiteness, the quantifier Q in (12) might be set simply to $\forall$. The main point is that (12) gets rid of as many occurrences of $\bullet$ as it can (under some choice of Q), but might still leave some behind. Now, because $\cdot_{\mathcal{O}}$ may output an expression in $\Phi^{\bullet} - \Phi$, it cannot simply be composed with the interpretation function $[\![\cdot]\!]$ for Φ. Instead, writing V for the range of the function $[\![\cdot]\!]$ and $\mathbf{D}$ for the set of partial disambiguations $\mathcal{O}$,[4] let us compose $\cdot_{\mathcal{O}}$ with a function $[\![\cdot]\!]^{\bullet} : \Phi^{\bullet} \to X$, where X satisfies the reflexive domain equation

$$X = V + (\mathbf{D} \to X) \tag{13}$$

and

$$[\![\varphi]\!]^{\bullet} = [\![\varphi]\!] \quad \text{for } \varphi \in \Phi \tag{14}$$

$$[\![e \bullet e']\!]^{\bullet} = \text{the function } \beta : \mathbf{D} \to X \text{ such that } \beta(\mathcal{O}') = [\![(e \bullet e')_{\mathcal{O}'}]\!]^{\bullet} \tag{15}$$

$$[\![\gamma(e, \ldots, e')]\!]^{\bullet} = F_{\gamma}^{X}([\![e]\!]^{\bullet}, \ldots, [\![e']\!]^{\bullet}) \quad \text{for each connective/quantifier } \gamma \text{ in } \Phi\ . \tag{16}$$

(16) requires that each F_{γ} (interpreting γ relative to $[\![\cdot]\!]$) be lifted to an operation F_{γ}^{X} on X,[5] while (15) simply stalls the disambiguation of $e \bullet e'$, leaving to some $\mathcal{O}' \in \mathbf{D}$ its (partial) disambiguation (as described in (12)). As the references to $\mathbf{D}$ in (13) and (15) make clear, syntax plays a prominent role in $[\![\cdot]\!]^{\bullet}$. Can we wash away such traces of syntax (in $[\![\cdot]\!]^{\bullet}$) by replacing expressions e with their semantic values $[\![e]\!]^{\bullet}$?

This is a delicate question already for unambiguous formulas. Their interpretation under line (14) is just Assumption A adapted to partial disambiguations $\mathcal{O}$. But disambiguation, whether it be partial or total, may require of an unambiguous formula φ more than the truth-conditional content typically comprising $[\![\varphi]\!]$. Consider again the example (U1) $\bullet$ (U2) mentioned in §1.2, comparing its disambiguation within

(A5) `Many people at work are fond of all kinds of vegetables. Many also are sports enthusiasts. Dorit likes swimming. Josef loves squash.`

to that within

(A6) `Many people at work are sports enthusiasts. Dorit likes swimming. Many also are fond of all kinds of vegetables. Josef loves squash.`

[4] E.g., $\mathbf{D} = \mathsf{Pow}(\mathsf{Pow}(\Phi^{\bullet} \times \Phi^{\bullet}))$, although we might be pickier and accept only certain families of certain binary relations on $\Phi^{\bullet}$.

[5] Let us not get distracted here by the details of how to implement (13) and (16), as an altogether different approach will soon be pursued.

It is natural to assume that (U1)$\bullet$ (U2) should be disambiguated as (U1) in (A5), and (U2) in (A6), even though the first three sentences of (A5) and (A6) may well be reduced to unambiguous formulas equated by $[\![\cdot]\!]$.[6] Of course, there would be no problem, should $[\![\cdot]\!]$ happen to be sufficiently fine-grained; but then how would we know that?

Caution suggests that we suspend any commitment to (14) or Assumption A, and that we consider a more general approach to compositionality covering the possibility that the mapping $[\![\cdot]\!]$ abstracts away too much for disambiguation. Towards that end, another shift in our conception of disambiguation is useful — a shift that merges the context-dependence mentioned in §2.2 (and more recently brought out by (A5) and (A6) above) with the partiality hinted at by Definition 3. Taking up context-dependence first, let us extend the binary "merge" operation $^\wedge$ on Φ to a binary operation $^{\wedge_a}$ so that our analysis in §2.2 of the discourse (A2) can be reformulated as

$$\text{(U3)}\ ^{\wedge_a}\ (\text{(U1)} \bullet \text{(U2)}) \;\; = \;\; \text{(U3)}^\wedge\text{(U1)}\ .$$

That is, relating $^{\wedge_a}$ to line (8) of §2.2,

$$\text{(U3)}\ ^{\wedge_a}\ (\text{(U1)} \bullet \text{(U2)}) \;\; = \;\; \text{(U3)}\ ^\wedge\ (\text{(U1)} \bullet \text{(U2)})_{\prec^{(\text{U3})}}\ ,$$

where (U1) $\prec^{(\text{U3})}$ (U2). The result of applying $\varphi^{\wedge_a} e$ is, in this case, an unambiguous formula (in Φ). In contrast to §§2.1 and 2.2, however, we will not insist that the result of such an operation always be unambiguous, construing the disambiguation effected by $^{\wedge_a}$ instead as partial. This would allow, for instance, $^{\wedge_a}$ to preserve the lexical ambiguity of word `squash` in the following fragment of the discourse (A4) in §1.2

(A7) `Josef loves squash. Simone can't stand it.`

(A7) suggests the possibility that the first argument of $^{\wedge_a}$ (also) be ambiguous. Furthermore, that ambiguity can be resolved in cases such as the discourse formed in §1.2 from (A1) and (A3), where the disambiguation is mutual

$$(\text{(U1)} \bullet \text{(U2)})\ ^{\wedge_a}\ \text{(A3)} \;\; = \;\; \text{(U2)}^\wedge(\texttt{Josef eats the vegetable squash raw})\ .$$

In general then, let us take $^{\wedge_a}$ to be some binary operation extending $^\wedge$ to E that partially, if not completely, disambiguates the combination of its two arguments.

But where does $^{\wedge_a}$ come from, and what exactly does a mean? We will put these questions off to §4, and only assume, for now through the next section, that such an operation on E is, along with the interpretation $[\![\cdot]\!]$ of the set Φ of unambiguous formulas, given (somehow). We will content ourselves with the challenge of defining an interpretation $[\![e]\!]_a$ of an expression e that supports a compositional interpretation of $^{\wedge_a}$ (and perhaps other functions on

[6]Perhaps more familiar and convincing examples of the extra-truth-conditional requirements of disambiguation involve pronominal anaphora resolution; the disambiguation of a pronoun such as `he` is obviously sensitive to (among other things) the order in which discourse referents are introduced.

E), incorporating, if necessary, more structure (for $e = \varphi \in \Phi$) into $[\![\varphi]\!]_a$ than $[\![\varphi]\!]$. That is, rather than tampering with a given notion of disambiguation, we will refine the equivalence induced by $[\![\cdot]\!]$ (and denoted $\equiv_0$ above), possibly at the expense of (14) and Assumption A.

3. Broadening the context for compositionality

Let $E \supset \Phi$ be a set of expressions that include not only every $\varphi \in \Phi$, subject to an "interpretation" function $[\![\cdot]\!] : \Phi \to V$ (to a set V of "semantic values"), but also ambiguous expressions (not necessarily constructed from unambiguous formulas [in Φ] through $\bullet$). For the record, let us state that a function $[\![\cdot]\!]_a$ with domain E is *compositional with respect to* (or is said to "support a compositional interpretation of") a family $\mathcal{F}$ of (multi-ary) functions on E if for every natural number k and every k-ary function $f : E^k \to E$ on E in $\mathcal{F}$, there is a k-ary function F_f on the range of $[\![\cdot]\!]_a$ such that for all $e_1, \ldots, e_k \in E$,

$$[\![f(e_1 \ldots, e_k)]\!]_a \quad = \quad F_f([\![e_1]\!]_a, \ldots, [\![e_k]\!]_a) \ . \tag{17}$$

In establishing the compositionality of $[\![\cdot]\!]_a$, it suffices to consider the equivalence relation $\equiv_a$ on E that captures the equations induced by $[\![\cdot]\!]_a$ as follows:

$$e \equiv_a e' \quad \text{iff} \quad [\![e]\!]_a = [\![e']\!]_a$$

for all $e, e' \in E$. An equivalence relation $\equiv$ on E is a *congruence relative to* a k-ary function f on E if

$$\frac{e_1 \equiv e_1' \qquad e_2 \equiv e_2' \qquad \cdots \qquad e_k \equiv e_k'}{f(e_1, e_2, \ldots, e_k) \equiv f(e_1', e_2', \ldots, e_k')}$$

for all $e_1, e_2, \ldots, e_k, e_1', e_2', \ldots, e_k' \in E$. Let us say that $\equiv$ is a *congruence relative to* $\mathcal{F}$ if it is a congruence relative to every function in $\mathcal{F}$. Clearly, a function $[\![\cdot]\!]_a$ on E is compositional with respect to $\mathcal{F}$ iff $\equiv_a$ is a congruence relative to $\mathcal{F}$. Conversely, given a congruence $\equiv_a$ (on E) relative to $\mathcal{F}$, we can define a function $[\![\cdot]\!]_a$ on E that is compositional with respect to $\mathcal{F}$ by (the "term model" construction)

$$[\![e]\!]_a \quad = \quad \{e' \in E \ : \ e \equiv_a e'\} \ .$$

The two extreme examples of equivalence relations on E, namely, the universal relation $E \times E$ and the equality relation on E, are congruences relative to $\mathcal{F}$ (for any $\mathcal{F}$). Neither describes a non-trivial equational theory over E; the latter adds nothing to syntactic identity (raising the question: what light does semantics throw on syntax, if it is just syntax renamed?[7]), whereas the former lumps together all expressions in a single undifferentiated mass. The question is what distinctions between expressions ought to be drawn by the semantic equivalence $\equiv_a$?

[7] In this regard, the present section (§3) should be contrasted with Zadrozny [**17**], where the equivalence induced is syntactic identity.

It is here where the interpretation $[\![\cdot]\!]$ of the subset Φ of (unambiguous formulas in) E enters the picture. If $[\![\cdot]\!]$ interprets two unambiguous formulas φ and φ' differently — i.e., $[\![\varphi]\!] \neq [\![\varphi']\!]$ —, then they ought not to be equated, qua expressions in E. (Note that this condition is a weakening of Assumption A from §2.3.) That is, the semantics $[\![\cdot]\!]_a$ of ambiguous expressions can be grounded on the interpretation $[\![\cdot]\!]$ of unambiguous formulas by requiring that $\equiv_a$ be (set-theoretically) contained in the equivalence relation

$$\begin{aligned} R \;\; = \;\; & \{(\varphi,\varphi') \in \Phi \times \Phi \;:\; [\![\varphi]\!] = [\![\varphi']\!]\} \;\cup \\ & ((E - \Phi) \times (E - \Phi)) \end{aligned} \tag{18}$$

on E. Assuming that we wish to make as *few* semantic distinctions as possible — otherwise, we may as well take $\equiv_a$ to be syntactic identity and go home —, the idea is to equate $\equiv_a$ with the *largest* equivalence relation on E contained in R that is a congruence relative to $\mathcal{F}$.

But just what, by the way, is $\mathcal{F}$? §3.1 proceeds without making any assumptions about $\mathcal{F}$. In §§3.2 and 3.3, we specialize on the case in which $\mathcal{F}$ is the singleton family $\{^{\wedge_a}\}$, where $^{\wedge_a}$ is the binary operation described at the end of §2.3. That choice can be motivated by considering how compositionality can be employed as a constraint on the interpretation $[\![e]\!]_a$ of an arbitrary expression e. To be specific, $[\![e]\!]_a$ can be put in either

(i) the lefthand side of (17) — i.e., taking e to be equal to $f(e_1, \ldots, e_k)$ for some k-ary function $f \in \mathcal{F}$ and $e_1, \ldots, e_k \in E$ —

or

(ii) the righthand side of (17) — i.e., choosing some k-ary function $f \in \mathcal{F}$ and $e_1, \ldots, e_k \in E$, where $e = e_i$ for some $i \in \{1, \ldots k\}$.

Option (i) analyzes e in terms of its parts, suggesting globally (i.e., for all e's) an inductive definition, starting from simple atomic expressions (the base case) and building up to more complicated ones given by $\mathcal{F}$ (the inductive case). It is perhaps fair to say that a good deal of the "computational" interest in compositionality lies in the way it is (in practice) established by such an inductive procedure. Nonetheless, a complementary *co-inductive* approach that is also illuminating is suggested by option (ii), under which the operation $^{\wedge_a}$ can be viewed as a "master" disambiguating function f that merges e with other expressions. For instance, whereas (A1) (from §1.2) is broken apart, under the inductive definitions of §2, as (U1)•(U2), it can also be embedded within larger discourses such as (A2) and (A4) (as well as (A5), (A6) and (A7), at the end of §2.3).[8] It is largely the latter (co-inductive) approach that the present section pursues.

[8]As options (i) and (ii) are just different sides of the one coin of compositionality, the division outlined (i.e., according to which side of (17) we put e) should not be taken too seriously. What ought *not* to be brushed aside lightly is the genuine technical difference between induction and co-induction (i.e., between least and greatest fixed points; see, e.g., Aczel [**1**]), which distinguishes §§2 and 3.

§3.1 is quite abstract and general, sidestepping a direct definition of a (semantic) interpretation $[\![e]\!]_a$ of expressions $e \in E$, by setting its sights more modestly on an equivalence relation $\equiv_a$ on E (which, as we have already noted, is sufficient for the purpose of establishing compositionality). The reader with little patience for abstractness is asked to focus here on the case where $\mathcal{F}$ is $\{{}^{\wedge_a}\}$ and the equivalence relation R is that induced by $[\![\cdot]\!]$ according to (18). That case will be fleshed out further in §3.2, which considers definitions of $[\![\cdot]\!]_a$ approximating the intuition that $[\![e]\!]_a$ is the set of disambiguations of e, for an appropriate notion of disambiguation based on ${}^{\wedge_a}$. Curiously, in looking more closely at ${}^{\wedge_a}$, §3.3 edges back to an inductive analysis, in which the additional generality of §3.1 is exploited (§4.1).

3.1. An abstract equivalence on expressions. Suppose that an equivalence relation R on E and a family $\mathcal{F}$ of multi-ary functions on E are given. Define the set of congruences relative to every function in $\mathcal{F}$ that are contained in R

$$\mathsf{Cng}[R,\mathcal{F}] \quad = \quad \{\equiv\ \subseteq R\ :\ \equiv \text{ is a congruence relative to } \mathcal{F}\}\ ,$$

with the idea of looking at its ($\subseteq$-)largest element, if it, in fact, exists. Let us first consider the simple case where $\mathcal{F}$ is a singleton $\{f\}$, and write $\mathsf{Cng}[R,f]$ for $\mathsf{Cng}[R,\{f\}]$.

Lemma 4. *For every unary $f : E \to E$ and equivalence relation R on E, $\mathsf{Cng}[R,f]$ has a largest element.*

Proof. The largest element $\equiv$ is easily constructed by co-induction (e.g., Aczel [**1**]):

$$\begin{aligned} \equiv_0 &= R \\ \equiv_{n+1} &= \{(e,e') \in\ \equiv_n\ :\ f(e) \equiv_n f(e')\} \\ \equiv &= \bigcap_{n<\omega} \equiv_n \ . \end{aligned}$$

Clearly, $\equiv$ must contain every element of $\mathsf{Cng}[R,f]$. Furthermore,

$$\begin{aligned} e \equiv e' \quad &\text{iff} \quad (\forall n)\ e \equiv_n e' \\ &\text{iff} \quad eRe' \text{ and } (\forall n)\ f(e) \equiv_n f(e') \\ &\text{iff} \quad eRe' \text{ and } f(e) \equiv f(e') \end{aligned}$$

for all $e, e' \in E$. $\dashv$

What if the function f is (say) binary? We might proceed as before, but now taking

$$\begin{aligned} \equiv_{n+1} \quad = \quad \{(e,e') \in\ \equiv_n\ :\ &(\forall d, d' \text{ s.t. } d \equiv_n d') \\ &f(e,d) \equiv_n f(e',d') \text{ and } f(d,e) \equiv_n f(d',e')\}\ . \end{aligned}$$

The problem here, however, is that the occurrence of $d \equiv_n d'$ is not positive. Accordingly, we make do with $d' = d$, reducing a k-ary function to its unary projections as follows.

Lemma 5. *Given a k-ary function $f : E^k \to E$ on E, an $i \in \{1, \ldots, k\}$ and a $(k-1)$-tuple $\overline{e} \in E^{k-1}$, define the function $f_{i,\overline{e}} : E \to E$ by*

$$f_{i,\overline{e}}(e) \quad = \quad f(\langle e, \overline{e}\rangle_i) \ ,$$

where $\langle e, \overline{e}\rangle_i$ is the k-tuple obtained from $\overline{e}$ by sticking e at its ith position. Then for every equivalence relation R on E,

$$\mathsf{Cng}[R, f] \quad = \quad \bigcap_{i \in \{1,\ldots,k\}} \ \bigcap_{\overline{e} \in E^{k-1}} \mathsf{Cng}[R, f_{i,\overline{e}}] \ .$$

Proof. The direction $\subseteq$ is immediate from the reflexivity of equivalence relations. For the converse, it suffices to observe that k applications of the rule

$$\frac{e \equiv e'}{f_{i,\overline{e}}(e) \equiv f_{i,\overline{e}}(e')} \qquad \text{for all } i \in \{1, \ldots, k\} \text{ and } \overline{e} \in E^{k-1}$$

and the transitivity of equivalence relations yield

$$\begin{array}{c}
\dfrac{\dfrac{e_1 \equiv e_1'}{f(e_1, \ldots, e_k) \equiv f(e_1', e_2, \ldots, e_k)} \quad e_2 \equiv e_2'}{f(e_1, \ldots, e_k) \equiv f(e_1', e_2', e_3, \ldots, e_k)} \quad e_3 \equiv e_3' \\
\vdots \qquad e_k \equiv e_k' \\
\hline
f(e_1, \ldots, e_k) \equiv f(e_1', \ldots, e_k')
\end{array}$$

or (more concisely)

$$\frac{e_1 \equiv e_1' \quad e_2 \equiv e_2' \quad \cdots \quad e_k \equiv e_k'}{f(e_1, e_2, \ldots, e_k) \equiv f(e_1', e_2', \ldots, e_k')} \ .$$

$\dashv$

We can now step up from $\mathsf{Cng}[R, f]$ to $\mathsf{Cng}[R, \mathcal{F}]$, where $\mathcal{F}$ is a family of multi-ary functions on E.

Theorem 6. *For every equivalence relation R on E and family $\mathcal{F}$ of multi-ary functions on E, $\mathsf{Cng}[R, \mathcal{F}]$ has a largest element.*

Proof. Let $\mathcal{F}^1$ be the family of unary projections of functions in $\mathcal{F}$

$$\begin{aligned} \mathcal{F}^1 \quad = \quad \{ f_{i,\overline{e}} \ : \ & f \text{ is a } k\text{-ary function in } \mathcal{F} \text{ and} \\ & i \in \{1, \ldots, k\} \text{ and } \overline{e} \in E^{k-1} \} \ . \end{aligned}$$

From Lemma 4, it follows that $\mathcal{F}^1$ has a largest element, namely

$$\bigcap_{f \in \mathcal{F}^1} \text{largest element of } \mathsf{Cng}[R, f] \ ,$$

which, by Lemma 5, is the largest element of $\mathcal{F}$. $\dashv$

Remarks.

(i) The intersection formed in the proof of Theorem 6 can be unwound as a co-induction that closes at ω, where

$$\begin{aligned} \equiv_0 &= R \\ \equiv_{n+1} &= \{(e,e') \in \equiv_n : (\forall f \in \mathcal{F}^1)\ f(e) \equiv_n f(e')\} . \end{aligned}$$

This more explicit argument is of interest inasmuch as closure at ω provides an upper bound on the complexity of a logic to reason about the largest equivalence in $\mathsf{Cng}[R,\mathcal{F}]$.

(ii) For $R = E \times E$, the co-induction stops at the earliest possible stage, the largest element of $\mathsf{Cng}[E \times E, \mathcal{F}]$ being just $E \times E$, for any $\mathcal{F}$. By contrast, line (18) bases the semantics of ambiguous expressions on some interpretation $[\![\cdot]\!] : \Phi \to V$ of a subset Φ of E (ensuring additionally that all elements of $\mathsf{Cng}[R,\mathcal{F}]$ are congruences relative to connectives on Φ that admit a compositional analysis under $[\![\cdot]\!]$). The next subsection focuses further on the case where $\mathcal{F}$ is the singleton $\{{}^\wedge{}^a\}$.

3.2. Merging under a non-well-founded set theory. Although an interpretation $[\![\cdot]\!]_a$ of E can always be derived from an equivalence $\equiv_a$ on E by (the term model) $[\![e]\!]_a = \{e' \in E : e' \equiv_a e\}$, it can be instructive to explore other interpretations that capture intuitions specific to the set E at hand. In particular, the notion that

(†) an ambiguous expression can be equated with the *set* of its *disambiguations*

is worked out in the present subsection, with the terms "set" and "disambiguation" construed in a slightly enriched way. Very briefly, the word "set" is taken literally, by adopting the theory described in Aczel [**1**], while "disambiguation" is interpreted as a process, given by iterated applications of ${}^\wedge{}^a$. The idea roughly is to equate two expressions unless differences between them can, by merging expressions, be observed according to the semantics $[\![\cdot]\!]$ of unambiguous formulas.

Proceeding more formally, define, for every collection X, its collection X^* of (finitary) sequences by

$$\begin{aligned} X^0 &= \{\emptyset\} \\ X^{n+1} &= \{\rho x \ : \ \rho \in X^n \text{ and } x \in X\} \\ X^* &= \bigcup_{n<\omega} X^n , \end{aligned}$$

and write X^+ for $X^* - X^0$. Next, given a binary function $f : (E \times E) \to E$ (e.g. ${}^\wedge{}^a$), define its extension $f^+ : E^+ \to E$ to E^+ by

$$\begin{aligned} f^+(e) &= e \\ f^+(\rho e) &= f(f^+(\rho), e) , \end{aligned}$$

and let f^{Φ} be the (graph of the) restriction of f^+ to elements of E^+ that outputs values in Φ

$$f^{\Phi} \quad = \quad \{(\rho, \varphi) \in E^+ \times \Phi \ : \ f^+(\rho) = \varphi\} \ .$$

To a first approximation, the idea is to equate $[\![e]\!]_a$ with the set $\{(\rho, [\![\varphi]\!]) : (e\rho, \varphi) \in f^{\Phi}\}$, where f is $^{\wedge_a}$. But to capture the largest congruence relative to f contained in the relation R specified by (18), we need also to "reduce" the sequences ρ in the pairs (ρ, φ) in f^{Φ}. Towards that end, it is useful to appeal to the theory of non-well-founded sets described in Aczel [**1**], from which the following notions are drawn. Given a collection $\mathcal{X}$ of *indeterminates*, a *system of equations over* $\mathcal{X}$ is a function s with domain $\mathcal{X}$ such that for every $x \in \mathcal{X}$, $s(x)$ is a set with atoms drawn from $\mathcal{X}$. Such a system s is often presented as $x = s(x)$, the reason being that a *solution to* that system s is a function ϕ with domain $\mathcal{X}$ such that for every $x \in \mathcal{X}$, the *xth root* $\phi(x)$ is equal to the set obtained from $s(x)$ by replacing every indeterminate x' occuring in $s(x)$ by $\phi(x')$. Returning now to the problem at hand, form the family $\mathcal{X}_E = \{x_e : e \in E\}$ of indeterminates (with $x_e \neq x_{e'}$ whenever $e \neq e'$), and define the system $s^{f,[\![\cdot]\!]}$ of equations

$$x_e \quad = \quad \{(x_{e_1} x_{e_2} \cdots x_{e_n}, [\![\varphi]\!]) \ : \ (e e_1 e_2 \cdots e_n, \varphi) \in f^{\Phi}\}$$

over $\mathcal{X}_E$. The *Anti-Foundation Axiom* (AFA) in Aczel [**1**] provides a unique solution to any system of equations, such as $s^{f,[\![\cdot]\!]}$. More abstractly, the roots of $s^{f,[\![\cdot]\!]}$ live in a certain *final* F*-coalgebra*, where F is the functor on the category of classes given by

$$F(X) \quad = \quad \mathsf{Pow}(X^* \times V)$$

(recalling that V is the image of $[\![\cdot]\!]$). Roughly speaking, the more familiar notion of an algebra is dualized to a coalgebra to keep the "final" structure from collapsing to a single point (just as the equivalence relation R constrains the largest element of $\mathsf{Cng}[R, f]$). To be more precise, observe

Proposition 7 (AFA). *Given a binary function* $f : (E \times E) \to E$, *a function* $[\![\cdot]\!] : \Phi \to V$, *where* $\Phi \subseteq E$, *and an* $e \in E$, *let* r_e *be the* x_e*th root of the system* $s^{f,[\![\cdot]\!]}$ *of equations (defined above). Then*

$$r_{f(e,e')} \quad = \quad \{(\rho, v) \ : \ (r_{e'} \rho, v) \in r_e\}$$

for all $e, e' \in E$.

Proof. The proposition follows from proving by a routine induction on n that

$$f^+(ee'e_1 \cdots e_n) = e_0 \quad \text{iff} \quad f^+(f(e,e')e_1 \cdots e_n) = e_0$$

for all $e, e', e_0, e_1, \ldots, e_n \in E$. $\dashv$

Theorem 8 (AFA). *Given a binary function* $f : (E \times E) \to E$ *and a function* $[\![\cdot]\!] : \Phi \to V$, *where* $\Phi \subseteq E$, *let* $\equiv$ *be the largest element of* $\mathsf{Cng}[R, f]$, *for* R *given by (18). Then, under the definitions mentioned in Proposition 7,*

$$e \equiv e' \quad \textit{iff} \quad r_e = r_{e'}$$

for all $e, e' \in E$.

Proof. Let $\sim$ be the equivalence relation on E induced by $s^{f,[\![\cdot]\!]}$

$$e \sim e' \quad \text{iff} \quad r_e = r_{e'} \ .$$

Clearly, $\sim$ is contained in R since for every $\varphi \in \Phi$, $(\varphi, \varphi) \in f^{\Phi}$. Furthermore, it is immediate from the previous proposition that $\sim$ is a congruence relative to f. It remains to show that $\equiv$ is contained in $\sim$. Suppose $e \not\sim e'$, and (without loss of generality) say, $(\rho, v) \in r_e - r_{e'}$. Then it is easy to see that $e \not\equiv_n e'$, where n is the length of the sequence ρ, and $\equiv_n$ is as defined in Remark (i) at the end of §3.1. $\dashv$

Remarks.

(i) It is crucial for the congruence property established in Proposition 7 that the domain of the function f^{Φ} (used to define the system $s^{f,[\![\cdot]\!]}$ of equations) include sequences of length greater than 1.

(ii) As the co-induction underlying the relative consistency proof for AFA does not, in general, close at ω, what is gained by the term construction in §3.1 is a tighter lower bound on the complexity of the equivalence induced by the interpretation.

(iii) L. Moss has pointed out to the author that there is a striking similarity between Theorem 8 and the Myhill-Nerode Theorem on minimum state deterministic finite automaton (see, e.g., Hopcroft and Ullman [**11**]). Putting aside the finiteness assumptions in the Myhill-Nerode Theorem, it would seem that the construction behind Theorem 8 amounts simply to a straightforward adaptation R_{Φ} of Nerode's maximal *right invariant* relation (with acceptance in a language replaced by evaluation to one of infinitely many formulas in Φ up to semantic equivalence)

$$\begin{aligned} eR_{\Phi}e' \quad \text{iff} \quad (\forall x \in E^*) \ \ & (f^+(ex) \in \Phi \text{ iff } f^+(e'x) \in \Phi) \\ & \text{and } (\forall \varphi \in \Phi)(f^+(ex) = \varphi \\ & \quad \text{implies } [\![f^+(e'x)]\!] = [\![\varphi]\!]) \ . \end{aligned}$$

The critical difference, however, is not just that f (or $^{\wedge_a}$) need not be concatenation (or even associative), but that Theorem 8 defines an interpretation based on simultaneously solving a system of equations, with the effect that the equivalence induced by this interpretation need not be the relation R_{Φ} defined above. That is to say, the asymmetry (reminiscent of right invariance) in the definition of the system of equations for Theorem 8 is only apparent: an indeterminate x_e in $s^{f,[\![\cdot]\!]}$ occurs not only as the left hand side of an equation, but also in the right hand side of many other equations! The more appropriate notion corresponding to the equivalence induced by the interpretation in Theorem 8 is not Nerode's right-invariance, but rather Myhill's two-sided equivalence (Hopcroft and Ullman [**11**], pages 65 and 74), suitably modified in §3.1 to cover (possibly non-associative) binary functions and more generally k-ary functions (for any finite k).

3.3. Simplifying the set of disambiguations. Reviewing the interpretation in §3.2 of an expression $e \in E$ relative to a merge operation ${}^{\wedge_a} : (E \times E) \to E$, the question arises as to whether we can trim down the set of disambiguations to which e is, under the intuition labeled (†) there, equated to some critical characteristic core (sufficient to distinguish e from other expressions). In this connection, two notions of disambiguation can be discerned. For every $e \in E$, define a *right disambiguation of* e to be a pair $(\rho, \varphi) \in E^* \times \Phi$ such that $f^+(e\rho) = \varphi$, where $f = {}^{\wedge_a}$. That is, a pair $(e_1 \cdots e_n, \varphi) \in E^* \times \Phi$ is a right disambiguation of e iff $(\cdots((e^{\wedge_a} e_1)^{\wedge_a} e_2) \cdots {}^{\wedge_a} e_n) = \varphi$. By contrast, a *left disambiguation of* e is a pair $(e_1 \cdots e_n, \varphi) \in E^* \times \Phi$ such that $(e_2 \cdots e_n e, \varphi)$ is a right disambiguation of e_1 — i.e., $(\cdots((e_1^{\wedge_a} e_2)^{\wedge_a} e_3) \cdots {}^{\wedge_a} e) = \varphi$. An equation

$$x_e \;=\; \{(x_{e_1} x_{e_2} \cdots x_{e_n}, [\![\varphi]\!]) \;:\; (e e_1 e_2 \cdots e_n, \varphi) \in f^{\Phi}\}$$

in the system $s^{f,[\![\cdot]\!]}$ defined in §3.2 encodes the right disambiguations of e; what is perhaps less obvious is that the left disambiguations are also encoded in $s^{f,[\![\cdot]\!]}$ through occurrences of indeterminates x_{e_n} to the right of an equation.[9] Turning to concrete examples from §1.2,

(19) ((U3), (U3)^(U1)) is a left disambiguation of (A1)

while

((A3),(U2)^`Josef eats the vegetable squash raw`)
is a right disambiguation of (A1) and
(20) a left disambiguation of (A3).

The notion of a right disambiguation is natural from the point of view of process theory (see, e.g., Aczel [**1**]), where a process is analyzed in terms of the ways it might evolve with time (i.e., stretching forward to the future). On the other hand, the notion of a left disambiguation comes closer, as pointed out by N. Asher, to intuitions from discourse theory, where an expression e is evaluated relative more often to the context established by the preceding discourse (i.e., looking back at the past) — examples such as (20) notwithstanding. This bias reflects the preference to link pronouns with discourse referents from the past (anaphora), rather than in the future (kataphora).

At any rate, note that in both (19) and (20), the resulting discourse (i.e., the second component of the pair) is obtained by applying the (unambiguous) merge ^ to unambiguous formulas. These examples suggest the possibility that

[9] This takes us back to Remark (iii) in §3.2, the force of which is that the same equivalence on expressions would be induced by solving a system of equations obtained from turning the right disambiguations in $s^{f,[\![\cdot]\!]}$ to left disambiguations. (Or for that matter, by solving a system of equations that explicitly represents both right and left disambiguations.) The apparent asymmetry vanishes through the magic of non-well-foundedness. To see what exactly is going on here, non-well-foundedness can be eliminated via §3.1. The key point is that the intersection formed in Lemma 5 must take into account all unary projections. In this respect, the approach of §3.2 is somewhat misleading, minimizing, as it does, the distinction between left and right disambiguations through the circularity in the equations.

(P0) whenever $e^\wedge{}_a e' \in \Phi$, there are $\varphi, \varphi' \in \Phi$ such that $e^\wedge{}_a e' = \varphi^\wedge\varphi'$.

Were (P0) true, we could define a function $\hat{\delta} : E \to \mathsf{Pow}(E)$ postulated by the schemes $\vdash_{QQ'}$ described in §2.2, satisfying

(P1) whenever $e^\wedge{}_a e' \in \Phi$, there is a $\varphi \in \hat{\delta}(e)$ and a $\varphi' \in \hat{\delta}(e')$ such that $e^\wedge{}_a e' = \varphi^\wedge\varphi'$.

Indeed, we could go further and connect e and e', if we replace $\hat{\delta}$ by two functions $\theta, \alpha : (E \times E) \to \mathsf{Pow}(\Phi)$ such that

(P2) whenever $e^\wedge{}_a e' \in \Phi$, there is a $\varphi \in \theta(e, e')$ and a $\varphi' \in \alpha(e, e')$ such that $e^\wedge{}_a e' = \varphi^\wedge\varphi'$

with the intuition that $\theta(e, e')$ is the set of unambiguous formulas to which the e can be reduced, given that e' occurs to its right, while $\alpha(e, e')$ is the set of unambiguous formulas to which the e' can be reduced, given that e occurs to its left.[10] Examples (19) and (20) suggest

$$\begin{array}{rcl} \text{(U1)} & \in & \alpha((\text{U3}), (\text{A1})) \\ \text{(U2)} & \in & \theta((\text{A1}), (\text{A3})) \\ \texttt{Josef eats the vegetable squash raw} & \in & \alpha((\text{A1}), (\text{A3})) \ . \end{array}$$

Our treatment of ambiguous expressions in §§3.1 and 3.2 can be labeled as *underspecified* in two ways; we neither assume

(i) some explicit representation (e.g., via $\bullet$) of an ambiguous expression e in terms of the set $\hat{\delta}(e)$ of unambiguous formulas to which e can be reduced

nor

(ii) the possibility (P0) — equivalently, the existence of functions $\hat{\delta}, \alpha$ and θ verifying (P1) and (P2).

The abstract notion of disambiguation §3.2 associates with (†) is very different from $\hat{\delta}$ or even θ and α. The sets $[\![e]\!]_a$ given by the x_eth-roots of $s^{f,[\![\cdot]\!]}$ are not just — like $\hat{\delta}(e)$ — subsets of Φ, but quite hairy beasts. For an arbitrary binary function f on E, it is difficult, if not impossible, to sort out the contributions that e, as opposed to some $e' \in E$, makes to $f(e, e')$, or to $[\![f(e, e')]\!]$, whenever $f(e, e')$ should happen to fall in some designated subset Φ of E (on which $[\![\cdot]\!]$ is defined).

[10] There is little point in mentioning $\hat{\delta}$ or α and θ if we know nothing more about them than (P1) or (P2). Some form of converse to (P1) or to (P2) would be useful to preclude, for instance, $\hat{\delta}(e) = \alpha(e, e') = \theta(e, e') = \Phi$ for all e and $e' \in E$. But having failed to formulate some such simple condition, I must appeal instead to the reader's command of English to make more refined choices of $\hat{\delta}$, α and θ. In particular, we should resist reducing say, α to θ by setting $\alpha(e, e') = \theta(e', e)$, in view of the difference between

(A8) `She seemed happy. A woman stopped to chat.`

and

(A9) `A woman stopped to chat. She seemed happy.`

The thrust of §3 is not, however, to deny the plausibility of (P0) for ambiguous expressions in natural language, only to investigate just how far we can get without appealing to (P0). It is a very different matter to rule out (P0); that would be a most radical form of underspecification. And it would be quite difficult to defend, if only because we would be hard-pressed, under such a view, to compute $^{\wedge_a}$ (never mind, some notion $\equiv_a$ of equivalence on E that presupposes $^{\wedge_a}$).

4. Discussion

Summarizing §§2 and 3, one might say the following. An expression is ambiguous precisely because, in isolation (i.e., without disambiguating information), its meaning is indeterminate, although it may, by merging with other expressions, produce a piece of discourse with a determinate meaning. The question is whether that meaning can be decomposed into discernible parts, isolating the contributions made by the pieces constituting the discourse. Put technically, a general co-inductive analysis of an ambiguous merge is presented in §3, but the problem of providing an inductive characterization of that merge is left open.

Is the summary above coherent? The matter might be clearer perhaps if we pay careful attention to the distinction between (on the one hand) syntax (i.e., the expressions in E and $^{\wedge_a}$) and (on the other hand) semantics (i.e., the meanings as specified, and connected to the syntactic material E, by $[\![\cdot]\!]$ and $[\![\cdot]\!]_a$). What may be puzzling is that §3 treats E and $^{\wedge_a}$ as black boxes, but at the same time presumes to analyze these "semantically." All the definitions highlighted in §2 (viz., Definitions 1, 2 and 3), plus the notions of left and right disambiguation from §3.3 are purely syntactic, and neither mention semantic notions nor enter into the semantic deliberations in §§3.1 and 3.2. The obvious question is what could semantics have to do with disambiguation?

It is *not* my intention to argue that there is more to disambiguation than manipulations on E. Indeed, I would not deny (nor contend) that all the computational action takes place inside what §3 takes to be black boxes (viz., E and $^{\wedge_a}$). That could well be true, but there is also the problem of understanding all that syntax! As expressions may get quite unwieldy, semantics could play a useful role by abstracting away inessential syntactic details,[11] imposing, as it were, some discipline on the chaos of syntax. Furthermore, to the extent that E and $^{\wedge_a}$ are theoretical constructs, there is some leeway in defining them; and here semantics may exert some influence (albeit perhaps behind the scenes). More concretely, recall Assumption A (from §2.3) and our decision to relax it, leading to the abstract investigations of §3. We return presently to the inductive generation of E, and an analysis of $^{\wedge_a}$ based, in no small measure, on semantic considerations centering around context.

[11] Under this picture, the coarser the semantic equivalence, the more work semantics has put to cleaning up after syntax — hence, the interest in §3 in the largest element of $\mathsf{Cng}[R, \mathcal{F}]$.

4.1. Ambiguous connectives for dynamic semantics. Recall that the intuition behind $^{\wedge_a}$ is that it is an extension of the merge $^\wedge$ on Φ to E that partially (if not completely) disambiguates arguments in $E - \Phi$. No mention of $^\wedge$ is, however, made in §3 before §3.3, and, indeed, precious little about $^\wedge$ is said anywhere above, other than it being some binary operation on Φ. We have been quite vague, for instance, about whether or not $^\wedge$ is (like $\vee$) one of the connectives γ of Φ mentioned in Definitions 1 and 3. In *Discourse Representation Theory* (DRT, Kamp and Reyle [**12**]), $^\wedge$ is *not* — even under a loose reading of "connective" allowing for the division between so-called discourse representations structures and conditions. It will be convenient in what follows, however, to construe $^\wedge$ as a binary connective not only of Φ but also of E; that is, let us assume that for all $e, e' \in E$, the string $e^\wedge e'$ is an element of E. This practice should not cause any confusion, because $e^\wedge e'$ will be assigned a meaning below faithful to an informal reading of the string as e juxtaposed (or concatenated) with e' (as is the case with $^\wedge$, qua binary operation on Φ). Now, building on postulate (P2) from §3.3, the idea is to set

$$e^{\wedge_a} e' \quad = \quad \theta_a(e, e') \ ^\wedge \ \alpha_a(e, e')$$

for all $e, e' \in E$, where θ_a and α_a are two binary functions on E similar (respectively) to θ and α (from §3.3), but with range equal to E (rather than to $\mathsf{Pow}(\Phi)$)

$$\begin{aligned} \theta_a(e, e') \quad &\approx \quad \text{a partial (if not total) disambiguation of } e \\ &\qquad\ \text{given that } e' \text{ occurs to its right} \\ \alpha_a(e, e') \quad &\approx \quad \text{a partial (if not total) disambiguation of } e' \\ &\qquad\ \text{given that } e \text{ occurs to its left .} \end{aligned}$$

This description will presently be made rigorous by a definition, the thrust of which is to unify, under a common system of connectives, our approaches to ambiguity via $^\bullet$ and via $^{\wedge_a}$. An example might ease digestion.

Just how pronouns link up with discourse referents (or variables) is left out of accounts such as Barwise [**3**] and Groenendijk and Stokhof [**9**], which concentrate instead on formalizing the introduction of a discourse referent z by a random assignment $z := ?$, leaving later linkages to that variable z to magic. Without attempting to spell out that "magic", let us record the appeal to it, by providing a semantics for the expressions $x = ?$ used in Kamp and Reyle [**12**] for pronouns that need to be linked with some such variable z. In this way, the constructs $\bullet$, $^\wedge$ and $x = ?$ can be viewed as forms (respectively) of disjunction, conjunction and existential quantification, introduced to account for ambiguity. Consider again the ambiguous statements (A1) and (A3) from §1.2, and assume that these can be formulated in isolation as (using more logic-like notation, and some simplifications sound for this particular example)

$$\begin{aligned} z := ? \ ^\wedge \ z' := ? \ ^\wedge \ \mathtt{Josef}(z) \ ^\wedge \ (\mathtt{squash}_v(z') \bullet \mathtt{squash}_s(z')) \\ ^\wedge \ \mathtt{like}(z, z') \end{aligned}$$

and

$$x = ? \wedge x' = ? \wedge \texttt{eat-raw}(x, x')$$

respectively. The two-sentence discourse formed from (A1) and then (A3) can be analyzed as

$$\begin{aligned}(z := ? \wedge z' := ? \wedge \texttt{Josef}(z) \wedge (\texttt{squash}_v(z') \bullet \texttt{squash}_s(z'))\\ \wedge \texttt{like}(z, z')) \wedge (x = ? \wedge x' = ? \wedge \texttt{eat-raw}(x, x'))\ ,\end{aligned}$$

which reduces to

$$\begin{aligned}(z := ? \wedge z' := ? \wedge \texttt{Josef}(z) \wedge \texttt{squash}_v(z') \wedge \texttt{like}(z, z')) \wedge\\ (x = z \wedge x' = z' \wedge \texttt{eat-raw}(x, x'))\ ,\end{aligned}$$

under a particular disambiguation strategy a that specifies a choice of $\texttt{squash}_v(z')$ over $\texttt{squash}_s(z')$, and a resolution of $x = ?$ and $x' = ?$ to $x = z$ and $x' = z'$, respectively.

Definitions 1 and 3 (from §2) suggest a decomposition of $e \wedge_a e'$ into $(e \wedge e')_a$, for some function $\cdot_a : E \to E$. But in order to accomodate contextual contributions to disambiguation, let us expand $\cdot_a$ into a function from $E \times E \times E$ to E such that for all $l, e, r \in E$,

$$\begin{aligned}(l, e, r)_a \quad \approx \quad & \text{the expression to which } a \text{ reduces } e \text{, given that}\\ & l \text{ occurs to } e\text{'s left, and } r \text{ occurs to its right}\end{aligned}$$

and, in particular,

$$\begin{aligned}\theta_a(e, r) &= (\top, e, r)_a\\ \alpha_a(l, e) &= (l, e, \top)_a\end{aligned}$$

for some (contentless) tautology $\top$ in Φ (such that $[\![\top \wedge \varphi]\!] = [\![\varphi \wedge \top]\!]$ for every $\varphi \in \Phi$).

Definition 4. *Given some set X of variables and some set Φ of formulas (over X)*[12] *closed under a set of connectives that includes $\wedge$ and $\neg$, let E be the set of expressions obtained by closing $\Phi \cup \{$'$x = ?$' $: x \in X\}$ under the connectives $\wedge$, $\neg$ and, in addition, a binary connective $\bullet$. Suppose also that a is a pair consisting of*

(i) *a function from triples $(x, l, r) \in X \times E \times E$ to expressions $\psi_a(x, l, r) \in E$*

and

(ii) *a function from pairs $(l, r) \in E \times E$ to asymmetric binary relations $\prec_a^{l,r}$ on E.*

[12] To capture first-order logic, it is useful (following Groenendijk and Stokhof [**9**]) to assume that for every $x \in X$, the random assignment $x := ?$ is an (atomic) element of Φ.

Define $(\cdot,\cdot,\cdot)_a : (E \times E \times E) \to E$ *by induction on the second argument as follows:*

$$
\begin{array}{rcl}
(l,\varphi,r)_a & = & \varphi \qquad \textit{for } \varphi \in \Phi \\
(l,x=?,r)_a & = & \psi_a(x,l,r) \\
(l,e \bullet e',r)_a & = & \left\{ \begin{array}{ll} (l,e,r)_a & \textit{if } e \prec_a^{l,r} e' \\ (l,e',r)_a & \textit{if } e' \prec_a^{l,r} e \\ (l,e,r)_a \bullet (l,e',r)_a & \textit{otherwise} \end{array} \right. \\
(l,e^\wedge e',r)_a & = & (l,e,e'^\wedge r)_a \;^\wedge\; (l^\wedge e,e',r)_a \\
(l,\neg e,r)_a & = & \neg(l,e,r)_a \;.
\end{array}
$$

Observe that Definition 4 does not so much extend a as it repackages it, as is obvious from the clause for $(l,e^\wedge e',r)_a$ (the righthand side of which simply shifts the complexity to the first and third arguments of $(\cdot,\cdot,\cdot)_a$). The point of the repackaging is to set the stage for a compositional analysis of ambiguous expressions (relative to a). A natural notion of semantic equivalence for the expressions specified in Definition 4 can be derived from an interpretation $[\![\cdot]\!] : \Phi \to V$ via Theorem 6 (from §3.1), with R as in (18), and $\mathcal{F}$ including $(\cdot,\cdot,\cdot)_a, \bullet, ^\wedge$ and the other connectives of Φ. The change in semantics induced by ambiguity is bound to be non-trivial; not even the associativity of (the semantic interpretation of) $^{\wedge_a}$ can be taken for granted. (Compare, from §1.2, (U3)$^{\wedge_a}$((A1)$^{\wedge_a}$(A3)) with ((U3)$^{\wedge_a}$(A1))$^{\wedge_a}$(A3); and see also Asher [**2**] and Fernando [**5**].)

4.2. Between monotonic interpretation and non-monotonic entailment. Beyond providing a compositional analysis of ambiguous expressions, Definition 4 suggests the following notions of monotonicity: a is *left-monotonic* if for all $l,e,r \in E$,

$$(l,e,r)_a \in \Phi \quad \text{implies} \quad (\forall e' \in E) \;\; (e'^\wedge l,e,r)_a = (l,e,r)_a$$

and *right-monotonic* if for all $l,e,r \in E$,

$$(l,e,r)_a \in \Phi \quad \text{implies} \quad (\forall e' \in E) \;\; (l,e,r^\wedge e')_a = (l,e,r)_a \;.$$

An argument that a proper choice of a should validate monotonicity might be based on construing the non-ambiguous character of formulas in Φ to mean that whenever $(l,e,r)_a$ *does* happen to be in Φ, it can*not* be re-interpreted, whatever further contextual information to its left or right there might be; otherwise, such an $(l,e,r)_a$ should have been set to an element of $E - \Phi$. I am not sure how compelling this reasoning is, as it may be turned into the (absurd?) requirement on a that $(l,e,r)_a \in \Phi$ only when $e \in \Phi$ (the presumption being that contexts can be expanded indefinitely to reverse an interpretation).

If it is not too implausible to assume that a is monotonic (by choosing a value in $E - \Phi$ whenever there is any uncertainty in how to disambiguate an expression), it ought not be too surprising that total disambiguations (which assume values in Φ, even when choices can change, depending on further contextual information) should have non-monotonic consequences. Indeed, a wide variety of non-monotonic consequence relations (in the sense, for example, of Kraus, Lehmann and Magidor [**13**]) can be represented within Definition 2 (from §2.2), based on disambiguations δ with range equal to Φ. More precisely, so-called preferential consequence relations $\mid\!\sim$ on Φ can be reconstructed by taking

$$\varphi \mid\!\sim \varphi' \quad \text{iff} \quad \varphi{}^\wedge c \vdash^{\mathcal{D}} \varphi'$$

for all $\varphi, \varphi' \in \Phi$, where c is some "rogue" expression in $E - \Phi$, expressing contextual indeterminacy, and subject to a certain choice of $\mathcal{D}$ for Definition 4 (Fernando [**7**]). In view of the sensitivity of disambiguation to expressions to the right as well as to the left, it is worth pointing out perhaps that the consequent e' in the judgment $e \vdash^{\mathcal{D}} e'$ does not have the same status in disambiguating e as it would in $e{}^{\wedge_a} e'$, for the truth of e' ought not to be assumed in deciding whether or not $e \vdash^{\mathcal{D}} e'$. Nonetheless, Definition 4 does suggest a notion of order for monotonicity. Call a family $\mathcal{D}$ of subsets of some set $\Delta \subseteq (\Phi \times E) \rightarrow \Phi$ *left-monotonic* if whenever $e \vdash^{\mathcal{D}} e'$, $l{}^\wedge e \vdash^{\mathcal{D}} e'$ for every $l \in E$. Similarly, $\mathcal{D}$ is *right-monotonic* if whenever $e \vdash^{\mathcal{D}} e'$, $e{}^\wedge r \vdash^{\mathcal{D}} e'$ for every $r \in E$. Relating the partiality of $(\cdot,\cdot,\cdot)_a$, the totality in $\mathcal{D}$, monotonic interpretation and non-monotonic entailment is just one of the many topics that the present paper must leave for further investigation (concerning which a perspective from programming language semantics is offered in the companion paper Fernando [**8**]).

Acknowledgments. My thanks to N. Asher, H, Kamp, L. Moss, S. Peters and U. Reyle for helpful discussions, and to an anonymous referee for useful suggestions.

References

[1] Peter Aczel. *Non-Well-Founded Sets.* CSLI Lecture Notes Number 14, Stanford, 1988.

[2] Nicholas Asher. *Reference to Abstract Objects in Discourse.* Kluwer Academic Publishers, Dordrecht, 1993.

[3] Jon Barwise. Noun phrases, generalized quantifiers and anaphora. In E. Engdahl and P. Gärdenfors, editors, *Generalized Quantifiers*, Studies in Language and Philosophy. Dordrecht: Reidel, 1987.

[4] Kees van Deemter. Towards a logic of ambiguous expressions. In K. van Deemter and S. Peters, editors, *Semantic Ambiguity and Underspecification.* CSLI Lecture Notes Number 55, Stanford, 1996.

[5] Tim Fernando. Are context change potentials functions? Manuscript, distributed at a workshop on context dependency in Prague, 1995.

[6] Tim Fernando. A logical connective for ambiguity requiring disambiguation. In J. Groenendijk, editor, *Ellipsis, Underspecification, Events and More in Dynamic Semantics.* Dyana deliverable R2.2.B, 1995.

[7] Tim Fernando. Non-monotonic consequences of preferential contextual disambiguation. In *Proc. Tenth Amsterdam Colloquium.* ILLC, University of Amsterdam, 1996.

[8] Tim Fernando. Analyzing ambiguity denotationally, operationally and logically. Manuscript, 1996.

[9] J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14, 1991.

[10] Irene Heim. The semantics of definite and indefinite noun phrases. Dissertation, University of Massachusetts, Amherst, 1982.

[11] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Language and Computation*. Addison-Wesley, 1979.

[12] H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht, 1993.

[13] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44, 1990.

[14] Massimo Poesio. Semantic ambiguity and perceived ambiguity. In K. van Deemter and S. Peters, editors, *Semantic Ambiguity and Underspecification*. CSLI Lecture Notes Number 55, Stanford, 1996.

[15] Uwe Reyle. Dealing with ambiguities by underspecification: construction, representation and deduction. *Journal of Semantics*, 10(2), 1993.

[16] Uwe Reyle. On reasoning with ambiguity. In *Proc. of the 7th Conference of the European Chapter of the Association for Computational Linguistics*. Dublin, 1995.

[17] Wlodek Zadrozny. From compositional to systematic semantics. *Linguistics and Philosophy*, 17(4), 1994.

7

Non-monotonic consequences of preferential contextual disambiguation

ABSTRACT. Logical consequence between (unambiguous) formulas is extended to ambiguous expressions through disambiguations mapping the latter to the former. A particular system of *preferential contextual* disambiguation is related to well-known rules for non-monotonic consequence relations $\mathrel{|\!\sim}$. On the one hand, the system provides a representation of $\mathrel{|\!\sim}$ based on an ambiguous expression of the background conditions underlying the entailment. On the other hand, rules for $\mathrel{|\!\sim}$ suggest requirements on disambiguations and on preferences between them.

1. INTRODUCTION

A familiar obstacle to applying logic to analyze the meaning of a natural language utterance is that pragmatics is commonly abstracted out of logic. Frustration with logic short on pragmatics might well drive one to the bold alternative slogan

logic is a consequence of pragmatics!

But what could the words "logic" and "consequence" (not to mention "pragmatics") mean? Notions of logical *consequence* going back to Tarski and Gentzen have been relaxed in more recent years to provide a minimal criteria for "logic," admitting, in particular, the possibility of non-monotonicity. A concrete test then of the slogan above is provided by the matter of ambiguity, construed broadly and pragmatically.

1.1. Summary. The present paper relates the theory of non-monotonic consequence relations $\mathrel{|\!\sim}$ (e.g., Gabbay [**6**]) to "preferential, contextual disambiguation." An indeterminate background context is ascribed to $\mathrel{|\!\sim}$, and formulated as an "ambiguous" expression c that is then "merged" with the antecedent (to the left of $\mathrel{|\!\sim}$), yielding an ambiguous expression. Ambiguous expressions are "disambiguated" into unambiguous formulas by functions, called *disambiguations*, that take context into account (hence, the term "contextual disambiguation"). A "preference" relation $\prec$ on (the set of) disambiguations is assumed that induces an extension $\vdash_{\prec}$ of a classical consequence relation $\vdash$ (on unambiguous formulas) to possibly ambiguous expressions. The consequence relation $\mathrel{|\!\sim}$ between expressions e and e' is then characterized by the equivalence

$$e \mathrel{|\!\sim} e' \quad \text{iff} \quad c^{\wedge}e \vdash_{\prec} e' \ ,$$

for some binary (merge) operation $^{\wedge}$ on expressions, generalizing classical conjunction on unambiguous formulas.

1.2. Aims. Properties postulated for non-monotonic consequence relations (e.g., the cumulative and preferential systems described in Kraus, Lehmann and Magidor [**9**]) translate into requirements on the set Δ of disambiguations, and on the preference relation $\prec \subseteq \Delta \times \Delta$ on Δ. Conversely, representation theorems based on Δ and $\prec$ constitute an alternative to the usual (forms of) preferential semantics (going back to Shoham [**14**], Makinson [**10**], etc.). Extensions of $\mid\!\sim$ to ambiguous expressions, including (nested) conditionals, are also provided by $\vdash_{\prec}$.

1.3. Underlying intuitions. The intuition behind preferential semantics is that agents cope with uncertainty about information by preferring certain interpretations over others. The notion of context can be understood either as an object that supports some information or as a piece of information itself. In either case, the partiality of contexts suggests a formalization of contexts as formulas[1] (or as sets of models), with indeterminate contexts analyzed as ambiguous expressions. What is novel about the present work is that the preference relation $\prec$ above is defined *not* on contexts, but on functions that disambiguate contexts; that is, $\prec$ is a binary relation on the forementioned disambiguations. The idea in linking $\mid\!\sim$ to a notion $\vdash_{\prec}$ of logical consequence between possibly ambiguous expressions is to trace deviant features of commonsense reasoning to the step from classical logic (with an unambiguous consequence relation $\vdash$) to natural language (interpreted relative to a context that admits no unambiguous formulation). The syntactic nature of this characterization of $\mid\!\sim$ is deliberate; the thrust of representation theorems (in terms of Δ and $\prec$) is not so much to describe model-theoretic systems (derived by replacing formulas with sets of models) relative to which certain proof rules are complete, but rather to explore the scope of a "*representation*alist" conception of disambiguation that need not appeal to models.

2. Background

A natural operation to perform on two formulas φ and φ' is to "merge" them into a formula $\varphi^\wedge\varphi'$ that contains the information in φ and in φ'. This operation supports a view of formulas as both contexts φ and "context change potentials" φ' insofar as

$$\varphi^\wedge\varphi' \quad \approx \quad \text{the context returned by asserting } \varphi' \text{ at context } \varphi\ .$$

Classical conjunction is an obvious example of such an operation, but, in fact, the term "context change potential" is used in certain formal approaches to natural language discourse (e.g., Kamp and Reyle [**8**]), where $^\wedge$ is not commutative (hence the elevation of the symbol $\wedge$, suggesting concatentation). These approaches are motivated in no small measure by the problem of *anaphora* in natural language, which, very roughly, has to do with the question of what pronouns refer to, as in the piece of discourse

[1] This point is taken up at length in §4.1 below.

```
Ernest has a brother.
He would understand this paper.
```

As illustrated above, a pronoun (occurrence) may well have more than one possible reference, leading to *ambiguity*.[2] With this in mind, let us distinguish the set Φ of unambiguous formulas from a larger set $E \supset \Phi$ of (possibly ambiguous) expressions (regarding say, a formula with an unbound variable as an element of $E - \Phi$). Under this division, a classical notion of logical consequence $\vdash$ can be assumed for Φ that treats (the restriction to Φ of) $\hat{}$ as just ordinary conjunction $\wedge$. This point is discussed in §4.2 below. For now, let us take it for granted, and turn our attention to how to extend $\vdash$ to E.

2.1. Eliminating ambiguity. Inasmuch as ambiguous expressions differ from unambiguous formulas in that they require "disambiguation," it is natural to bring such a notion into the picture. Toward that end, let Δ be a set of *disambiguations* $\delta : (\Phi \times E) \to \Phi$, with the intuition that

$$\delta(\varphi, e) \quad \approx \quad \text{a disambiguation of } e \text{ given context } \varphi \ .$$

Observe that an ambiguous expression e is totally disambiguated (relative to an unambiguous context), but that "partial" disambiguation might be modeled as a set of "total" disambiguations. Also, Δ cannot, of course, be the full function space $(\Phi \times E) \to \Phi$. For instance, every $\delta \in \Delta$ must leave unambiguous formulas as is; that is, $\delta(\varphi, \varphi') = \varphi'$, for all φ and $\varphi' \in \Phi$. Other requirements should doubtless be imposed, and indeed one of the very points of this paper is to uncover just what those might be. Let us move along then, and notice that a disambiguation $\delta : (\Phi \times E) \to \Phi$ induces a natural extension of $\vdash \subseteq \Phi \times \Phi$ to $\vdash^{\delta} \subseteq E \times E$ relative to an (implicit) "empty context" $\top \in \Phi$ (such that, for example, $\varphi \vdash \top$ for every $\varphi \in \Phi$) as follows

$$e \vdash^{\delta} e' \quad \text{iff} \quad \delta(\top, e) \vdash \delta(\delta(\top, e), e') \ .$$

Note that the disambiguation of the succedent e' is carried out relative to the context $\delta(\top, e)$ created by the antecedent (asserted at the context $\top$).

Next, let us quantify out the choice of δ by asking

$$\textit{is the set } \{\delta \in \Delta : e \vdash^{\delta} e'\} \textit{ large?}$$

Without saying exactly what "large" means, let us assume some answer given by a family $\mathcal{D} \subseteq \mathsf{Pow}(\Delta)$ of sets of disambiguations (so that a subset D of Δ is "large" iff $D \in \mathcal{D}$). Then, it is natural to extend $\vdash \subseteq \Phi \times \Phi$ to $\vdash^{\mathcal{D}} \subseteq E \times E$ according to

$$e \vdash^{\mathcal{D}} e' \quad \text{iff} \quad \{\delta \in \Delta \ : \ e \vdash^{\delta} e'\} \in \mathcal{D} \ .$$

Now, returning to $\mathcal{D}$, we may assume the full set Δ is large — i.e., $\Delta \in \mathcal{D}$ —, and that all sets larger than a large set are large — i.e.,

$$\frac{D \in \mathcal{D} \qquad D \subseteq D'}{D' \in \mathcal{D}} \ .$$

[2]As it happens, I have a brother named Ernest. Alas, I doubt he'd understand this paper.

Assuming further that $\mathcal{D}$ is closed under intersections,

$$\frac{D \in \mathcal{D} \qquad D' \in \mathcal{D}}{D \cap D' \in \mathcal{D}} \, ,$$

turns $\mathcal{D}$ into a *filter*. This third assumption will prove useful for technical reasons, but is certainly more questionable than the other two.

2.2. Non-monotonic consequences of preferential semantics. The "technical reasons" alluded to above concern the fitness of $\vdash^{\mathcal{D}}$ to be a notion of logical consequence. The criteria for such a notion have changed over the years, reflecting the interest in non-monotonicity. A number of rules (or more precisely, closure conditions) for a binary relation $\mid\!\sim \subseteq L \times L$ on a set L of formulas are presented in Gabbay [**6**], based on a classical consequence relation $\vdash \subseteq L \times L$. Typically, $\mid\!\sim$ is both stronger and weaker than $\vdash$, in that $\vdash \subseteq \mid\!\sim$, but $\mid\!\sim$ respects less rules. This is the case for what Kraus, Lehmann and Magidor [**9**] describe as Gabbay's base system **C** of *cumulative reasoning*, which consists of the rules of *reflexivity*

$$\text{(Ref)} \qquad \frac{}{e \mid\!\sim e} \, ,$$

left logical equivalence

$$\text{(LLE)} \qquad \frac{e_1 \mid\!\sim e \qquad e_1 \vdash e_2 \qquad e_2 \vdash e_1}{e_2 \mid\!\sim e} \, ,$$

right weakening

$$\text{(RW)} \qquad \frac{e \mid\!\sim e_1 \qquad e_1 \vdash e_2}{e \mid\!\sim e_2} \, ,$$

cut

$$\text{(Cut)} \qquad \frac{e_1 \mid\!\sim e_2 \qquad e_1{}^\wedge e_2 \mid\!\sim e}{e_1 \mid\!\sim e} \, ,$$

and *cautious monotonicity*

$$\text{(CM)} \qquad \frac{e_1 \mid\!\sim e_2 \qquad e_1 \mid\!\sim e}{e_1{}^\wedge e_2 \mid\!\sim e} \, .$$

Observe that all these rules hold for $\mid\!\sim$ equal to $\vdash$, and that (more generally) the containment $\vdash \subseteq \mid\!\sim$ is an immediate consequence of (Ref) and (RW). As it turns out, the rule (CM) has proved to be somewhat controversial, in no small measure because it fails in Reiter [**12**]'s *default logic* (as pointed out in Makinson [**10**]). A weaker system than **C** is proposed in Voorbraak [**16**] to capture default logic. Voorbraak [**16**] follows Makinson [**10**] in considering infinitary (Tarski-syle) consequence relations. Following instead Kraus, Lehmann and Magidor [**9**] in studying finitary (Gentzen-style) consequence relations, let us weaken **C** to $\mathbf{C}^-$ by replacing (CM) with

$$\text{(And)} \qquad \frac{e \mid\!\sim e_1 \qquad e \mid\!\sim e_2}{e \mid\!\sim e_1{}^\wedge e_2} \, .$$

The rule (And) is valid for default logic, and can be derived in **C**.

A systematic approach to evaluating the soundness of rules for $\mid\sim$ is, of course, to provide a general semantics for $\mid\sim$. Proposals based on a preferential relation on contexts go back (at least) to Shoham [**14**], where contexts are taken to be models, and to Makinson [**10**], where that identification is loosened. More recent developments are described in Kraus, Lehmann and Magidor [**9**] and Voorbraak [**16**]. The present work can be viewed as yet another variation on the theme, the essential novelty being that a distinction between contexts and interpretations is drawn (motivated by the problem of ambiguity), and that it is on the latter that preferences are defined.

2.3. Disambiguating entailment, abduction and deduction. Let us now merge the two previous subsections by analyzing an entailment $e \mid\sim e'$ in terms of an ambiguous expression $c \in E - \Phi$ that makes the entailment's background conditions explicit. That is, let us stipulate

$$(\dagger) \qquad e \mid\sim e' \ \text{ iff } \ c^\wedge e \vdash^{\mathcal{D}} e' \ .$$

Note that for $e, e' \in \Phi$, it suffices (as far as $(\dagger)$ is concerned) to consider the restriction of disambiguations δ to $(\Phi \times \Phi) \cup (\{\top\} \times \{c^\wedge\varphi : \varphi \in \Phi\})$. On $\Phi \times \Phi$, we have already decided that δ should respect the unambiguous character of formulas in Φ — i.e.,

$$\text{(C1)} \qquad \delta(\varphi, \varphi') \ = \ \varphi' \ , \quad \text{for all } \varphi, \varphi' \in \Phi \ .$$

As for $\{\top\} \times \{c^\wedge\varphi : \varphi \in \Phi\}$, let us observe that contexts not only disambiguate, but can themselves be disambiguated — i.e.,

$$\text{(C2)} \qquad \delta(\top, c^\wedge\varphi) \ = \ c^\delta_\varphi \wedge \varphi \ , \quad \text{for every } \varphi \in \Phi \ ,$$

for certain choices $c^\delta_\varphi \in \Phi$.

Next, let us derive the notion $\mathcal{D}$ of a "large" set from some preference relation $\prec \subseteq \Delta \times \Delta$ on Δ. An obvious candidate is that of a $\prec$-dense set, where a subset $D \subseteq \Delta$ is defined to be *$\prec$-dense* if for every $\delta \in \Delta$, there is a $\delta' \in D$ such that $\delta' \preceq \delta$ (writing $\preceq$ for $\prec \cup =$). Unfortunately, the family of $\prec$-dense sets is not necessarily closed under intersection.[3] This defect is easily corrected by requiring that a large set D be *stably $\prec$-dense* in that for every $\delta \in \Delta$, there is a $\delta' \preceq \delta$ such that for every $\delta'' \preceq \delta'$, $\delta'' \in D$.

Proposition 1 (Folklore). *For any transitive relation $\prec$, the family $\mathcal{D}_\prec$ of stably $\prec$-dense sets is a filter.*

Let us pause for a moment to reflect on where things stand. A characterization of $e \mid\sim e'$ is proposed, in which ambiguity is injected through a "rogue" context $c \in E - \Phi$ that validates the entailment, under a disambiguation from the set $\{\delta \in \Delta : \ c^\wedge e \vdash^\delta e'\}$. Although preferences have been defined here not (as in the usual case) on contexts, but rather on disambiguations, note that the restriction of the equivalence $(\dagger)$ above to unambiguous formulas $e, e' \in \Phi$

[3] See the pair $(\Delta, \prec)$ constructed for the representation theorem in §3.2. Alternatively, closure under intersections might be required only for "definable" sets in $\mathcal{D}$.

reduces (for all purposes) a disambiguation δ to its value $\delta(\top, c^\wedge e)$ at the antecedent. That is, for $\varphi, \varphi' \in \Phi$, the set

$$C_{\varphi,\varphi'} \;=\; \{c^\delta_\varphi \;:\; \delta \in \Delta \text{ and } c^\wedge\varphi \vdash^\delta \varphi'\}$$

can be regarded as the set of possible "explanations" for the entailment $\varphi \mathrel{|\!\sim} \varphi'$. The reader familiar with Peirce [**11**] and the more recent paper, Hobbs et al [**7**], might at this point bring in *abduction*. If explanations are equated with partial disambiguations of c, modeled by sets of total disambiguations of c, then the preference relation for abduction might simply be the superset relation, with the set $C_{\varphi,\varphi'}$ as *the* explanation abduced for the entailment $\varphi \mathrel{|\!\sim} \varphi'$. If, moreover, $C_{\varphi,\varphi'}$ (or more precisely, the corresponding set $\{\delta \in \Delta : c^\delta_\varphi \in C_{\varphi,\varphi'}\}$) is stably $\prec$-dense, then abduction *is* deduction! Alternatively, the preference relation for abduction can also be equated with the preference relation $\prec$ on Δ, in which case the explanation abduced for the entailment $\varphi \mathrel{|\!\sim} \varphi'$ must be drawn from the set $\{\delta \in \Delta : c^\delta_\varphi \in C_{\varphi,\varphi'}\}$. Just what is entailed by the assumption that optimal explanations exist will be taken up in §4.3.

3. Cumulative consequence relations interpreted ambiguously

The rules for $\mathrel{|\!\sim} \subseteq L \times L$ in Kraus, Lehmann and Magidor [**9**] mention a consequence relation $\vdash \subseteq L \times L$ that is classical on L. For this reason, let us identify L with Φ, and restrict $\mathrel{|\!\sim}$ to Φ, leaving to §3.3 a brief consideration of the possibility of extending the theory to E. Recall, for the record, that the system $\mathbf{C}$ of cumulative reasoning is given by

$$\mathbf{C} \;=\; \{(\text{Ref}), (\text{LLE}), (\text{RW}), (\text{Cut}), (\text{CM})\}\ ,$$

which has a weakened form

$$\mathbf{C}^- \;=\; \{(\text{Ref}), (\text{LLE}), (\text{RW}), (\text{Cut}), (\text{And})\}\ .$$

Let us also write $e \equiv e'$ to abbreviate the conjunction, $e \vdash e'$ and $e' \vdash e$, and observe that $\equiv$ is an equivalence on Φ, but not on E.

3.1. Soundness. The following choices of sufficient conditions are not so ad hoc, in view of the representation theorem from the next section. A pair $(\Delta, \prec)$ is $\mathbf{C}^-$-*sufficient* if

(i) the family $\mathcal{D}_\prec$ of stably $\prec$-dense sets is a filter (e.g., $\prec$ is transitive),

(ii) for all $\delta \in \Delta$, and $\varphi, \varphi' \in \Phi$, and $e \in E$,

$$\text{(C3)} \qquad \frac{\varphi \equiv \varphi'}{\delta(\varphi, e) \vdash \delta(\varphi', e)}$$

and

(iii) for all $\varphi, \varphi' \in \Phi$,

$$\frac{\{\delta \in \Delta \;:\; c^\wedge\varphi \vdash^\delta \varphi'\} \in \mathcal{D}_\prec \;\; (\text{i.e., } \varphi \mathrel{|\!\sim} \varphi')}{\{\delta \in \Delta \;:\; \delta(\top, c^\wedge\varphi) \vdash \delta(\top, c^\wedge(\varphi \wedge \varphi'))\} \;\in\; \mathcal{D}_\prec}\ .$$

Weakening the premiss in (C3) to $\varphi \vdash \varphi'$, would strengthen (LLE) as in (RW), leading to monotonicity. Condition (iii) is unreletingly ugly, but can be made prettier if we pass from $\mathbf{C}^-$ to a preferential version of it, described in §3.3. In the meantime, call $(\Delta, \prec)$ $\mathbf{C}$-*sufficient* if it is $\mathbf{C}^-$-sufficient, and (iii) is strengthened into

$$\frac{\{\delta \in \Delta \ : \ c^\wedge\varphi \vdash^\delta \varphi'\} \in \mathcal{D}_\prec}{\{\delta \in \Delta \ : \ \delta(\top, c^\wedge(\varphi \wedge \varphi')) \equiv \delta(\top, c^\wedge\varphi)\} \ \in \ \mathcal{D}_\prec}$$

for all $\varphi, \varphi' \in \Phi$.

Proposition 2 (Soundness). *If* $(\Delta, \prec)$ *is* $\mathbf{C}^-$*-sufficient, then the relation* $\{(\varphi, \varphi') \in \Phi \times \Phi : c^\wedge\varphi \vdash_\prec \varphi'\}$ *verifies* $\mathbf{C}^-$. *Similarly, for the full system* $\mathbf{C}$.

Proof. Given the conditions (C1) and (C2) from §2.3, use

(a) $\Delta \in \mathcal{D}_\prec$ for (Ref),
(b) $\mathcal{D}_\prec$ is upward closed for (RW),
(c) $\mathcal{D}_\prec$ is closed under intersections for (And),
(d) condition (C3) for (LLE),

and

(e) part (iii), as well as the closure properties of $\mathcal{D}_\prec$, for (Cut).

(CM) follows as in part (e) from the additional condition for $\mathbf{C}$-sufficiency. $\dashv$

3.2. Completeness [sic]. Fix a relation $\mid\!\sim \subseteq \Phi \times \Phi$ validating say, $\mathbf{C}^-$, with the idea of defining a suitable pair $(\Delta, \prec)$, from which $\mid\!\sim$ can be reconstructed. A first attempt might be to disambiguate c uniformly as $\{\psi : \top \mid\!\sim \psi\}$. However, even if we set aside the problem that $\{\psi : \top \mid\!\sim \psi\}$ is infinite, there remains the question of establishing the equivalence

$$(\ddagger) \qquad \varphi \mid\!\sim \varphi' \quad \text{iff} \quad \{\psi : \top \mid\!\sim \psi\}, \varphi \vdash \varphi' \ .$$

(As usual, $\vdash$ can be extended to $\mathsf{Pow}(\Phi) \times \Phi$ by

$$\Gamma \vdash \varphi \quad \text{iff} \quad (\exists \text{ finite } \Gamma_0 \subseteq \Gamma) \bigwedge \Gamma_0 \vdash \varphi \ ,$$

with $\Gamma, \varphi \vdash \varphi'$ written instead of $\Gamma \cup \{\varphi\} \vdash \varphi'$.) It suffices for the forward direction $\Rightarrow$ of $(\ddagger)$ that

$$\varphi \mid\!\sim \varphi' \quad \text{implies} \quad \top \mid\!\sim \varphi \supset \varphi' \ ,$$

which is a special case of the "hard half of the deduction theorem" called rule (S) in Kraus, Lehmann and Magidor [**9**] (page 191). Avoiding this rule altogether, let us disambiguate c instead as $\{\psi \supset \psi' : \psi \mid\!\sim \psi'\}$. But there is still the backward direction $\Leftarrow$ of $(\ddagger)$, for which the relativization c^δ_φ of the disambiguation of c to φ, given by (C2), is useful. In particular, let us trim down the disambiguation further by setting our sights on the theory

$$\Gamma_\varphi \ = \ \{\varphi \supset \varphi' \ : \ \varphi \mid\!\sim \varphi'\} \ ,$$

which enjoys the following property.

Lemma 3.[4] *Assuming* $\mid\!\sim$ *verifies* $\mathbf{C}^-$,

$$\varphi \mid\!\sim \varphi' \quad \textit{iff} \quad \Gamma_\varphi, \varphi \vdash \varphi'$$

for all $\varphi, \varphi' \in \Phi$.

Proof. Suppose $\varphi \mid\!\sim \varphi'$. Then by the definition of Γ_φ, $\Gamma_\varphi \vdash \varphi \supset \varphi'$, and we are done by the deduction theorem for $\vdash$. The converse requires a bit more work. Assume $\Gamma_\varphi, \varphi \vdash \varphi'$. That is, there are $\varphi_1, \ldots, \varphi_n$ such that

$$(*) \qquad \varphi \ \mid\!\sim\ \varphi_i \quad \text{for every } i \in \{1, \ldots, n\} \ ,$$

and

$$(**) \qquad \bigwedge_{i=1}^{n} (\varphi \supset \varphi_i) \wedge \varphi \ \vdash \ \varphi' \ .$$

Now, $(**)$ implies

$$(\bigwedge_{i=1}^{n} \varphi_i) \wedge \varphi \ \vdash \ \varphi' \ ,$$

whence (since $\vdash$ is contained in $\mid\!\sim$)

$$(\bigwedge_{i=1}^{n} \varphi_i) \wedge \varphi \ \mid\!\sim \ \varphi' \ .$$

Also, by the rule (And), it follows from $(*)$ that

$$\varphi \ \mid\!\sim \ \bigwedge_{i=1}^{n} \varphi_i \ .$$

Thus, (Cut) yields $\varphi \mid\!\sim \varphi'$, as required. $\dashv$

Next, let us attend to the problem that the set Γ_φ is infinite. This is where Δ and $\prec$ come to play. Assuming Φ is countable, fix an enumeration $\{\psi_0, \psi_1, \ldots\}$ of Φ, and build a set $\Delta = \{\delta_0, \delta_1, \ldots\}$ of disambiguations, where (for every $\varphi \in \Phi$) δ_i approximates Γ_φ up to $\{\psi_0, \ldots, \psi_i\}$ in the sense that

$$c_\varphi^{\delta_i} \ = \ \bigwedge\{\varphi \supset \varphi' \ : \ \varphi \mid\!\sim \varphi' \text{ and } \varphi' \in \{\psi_0, \psi_1, \ldots, \psi_i\}\} \ .$$

Arrange

$$\delta_i(\varphi, \varphi') \ = \ \varphi'$$

for every i and for all $\varphi, \varphi' \in \Phi$. Then define $\prec$ by

$$\delta_i \prec \delta_j \quad \text{iff} \quad i > j \ .$$

Let $\vdash_\prec$ be $\vdash^{\mathcal{D}_\prec}$, where $\mathcal{D}_\prec$ is the family of stably $\prec$-dense sets.

---

[4] This is the key lemma, corresponding to Lemma 3.18 in Kraus, Lehmann and Magidor [**9**]. It seems that (LLE) is unnecessary, unless, as suggested in §3.3, (Cut) is replaced by the preferential rule (Or) described there.

Theorem 4 (Representation). *Let Φ be countable, and $\mid\!\sim\ \subseteq \Phi \times \Phi$ verify* $\mathbf{C}^-$. *Then for Δ and $\prec$ constructed as above,*

$$\varphi \mid\!\sim \varphi' \quad \textit{iff} \quad c^\wedge\varphi \vdash_\prec \varphi'$$

for all $\varphi, \varphi' \in \Phi$. Furthermore, $(\Delta, \prec)$ is $\mathbf{C}^-$-sufficient. Similarly, for $\mathbf{C}$.

Proof. For the equivalence, it suffices by Lemma 3 to establish that

$$\Gamma_\varphi, \varphi \vdash \varphi' \quad \text{iff} \quad c^\wedge\varphi \vdash_\prec \varphi' \ ,$$

which is easy because $\prec$ is a linear order, and whenever $\delta \prec \delta'$, $c^\delta_\varphi \vdash c^{\delta'}_\varphi$.

Turning to $\mathbf{C}/\mathbf{C}^-$-sufficiency, use Proposition 1 and the definition of c^δ_φ for parts (i) and (ii)[5] respectively of $\mathbf{C}^-$-sufficiency. Part (iii) and the additional condition for $\mathbf{C}$-sufficiency are proved by reversing the argument for soundness. In particular, observe that

$$\frac{\{\delta \in \Delta \ : \ c^\delta_\varphi \wedge \varphi \vdash \varphi'\} \in \mathcal{D}_\prec}{\{\delta \in \Delta \ : \ c^\delta_\varphi \wedge \varphi \vdash c^\delta_{\varphi\wedge\varphi'}\} \in \mathcal{D}_\prec}$$

follows from (Ref), (Cut), (And) and the definition of c^δ_φ, while

$$\frac{\{\delta \in \Delta \ : \ c^\delta_\varphi \wedge \varphi \vdash \varphi'\} \in \mathcal{D}_\prec}{\{\delta \in \Delta \ : \ c^\delta_{\varphi\wedge\varphi'} \wedge (\varphi^\wedge\varphi') \vdash c^\delta_\varphi\} \in \mathcal{D}_\prec}$$

is a consequence of (Ref), (CM), (And) and the definition of c^δ_φ. $\dashv$

3.3. Extensions. Insofar as neither Δ nor $\prec$ need mention models, it is perhaps unsuitable to call Theorem 4 a "completeness theorem." Moreover, §3.2 is hardly "complete" in that there are at least two obvious directions in which to extend the analysis. Further rules might be added to $\mathbf{C}^-$, and the set Φ might be expanded to cover more of E. Let us take these up briefly, in turn.

The equivalence stated in Theorem 4 can be maintained for any extension of $\mathbf{C}^-$ by rules (over the same set Φ). If necessary, the rules can be translated mechanically into conditions on Δ and $\prec$. Take, for instance, what Kraus, Lehmann and Magidor [**9**] call *preferential reasoning*, which is characterized by adding to $\mathbf{C}$ the rule

$$\text{(Or)} \qquad \frac{e_1 \mid\!\sim e \qquad e_2 \mid\!\sim e}{e_1 \vee e_2 \mid\!\sim e} \ .$$

Put in terms of $(\Delta, \prec)$, this rule becomes simply

$$\text{(OR)} \ \frac{\{\delta \in \Delta : c^\wedge e_1 \vdash^\delta e\} \in \mathcal{D}_\prec \quad \{\delta \in \Delta : c^\wedge e_2 \vdash^\delta e\} \in \mathcal{D}_\prec}{\{\delta \in \Delta : c^\wedge(e_1 \vee e_2) \vdash^\delta e\} \in \mathcal{D}_\prec}$$

and if we are content with such a "syntactic" requirement on $(\Delta, \prec)$, we need work no further for a representation of $\mathbf{P}$. (A parallel here might be drawn with the manner in which the completeness theorem for first-order logic applies to

[5] Observe that the reversal of (LLE) would not have been possible had $c^{\delta_i}_\varphi$ been defined instead as $\bigwedge(\Gamma_\varphi \cap \{\psi_0, \ldots, \psi_i\})$.

different first-order theories.) Of course, a little labor is always good, and a simple restatement of (OR) might be illuminating. Or perhaps with a bit more effort, we might even strengthen (OR) by a sufficient condition met by the model constructed in the proof of Theorem 4, adapted to **P**.[6] Following a suggestion of D. Lehmann's (made in a conversation with me) that (Or) be regarded as a basic rule, consider the system $\mathbf{P}^-$ obtained from **P** by replacing (CM) by (And). As proved in page 191 of Kraus, Lehmann and Magidor [**9**], $\mathbf{P}^-$ enjoys *cut-elimination* — which is to say that $\mathbf{P}^-$ can be redefined as $\mathbf{C}^-$ with (Cut) replaced by (Or)

$$\mathbf{P}^- \quad = \quad \{\text{(Ref), (LLE), (RW), (And), (Or)}\} \ .$$

Now, improving on condition (iii) of $\mathbf{C}^-$-sufficiency, define a pair $(\Delta, \prec)$ to be $\mathbf{P}^-$-*sufficient* if

(i) $\prec$ is transitive,
(ii) for all $\varphi, \varphi' \in \Phi,\ \delta \in \Delta$ and $e \in E$,

$$\frac{\varphi \equiv \varphi'}{\delta(\varphi, e) \equiv \delta(\varphi', e)}$$

and

(iii) for all $\varphi_1, \varphi_2, \in \Phi,$ and $\delta \in \Delta$,

$$\delta(\top, c^\wedge(\varphi_1 \vee \varphi_2)) \quad \vdash \quad \delta(\top, c^\wedge\varphi_1) \vee \delta(\top, c^\wedge\varphi_2) \ .$$

The arguments in §§ 3.1 and 3.2 can be modified easily to prove

Theorem 5. *For every $\mathbf{P}^-$-sufficient pair $(\Delta, \prec)$, the relation $\{(\varphi, \varphi') \in \Phi \times \Phi : c^\wedge\varphi \vdash_\prec \varphi'\}$ verifies $\mathbf{P}^-$. Conversely, if Φ is countable, then for every $\mid\!\sim\ \subseteq \Phi \times \Phi$ verifying $\mathbf{P}^-$, there is a $\mathbf{P}^-$-sufficient pair $(\Delta, \prec)$ such that*

$$\varphi \mid\!\sim \varphi' \quad \textit{iff} \quad c^\wedge\varphi \vdash_\prec \varphi'$$

for all $\varphi, \varphi' \in \Phi$.

A second direction in which to push the present analysis of $\mid\!\sim$ is to enlarge the set of formulas to E. This is, in my mind, no mean enterprise, requiring as it does some fundamental reflection on the nature of ambiguity, and on the plausibility that $\vdash^{\mathcal{D}}$ should be "logical" (in any recognizable sense). Two assumptions made in the present paper, for instance, deserve more careful consideration: (i) the claim that the "partial" character of (the "actual" process of) disambiguation can be modeled through sets of total disambiguations, and (ii) the assumption that $\mathcal{D}$ is closed under intersections. In confining its attention to the disambiguation of a rogue context c, the present paper takes the tiniest bite at the monster that is ambiguity; the reader with an appetite for more is referred to Fernando [**4**].

[6] Notice, for instance, that condition (C3) (in the definition of $\mathbf{C}^-$-sufficiency) is stronger than the mechanical translation of (LLE), but that we have taken pains to construct, in the proof of Theorem 4, an interpretation $(\Delta, \prec)$ satisfying (C3).

4. Discussion

The basic hypothesis about non-monotonicity explored above is, very briefly, that non-monotonicity arises from the non-persistent interpretation (— identified here with disambiguation —) of implicit (background) conditions. To help us digest that point, let us try to put things in their proper context.

4.1. Contexts: between models and formulas. Few would dispute the claim that contexts are partial, but just what the consequences of this claim are varies widely from researcher to researcher. For the approach developed above, it is crucial that this partiality *not* be confused with uncertainty *about* contexts, as such a distinction is necessary in formulating the following principle of *persistence*

$$\frac{\varphi \vdash \varphi'}{\delta(\varphi, e) \vdash \delta(\varphi', e)} \; .$$

The failure (above) of persistence gives rise to the essential asymmetry between the rules (LLE) and (RW) of Kraus, Lehmann and Magidor [**9**] (— an asymmetry that all other rules can be viewed as being devised to diminish). In fact, whereas the rule (RW) is, in the present work, an immediate consequence of the innocuous assumption that only "large" sets can contain "large" sets, the rule (LLE) must, weak as it is, be introduced through condition (C3). (C3) can be restated (as in the definition of $\mathbf{P}^-$-sufficiency) as

$$\frac{\varphi \equiv \varphi'}{\delta(\varphi, e) \equiv \delta(\varphi', e)} \; ,$$

asserting that, so far as disambiguation is concerned, an unambiguous formula matters only up to its set of models. That is, a perfectly finitary formula can be replaced by an infinite set (if not proper class) of its possibly infinite models. Of course, only the most rabid semanticist could (shamelessly) describe such a move as a "reduction." On the other hand, there is no denying the widespread bias for models over formulas, manifested, for instance, by the fact that preferences were initially defined on models (rather than on formulas). We may well expect preferences on infinite sets of formulas (corresponding to theories of models) to be reducible to preferences on (single) formulas (corresponding to basic open sets of models), if only because compactness has become so familiar.[7] But defining preferences between states "labeling" the same sets of formulas (as in the preferential models constructed in §5.3 of Kraus, Lehmann and Magidor [**9**]) is a different matter altogether, requiring a shift in perspective. Staying with our interpretation of $\mid\!\sim$ in terms of Δ and $\prec$, it perhaps bears repeating that in view of the dispensability of models to that interpretation, it is more appropriate to describe Theorem 4 as a "representation theorem," rather than a "completeness theorem." Indeed, I conceived

[7]Passing to effectively presented sets, there is also Kleene's theorem stating that a recursively enumerable first-order theory is equivalent to a first-order sentence, allowing an expansion in vocabulary.

Theorem 4 originally to serve a certain *representationalist* end, establishing the eliminability of models (poorly suited, as they are, for mechanization).

4.2. Discourse representation. Although a model-theoretic interpretation of a notion $\vdash \subseteq \Phi \times \Phi$ of consequence allows formulas (in Φ) to be replaced by models, it need not imply that $\vdash$ is classical. In the case of *Discourse Representation Theory* (DRT, Kamp and Reyle [**8**]), for instance, an equivalence between model-theoretic and syntactic accounts of Φ was more or less obvious before a classical consequence relation $\vdash$ was isolated. The simple reason was that the model-theoretic interpretation commonly associated with the DRT merge $^\wedge$ is not commutative. More precisely, a certain first-order fragment Φ of DRT can be supplied a semantics $[\![\cdot]\!]$, under which

(i) every $\varphi \in \Phi$ is interpreted as a binary relation $[\![\varphi]\!] \subseteq S \times S$ on a certain fixed set S of states (constructed from models), with the intuition that

$$s[\![\varphi]\!]s' \quad \text{iff} \quad \begin{array}{l}\text{given an initial context } s,\\ \varphi \text{ can return the context } s'\end{array}$$

for all $s, s' \in S$,

and

(ii) $^\wedge$ is relational composition

$$[\![\varphi^\wedge\varphi']\!] \;=\; \{(s,s') \in S \;:\; (\exists s'')\; s[\![\varphi]\!]s'' \text{ and } s''[\![\varphi']\!]s'\}\;,$$

for all $\varphi, \varphi' \in \Phi$.

Nevertheless, it turns out that the relational semantics $[\![\cdot]\!]$ is determined by $^\wedge$ and a set $\Phi_\perp \subseteq \Phi$ of "absurd" formulas (Fernando [**2**]). The situation is completely analogous to the way conjunction and negation determine Boolean connectives. Hence, it is perhaps not terribly surprising that, for this case, relational semantics can be reconciled with Boolean-valued semantics (Fernando [**3**]). That is to say, a classical notion $\vdash$ of consequence can be defined for an unambiguous fragment Φ of DRT.

Far more interesting, however, are parts of DRT that go beyond Φ. Two examples are *Segmented* DRT (SDRT, Asher [**1**]) and treatments of plurals. In SDRT, a non-monotonic entailment relation $\mid\!\sim_{KB}$ relative to some knowledge base *KB* lies at the heart of the discourse interpretation process (involving so-called discourse relations). The idea is to put in the lefthand side of $\mid\!\sim_{KB}$ not only the natural language utterance to be interpreted, but also the "common ground" — that is, the discourse context. Now, it is natural to hypothesize that "conversational leaps" in the common ground give rise to the rogue context c (above). In particular, the presupposition associated with the utterance of a sentence φ at context K might be expressed as c so that

$$K, \varphi \mid\!\sim_{KB} K' \quad \text{iff} \quad K, c, \varphi \vdash_{KB} K'$$

for all (candidate consequences) K'. As for plural statements, ambiguity is a notorious problem, motivating, for example, the work of Reyle [**13**]. And whether or not the difficulty is labeled to be one of "ambiguity," interpretation does not become any easier where there is no explicit quantification, as in the case of

`Birds fly.`

In contrast to the case of $\mid\!\sim_{KB}$, it would seem useful to formulate such "bare plurals" as conditional expressions, rather than as judgments about $\mid\!\sim$. In any event, a closer investigation of $\mid\!\sim$, including its relation with conditionals, is clearly called for.

4.3. Smoothness, optimal disambiguations and generics. The assumption in abduction that there is a most preferred explanation, the *limit assumption* in Stalnaker [**15**], and the "smoothness condition" in Kraus, Lehmann and Magidor [**9**] correspond in the present approach to the following notion. A pair $(\Delta, \prec)$ is said to be *smooth at* $(\varphi, e) \in \Phi \times E$ if

$$(\forall \delta \in \Delta)(\exists \delta' \preceq \delta)(\forall \delta'' \preceq \delta') \qquad \delta'(\varphi, e) \equiv \delta''(\varphi, e) \ .$$

That is, there is a set $\Psi_{\varphi,e} \subseteq \Phi$ such that $\{\delta \in \Delta : (\exists \psi \in \Psi_{\varphi,e})(\forall \delta' \preceq \delta)\ \delta'(\varphi, e) \equiv \psi\}$ is stably $\prec$-dense. (Note that the particular pair $(\Delta, \prec)$ constructed in §3.2 is smooth at (φ, c) iff Γ_φ is equivalent to a formula.) Specializing to the case $e = c$ relevant to our analysis of $\mid\!\sim\ \subseteq \Phi \times \Phi$, call a pair $(\Delta, \prec)$ *c-smooth* if it is smooth at (φ, c), for every $\varphi \in \Phi$.

Proposition 6. *If $(\Delta, \prec)$ is c-smooth, then for every $\varphi \in \Phi$, there is a set $\Psi_\varphi \subseteq \Phi$ of unambiguous formulas such that*

$$c^\wedge \varphi \vdash_\prec \varphi' \quad \text{iff} \quad (\forall \psi \in \Psi_\varphi)\ \psi \vdash \varphi \supset \varphi' \ ,$$

for every $\varphi' \in \Phi$.

Proposition 6 not only reduces $\mid\!\sim$ to material implication, but also characterizes the background condition as a disjunction $\bigvee \Psi_\varphi$ (which can be formalized, assuming the disjunction can be formed; e.g., if Ψ_φ is finite). Observe that the background condition is defined relative to the antecedent φ, and that the more general case where antecedents and/or succedents might be ambiguous requires a notion of an "optimal disambiguation" that checks every ambiguous expression (not just c). Whether or not optimal choices at every pair (φ, e) (for all $e \in E$ as well as $\varphi \in \Phi$) can be assembled in a disambiguation presumably depends on the application at hand.

Returning to the simple case of c, the following example is instructive, particularly for a comparison with Kraus, Lehmann and Magidor [**9**].

Example. Let Φ be the set of propositions generated by a countable set $\{p_0, p_1, \dots\}$ of propositional variables, and let $\mathcal{L}$ be the first-order language induced by a countable set $\{U_0, U_1, \dots\}$ of unary relation symbols. Fix a constant symbol a, and define the translation $\cdot^a$ from Φ to $\mathcal{L}(a)$ as follows

$$\begin{aligned} {p_n}^a &= U_n(a) \\ (\varphi \wedge \psi)^a &= \varphi^a \wedge \psi^a \\ (\neg\varphi)^a &= \neg(\varphi^a) \ . \end{aligned}$$

Next, let M be the $\mathcal{L}$-model with universe ω $(= \{0, 1, \dots\})$ that interprets U_n as $\{m \in \omega : m \geq n\}$, for every $n < \omega$. Form the $\mathcal{L}(a)$-theory T by adding to the $\mathcal{L}$-theory of M the $\mathcal{L}(a)$-sentences $U_n(a)$, for every $n < \omega$:

$$T \;=\; \{\chi \in \mathcal{L} : M \models \chi\} \;\cup\; \{U_0(a), U_1(a), \dots\}$$

(which is consistent, by compactness). Define

$$\varphi \mathrel{|\!\sim} \psi \quad \text{iff} \quad T \vdash_{\mathcal{L}(a)} \varphi^a \supset \psi^a \ ,$$

for all $\varphi, \psi \in \Phi$, and observe that $|\!\sim$ satisfies **C** (including (CM)), but cannot be captured by a c-smooth pair $(\Delta, \prec)$, since T is not satisfiable in M (but only finitely satisfiable there). [End of Example]

By contrast, note that in Kraus, Lehmann and Magidor [**9**], a similar "smoothness condition is necessary to ensure the validity of Cautious Monotonicity" (page 182). A notion central to the representation arguments there is that of a *normal model of* φ, by which is meant a model of $\{\varphi' : \varphi \mathrel{|\!\sim} \varphi'\}$. This leads to so-called *strongly cumulative models* (the essential property of which is that every antecedent φ has a minimum state), because normal models contain infinite information, whereas disambiguations only encode single formulas (in values assigned to pairs $(\varphi, e) \in \Phi \times E$). As it turns out, it is fruitful to introduce *types* (as they are known in model theory) to express more information than can be packaged in single first-order formulas. The question is, however, what are these types *of*? As has already been mentioned, preferences in Kraus, Lehmann and Magidor [**9**] can be defined between states labeling the same sets of formulas. (This, at least, is the case for the preferential models in §5.3, though not, as noted in Voorbraak [**16**], for the cumulative models in §3.5.) In conceptual terms, what do such states reflect? Toward an answer, Fernando [**5**] treats these states as objects in a first-order model, explores a notion of *generic* object (or better: *type*) that reduces defeasibility to material implication, and presents a proposal relating entailment $|\!\sim$ to a connective producing conditional expressions. (At the risk of sounding obscure, suffice it to say that what is gained by introducing a preferential relation is the possibility of *omitting the type*.)

Acknowledgments

My thanks to Nicholas Asher, Hans Kamp, Emil Weydert and Daniel Lehmann (who suggested $\mathbf{P}^-$ in §3.3, among other things) for helpful discussions.

References

[1] Nicholas Asher. *Reference to Abstract Objects in Discourse.* Kluwer Academic Publishers, Dordrecht, 1993.

[2] Tim Fernando. What is a DRS? In R. Cooper and J. Groenendijk, editors, *Integrating Semantic Theories II.* Dyana deliverable R2.1.B, 1994. Presented in a Computational Semantics workshop at Tilburg (December 1994).

[3] Tim Fernando. A persistent notion of truth in dynamic semantics. In D. Westerståhl and J. Seligman, editors, *Language, Logic and Computation: The 1994 Moraga Proceedings*. CSLI, Stanford, To appear.

[4] Tim Fernando. Ambiguity under changing contexts. Manuscript, 1995. Presented in *Mathematics of Language 4*, Philadelphia, October 1995.

[5] Tim Fernando. Defeasible generic types. Manuscript, 1995.[8]

[6] Dov M. Gabbay. Theoretical foundations for non-monotonic reasoning in expert systems. In K.R. Apt, editor, *Proc. NATO Advanced Study Institute on Logics and Models of Concurrent Systems*. Springer Verlag, Berlin, 1985.

[7] J. Hobbs, M. Stickel, D. Appelt, and P. Martin. Interpretation as abduction. *Artificial Intelligence*, 63, 1993.

[8] H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht, 1993.

[9] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44, 1990.

[10] David Makinson. General theory of cumulative inference. In M. Reinfrank, J. de Kleer, M.L. Ginsberg, and E. Sandewall, editors, *Proc. Second International Workshop on Non-monotonic Reasoning*, Lecture Notes in Computer Science. Springer Verlag, Berlin, 1989.

[11] Charles Sanders Peirce. Abduction and induction. In J. Buchler, editor, *Philosophical Writings of Peirce*. Dover Books, New York, 1955.

[12] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13, 1980.

[13] Uwe Reyle. Dealing with ambiguities by underspecification: construction, representation and deduction. *Journal of Semantics*, 10(2), 1993.

[14] Yoav Shoham. *Reasoning about Change*. MIT Press, Cambridge, MA, 1988.

[15] Robert Stalnaker. A theory of conditionals. In *American Philosophical Quarterly Monograph Series 2*. Blackwell, Oxford, 1968.

[16] Frans Voorbraak. Preference-based semantics for non-monotonic logics. In *Proc. 13th IJCAI*. Morgan Kaufmann, San Mateo, CA, 1993.

[8] *Added, February 1997*. This manuscript has since been superseded by "In conjunction with qualitative probability."

8

Analyzing ambiguity denotationally, operationally and logically

Abstract. Ambiguous expressions are interpreted relative to notions of disambiguation, conceived as processes, and analyzed, in turn, under denotational, operational and logical perspectives (following the custom in programming language theory).

Keywords. Underspecified representations; discourse interpretation; disambiguation; denotational, operational and logical semantics

1. Introduction

A good deal has been written, under the shadow of Montague semantics, about just how compositional and non-representa-tional *Discourse Representation Theory* (DRT, Kamp and Reyle [**18**]) is (or can, in principle, be made). A recent paper, which provides many (other) references, is Muskens [**21**]. In it is presented a formalism called *Compositional DRT*, which leaves the question of non-representationality somewhat open, but provides a compositional analysis of (various) sentences that are assumed to come indexed, as in

$$\texttt{Sue}^1\ \texttt{has a pigeon}^2.\quad \texttt{She}_1\ \texttt{feeds it}_2.$$

Precisely how the anaphoric linkages are established is left to some magic black box, falling completely outside the scope of the formalism (as has become customary in dynamic semantics). Anaphora resolution is, however, hardly an insignificant matter, having, for instance, been implicated in presupposition projection (van der Sandt [**24**]). Nor is it a straightforward matter, involving, as it does, the thorny possibility of ambiguity. Setting aside troublesome cases of presupposition that can arguably be ascribed to ambiguity, it is easy to produce simple examples such as

`Ernest has a brother; he feeds him`

where it is anyone's guess, who feeds who.

A formalism covering preferences in anaphora resolution is proposed in Jaspars and Kameyama [**17**], extending "a fragment of van Benthem's and de Rijke's dynamic modal logic with additional preferential operators." A different approach is pursued in the present paper, which (to simplify notation) builds on first-order-like formulas closer to the reformulation in Groenendijk and Stokhof [**14**] of a fragment of DRT. Precious little is said below about the (undoubtedly important) problem of preferentially resolving ambiguous pronouns in discourse. The emphasis rather is on shedding some light on basic conceptual questions (such as compositionality and non-representationality) in relation to various forms of ambiguity — be that pronominal, lexical (e.g., river `bank` versus money `bank`) or structural (e.g., quantifier scope). Whether or not an instance of ambiguity can be traced to words or to sentential operators, the challenge to compositionality (taken up in §2) is that disambiguation often requires an examination of larger units of discourse.

The reader should be warned that a detailed study of a particular process of disambiguation is *not* undertaken below. What is offered instead is a hook (onto dynamic semantics) upon which *any* of one's favorite disambiguation strategies can be hung, thereby facilitating a comparison of these strategies within dynamic semantics. Disambiguation strategies may differ, for instance, as to *monotonicity*, an issue discussed in (among other places) Poesio [**23**]. That issue will be considered in §3 under distinct *operational* and *logical* perspectives, bearing also upon the question of non-representationality. But first, let us attend to the more rudimentary matter of assigning an ambiguous expression meaning — that is to say, *denotational* semantics.

2. Regular constructs for ambiguity

Consider the case of an expression e that is ambiguous (say) between two unambiguous formulas φ and ψ. A first attempt at assigning a meaning to e is to characterize e as the disjunction $\varphi \vee \psi$. As natural as it may seem, this idea has, nonetheless, been discredited on many grounds (e.g., van Deemter [**7**]), not the least of which is that whereas e is ambiguous, the disjunction $\varphi \vee \psi$ is unambiguous — assuming a compositional analysis of $\vee$ under which the meaning of $\varphi \vee \psi$ is determined (unambiguously) by the meanings of its parts, φ and ψ. That is, more concretely, if the meaning $[\![\phi]\!]$ of an unambiguous formula ϕ is taken to be the set of "states" in which ϕ is "true" then common practice has it that for all unambiguous expressions φ and ψ,

$$[\![\varphi \vee \psi]\!] \quad = \quad [\![\varphi]\!] \cup [\![\psi]\!]$$

(whence $\varphi \vee \psi$ is no more ambiguous than $\varphi \wedge \psi$). By contrast, van Deemter [**7**] (p. 234) proposes a "meta-or" connective $|$ such that

(1) $$[\![\varphi|\psi]\!] = [\![\varphi]\!] \quad \text{or} \quad [\![\varphi|\psi]\!] = [\![\psi]\!] \ .$$

Converting (1) from a disjunction into a definition, and writing $\bullet$ instead of $|$, Fernando [**9**] defines the meaning of $\varphi \bullet \psi$ relative to a binary relation $\prec$ on unambiguous formulas that chooses between φ and ψ as follows

$$[\![\varphi \bullet \psi]\!]_{\prec} \quad = \quad \begin{cases} [\![\varphi]\!] & \text{if } \varphi \prec \psi \\ [\![\psi]\!] & \text{otherwise.} \end{cases}$$

More generally, if $[\![\varphi]\!]_{\prec}$ is set to $[\![\varphi]\!]$ for every unambiguous formula φ, and $\prec$ is extended to formulas e and e' that might be built with $\bullet$, then

(2) $$[\![e \bullet e']\!]_{\prec} \quad = \quad \begin{cases} [\![e]\!]_{\prec} & \text{if } e \prec e' \\ [\![e']\!]_{\prec} & \text{otherwise.} \end{cases}$$

(2) is broadly reminscent of Kripke semantics for modal operators, the essential difference being that whereas an accessibility relation is a binary relation on

semantic entities (called worlds), the binary relation $\prec$ in (2) relates the very syntactic entities whose meanings are to be defined. This difference casts doubt on the compositionality of (2): is there a function $F_\bullet$ such that for all e and e',

$$[\![e \bullet e']\!]_\prec \quad = \quad F_\bullet([\![e]\!]_\prec, [\![e']\!]_\prec) \ ?$$

(The problem is that the test $e \prec e'$ in (2) may be impossible to carry out if e and e' are reduced to $[\![e]\!]_\prec$ and $[\![e']\!]_\prec$ respectively.) Whether or not (2) supports a compositional interpretation of $\bullet$ will obviously depend on $\prec$; the technical details of that dependence are taken up at some length in Fernando [**9**] (where slight modifications of (2) are considered).

Turning to the question of which relations $\prec$ are linguistically reasonable, observe that any number of factors come to play, that together might be conveniently labeled context.[1] Among these factors is the surrounding text, as can be seen by formulating the discourse

(3) `Ernest has a brother; he is happy`

as

(`Ernest`^{1} `has a`^{2} `brother; he`$_1$ `is happy`) $\bullet$
(4) (`Ernest`^{1} `has a`^{2} `brother; he`$_2$ `is happy`)

and comparing

`Ernest has a brother; he is happy;`
(5) `he'd feel lonesome otherwise`

with

`Ernest has a brother; he is happy;`
(6) `Ernest feeds him.`

The point of this example is that a preference for an interpretation of (4) relative to a relation $\prec$ such that

`Ernest`^{1} `has a`^{2} `brother; he`$_1$ `is happy` $\prec$
`Ernest`^{1} `has a`^{2} `brother; he`$_2$ `is happy`

can be maintained under (5) but not under (6). (The reader is invited to cook up examples, where the disambiguating text appears to the left.) The more general question is: *can the contribution of the surrounding text to disambiguation be brought within the compass of a compositional analysis of ambiguous expressions?*

[1] The ambiguity connective $\sharp$ of König and Reyle [**19**] is a notational variant of $\bullet$, requiring a context κ for interpretation. König and Reyle write $[\![A_1 \sharp A_2]\!]^\kappa$ for what would be written $[\![A_1 \bullet A_2]\!]_{\prec^\kappa}$ in Fernando [**9**].

To answer this question, it is useful to step back for a moment, and recall what the principle of *compositionality* says

(C) the meaning of an expression is a function of the meaning of its parts.

Now given the problem of determining the meaning of an expression e, the usual way to apply (C) is to decompose e into its parts, and determine the meanings of those parts. More specifically, to assign a meaning to an ambiguous expression e, one might build e up from unambiguous parts (with the understanding that a formula is unambiguous precisely because it has a determinate meaning). This is exactly the line taken above in characterizing an expression e ambiguous between φ and ψ as $\varphi \bullet \psi$, where the function mentioned in (C) is the connective $\bullet$ (disjunction $\vee$ having previously been deemed unacceptable). There is, however, another way (C) can be applied to assign e a meaning. Rather than breaking e down into its parts, e can be combined with other expressions to form an unambiguous whole. Consider again the discourse labeled (3) above. In isolation, (3) is ambiguous, having an indeterminate meaning. But when merged with other sentences, as in (5) and (6), (3) becomes unambiguous. Relating this example back to (C) and our arbitrary ambiguous expression e, the point is to identify the "function" mentioned in (C) not with a meta-disjunction $\bullet$ (as in the formalization (4) of (3)), but rather with a dynamic conjunction, denoted $^\wedge$, that merges e with other expressions, yielding, in some cases, an unambiguous whole. More formally, writing Φ for the set of unambiguous formulas, and $E \supset \Phi$ for the union of Φ with the set $(E - \Phi)$ of ambiguous expressions, we have

$$\begin{aligned} \texttt{Ernest}^1 \texttt{ has a}^2 \texttt{ brother} &\in \Phi \\ \texttt{he is happy} &\in E - \Phi \\ \texttt{he'd feel lonesome otherwise} &\in E - \Phi \\ \texttt{Ernest}^1 \texttt{ has a}^2 \texttt{ brother} \,^\wedge\, \texttt{he is happy} &\in E - \Phi \end{aligned}$$

but (let us agree)

$$\begin{aligned} (\texttt{Ernest}^1 \texttt{ has a}^2 \texttt{ brother} \,^\wedge\, \texttt{he is happy}) \,^\wedge & \\ \texttt{he'd feel lonesome otherwise} &= \\ \texttt{Ernest}^1 \texttt{ has a}^2 \texttt{ brother} \,;\, \texttt{he}_1 \texttt{ is happy} \,; & \\ \texttt{he}_1\texttt{'d feel lonesome otherwise} &\in \Phi\,. \end{aligned}$$

Now, an assignment of a meaning $[\![e]\!]_\wedge$ to an expression $e \in E$ is compositional relative to $^\wedge$ precisely when the equivalence relation $\equiv$ on E defined by

$$\text{for all } e, e' \in E, \qquad e \equiv e' \quad \text{iff} \quad [\![e]\!]_\wedge = [\![e']\!]_\wedge$$

is a *congruence* relative to $^\wedge$ — i.e., for all $e_1, e_1', e_2, e_2' \in E$,

$$e_1 \equiv e_1' \text{ and } e_2 \equiv e_2' \quad \text{imply} \quad (e_1{}^\wedge e_2) \equiv (e_1'{}^\wedge e_2')\,.$$

Of course, for the meaning $[\![e]\!]_\wedge$ of e to "respect" the meaning $[\![\varphi]\!]$ of an unambiguous formula φ ($\in \Phi$), we should at the very least require that

$$(7) \qquad \text{for all } \varphi, \psi \in \Phi, \qquad \varphi \equiv \psi \quad \text{implies} \quad [\![\varphi]\!] = [\![\psi]\!] \ .$$

Strengthening (7) to a bi-conditional is problematic, as there need *not* exist a congruence $\equiv$ relative to $^\wedge$ satisfying the converse of (7) (i.e., the requirement that $[\![\cdot]\!]_\wedge$ be compositional may well be incompatible with the condition that for all $\varphi \in \Phi$, $[\![\varphi]\!]_\wedge = [\![\varphi]\!]$). On the other hand, not only is a congruence relative to $^\wedge$ satisfying (7) guaranteed to exist, but there is, in fact, a ($\subseteq$-)largest one, which constitutes a natural notion of semantic equivalence on E. The gory details are spelled out in Fernando [**10**]; let us mention only a few concepts from that work which will be useful for the sequel.

Keeping in mind that an expression may become unambiguous only after it is merged with some finite number of expressions, let us define a *left* $^\wedge$-*disambiguation of* an expression $e \in E$ to be a pair $(e_1 \cdots e_n, \varphi) \in E^* \times \Phi$, the first component of which can be put to the left of e to yield an unambiguous formula, given by the second component of the pair — i.e.,

$$(\cdots(e_1^\wedge e_2)^\wedge \cdots {}^\wedge e_n)^\wedge e \ \ = \ \ \varphi \ .$$

For example, if e_1 is `Ernest`[1] `has a`[2] `brother`, e_2 is `he is happy`, and φ is

`Ernest`[1] `has a`[2] `brother; he`$_1$ `is happy; he`$_1$`'d feel lonesome otherwise`

then $(e_1 e_2, \varphi)$ is a left $^\wedge$-disambiguation of

`he'd feel lonesome otherwise`.

Similarly, call a pair $(e_1 \cdots e_n, \varphi) \in E^* \times \Phi$ a *right* $^\wedge$-*disambiguation of* an expression $e \in E$ if

$$(\cdots(e^\wedge e_1)^\wedge \cdots {}^\wedge e_{n-1})^\wedge e_n \ \ = \ \ \varphi \ .$$

Taking the pathological case where $n = 0$, and writing $\emptyset$ for the null sequence, note that for any $e \in E$,

$$\begin{aligned} (\emptyset, e) \text{ is a right } {}^\wedge\text{-disambiguation of } e \quad & \text{iff} \\ e \in \Phi \quad (\text{i.e., } e \text{ is unambiguous}) \quad & \text{iff} \\ (\emptyset, e) \text{ is a left } {}^\wedge\text{-disambiguation of } e \ . & \end{aligned}$$

For an *ambiguous* expresion e, however, the case of $n > 0$ is necessary, and both left and right $^\wedge$-disambiguations are useful to test the two argument places of $^\wedge$. (Obviously, it would be absurd to assume that $^\wedge$ is commutative.) Indeed, we may wish to apply sequences $e_1 \cdots e_n$ and $e_{n+1} \cdots e_{n+m}$ of expressions to the left and right of e, the crucial point being that the result be unambiguous

$$(\cdots(((\cdots(e_1^\wedge e_2)^\wedge \cdots {}^\wedge e_n)^\wedge e)^\wedge e_{n+1})^\wedge \cdots {}^\wedge e_{n+m}) \ \ \in \ \ \Phi \ .$$

In the absence of the assumption that $^\wedge$ is associative, there are also variations in parenthesization to consider, leading to a proliferation of notions of disambiguation. Fortunately (and somewhat surprisingly perhaps), the intuition that

(S) the meaning of an ambiguous expression e is the set of its disambiguations

can be formalized simply enough, with

(i) "set" construed literally albeit non-standardly under a conception allowing for non-well-foundedness (Aczel [**1**]),

and

(ii) "disambiguation" derived from a right[2] $\wedge$-disambiguation $(e_1 \cdots e_n, \varphi)$ of e, by turning each e_i into its meaning, and φ into $[\![\varphi]\!]$.

That is, within the set theory described in Aczel [**1**], an ambiguous expression e can be assigned a meaning $[\![e]\!]_\wedge$ such that, in accordance with (S),

$$[\![e]\!]_\wedge \;=\; \{([\![e_1]\!]_\wedge \cdots [\![e_n]\!]_\wedge, [\![\varphi]\!]) \;:\; (e_1 \cdots e_n, \varphi) \text{ is a right } {}^\wedge\text{-disambiguation of } e\} \;. \tag{8}$$

By contrast, observe that characterizing an expression e ambiguous between φ and ψ as $\varphi \bullet \psi$ corresponds, under (S), to setting

$$\text{the meaning of } e \;=\; \{[\![\varphi]\!], [\![\psi]\!]\} \;,$$

which provides no clue at all as to when e can be reduced to φ, and when it must be reduced to ψ. On the other hand, what is funny about (8) is that given a right $\wedge$-disambiguation $(e_1 \cdots e_n, \varphi)$ of e, the meaning $[\![\varphi]\!]$, buried in the right side of (8), encodes not just e, but also the preceding text $e_1 \cdots e_n$. Indeed, the question arises as to whether or not the contribution e makes to φ can be separated out from that of $e_1 \cdots e_n$. This, in turn, leads (somewhat belatedly perhaps) to the query, where does $\wedge$ come from, in the first place? What makes $\wedge$ particularly mysterious is that it may merge two ambiguous expressions e and $e' \in E - \Phi$ into an unambiguous expression $e {}^\wedge e' \in \Phi$. How could $\wedge$ be defined by induction on expressions, given that an unambiguous expression $e {}^\wedge e'$ can be built from ambiguous parts, e and e'?

To uncover that inductive definition, it is useful to assume that the set $E \supset \Phi$ of expressions is defined inductively from Φ, with closure conditions for (among other things) binary connectives $\bullet$ and ;. The latter connective is a dynamic conjunction similar to $\wedge$, except that for all $e, e' \in E$, if $e; e'$ is unambiguous then so is each of e and e' — i.e.,

$$e \in E - \Phi \;\text{ or }\; e' \in E - \Phi \quad \text{implies} \quad e; e' \in E - \Phi \;.$$

To cover examples involving anaphora resolution, let us assume also that for every variable x in some fixed set X, the string $x =?$ is in E. Borrowed

[2] It is also possible to use left $\wedge$-disambiguations, instead of right ones. What matters is that the system of equations, the roots of which are the required meanings, is solved simultaneously; see Fernando [**10**].

from informal DRT practice, the expression $x =?$ is employed to represent an unresolved pronoun (e.g., Asher [**4**]). More specifically, the idea is that

$$\texttt{Ernest}^1\ \texttt{has a}^2\ \texttt{brother}\ \ \wedge\ \ \texttt{he is happy}$$

is equal to

$$\texttt{Ernest}^1\ \texttt{has a}^2\ \texttt{brother}\ \ ;\ \ (\texttt{he}_1\bullet\ \texttt{he}_2)\ \texttt{is happy}$$

but not to

$$\texttt{Ernest}^1\ \texttt{has a}^2\ \texttt{brother}\ \ ;\ \ \texttt{he is happy}$$

(where the pronoun `he` remains un-indexed). Next, to bridge the gap between $\wedge$ and ;, let us specify a certain "disambiguation strategy" a that induces a function $(\cdot,\cdot,\cdot)_a$ from $E \times E \times E$ to E, satisfying the intuition that for all $l, e, r \in E$,

$$(l,e,r)_a \ \approx\ \text{the expression to which } a \text{ reduces } e\text{, given that } l \text{ occurs to } e\text{'s left, and } r \text{ occurs to its right}$$

where the second argument e occupies, so to speak, center stage. Towards a compositional analysis of expressions, let us define $(\cdot,\cdot,\cdot)_a$ by induction on its second argument, setting, in particular,

(9) $$(l,\ e;e',\ r)_a \ =\ (l,\ e,\ e';r)_a\ ;\ (l;e,\ e',\ r)_a$$

(shifting text around so as to reduce the complexity of the second argument). Now, assuming

(†) a is a pair consisting of
- (i) a function from triples $(x,l,r) \in X\times E\times E$ to expressions $\psi_a(x,l,r) \in E$,

and
- (ii) a function from pairs $(l,r) \in E\times E$ to asymmetric binary relations $\prec_a^{l,r}$ on E,

we can then define

(10) $$(l, x =?, r)_a \ =\ \psi_a(x,l,r) \qquad \text{for every } x \in X$$

and

(11) $$(l, e\bullet e', r)_a \ =\ \begin{cases} (l,e,r)_a & \text{if } e \prec_a^{l,r} e' \\ (l,e',r)_a & \text{if } e' \prec_a^{l,r} e \\ (l,e,r)_a \bullet (l,e',r)_a & \text{otherwise} \end{cases}$$

(where (11) is well-defined because $\prec_a^{l,r}$ is assumed to be asymmetric). For example, let e be the expression

$$x =?\ \ ;\ \ \mathsf{happy}(x)$$

formalizing the sentence `he is happy`, let l be the expression

$$\mathsf{Ernest}(y)\ \ ;\ \ \mathsf{brother}(y,z)$$

formalizing `Ernest has a brother`,[3] and r be the experssion

$$u =? \quad ; \quad \textsf{would-feel-lonesome-otherwise}(u)$$

formalizing `he'd feel lonesome otherwise`. Then our analysis above can be supported by setting $\psi_a(x, l, \textsf{happy}(x))_a$ to $x = y \bullet x = z$, and taking

$$x = y \quad \prec_a^{l,r} \quad x = z$$

so that

$$\begin{array}{rcl} (l,\ x =?\ , \textsf{happy}(x))_a & \text{is} & x = y \bullet x = z \\ (l, e, r)_a & \text{is} & x = y\ ;\ \textsf{happy}(x)\ . \end{array}$$

Observe that the function $(\cdot,\cdot,\cdot)_a$ takes values on E, and not necessarily Φ, allowing disambiguation to be a partial process that unfolds gradually (stretching over any number of neighboring expressions). In isolation, the pronoun `he` could mean any male, but within the discourse labeled (3) above, `he` refers either to `Ernest` or his `brother`, and within (5), `he` refers to `Ernest`. By contrast, observe that van Deemter [**7**]'s specification (1) of $\varphi|\psi$ requires that the interpretation function $[\![\cdot]\!]$ be based on a very specific "context" (whatever that might exactly be) which totally disambiguates between φ and ψ. But the more assumptions we make of that context, the less useful that interpretation function becomes. Accordingly, the assumption of total disambiguation built into (1) and (2), is abandoned in (11), which supports instead the possibility of stalling the disambiguation (when, as described by the third case in (11), the expressions l and r to the left and right respectively of the expression $e \bullet e'$ do not provide sufficient information to choose between e and e'). In this way, our formal picture captures more of the process of disambiguation than an interpretation constrained by (1). For the record, that picture is given by

Definition D. *Given some set X of variables and some set Φ of formulas (over X)*[4] *closed under a set of connectives that includes ; and $\neg$, let E be the set of expressions obtained by closing $\Phi \cup \{\text{'}x =?\text{'} : x \in X\}$ under the connectives ;, $\neg$ and, in addition, a binary connective $\bullet$. Then assuming that the condition (†) above holds for some a, define $(\cdot,\cdot,\cdot)_a : (E \times E \times E) \to E$ by induction on*

[3] The precise choice of the variables y and z is unimportant here, so long as it does not clash with other choices (following, for example, the *novelty-familiarity condition* of Heim [**15**]). The (unambiguous) random assignments $y :=?$ and $z :=?$ employed in the reformulation of DRT in Groenendijk and Stokhof [**14**] — and *not* to be confused with ambiguous expressions $x =?$ which lie outside the scope of that reformulation — are suppressed for the sake of simplicity. See footnote 4 below.

[4] It is useful to assume that, for every $x \in X$, the random assignment $x :=?$, mentioned in the previous footnote, is an (atomic) element of Φ. This approach (due to Groenendijk and Stokhof [**14**]) captures "essentially" a first-order fragment of DRT, but throws away the distinction (in DRT) between so-called *discourse representation structures* and *conditions* — a distinction arguably useful for disambiguation, as elaborated in Asher and Fernando [**3**].

its second argument as follows

$$(12) \qquad \begin{aligned} (l, \varphi, r)_a &= \varphi \qquad \text{for } \varphi \in \Phi \\ (l, \neg e, r)_a &= \neg(l, e, r)_a \end{aligned}$$

plus the clauses (9), (10) and (11).

Insofar as $\bullet$ can be likened to non-deterministic choice $+$, ; to concatenation, and $\cdot =?$ to Kleene star $\cdot^*$, Definition D might be said to describe a regular language for ambiguity (due to unresolved variables). The defect in this description, however, is that although $\cdot =?$ provides an *underspecified* representation of an infinitary form of $\bullet$ (just as $\cdot^*$ does of $+$), it does not (alas) apply to expressions, and cannot (as pointed out to me by P. Krause) be viewed as an infinite choice over finite iterations (as in the case of $\cdot^*$). At any rate, the limited expressive power of regular languages suggests enriching these constructs in order to express more complicated conditions (involving, for example, negation or recursion). Confining our attention to modest extensions, one possibility is to sort the variables according (say) to gender and the singular/plural distinction (to support analyses of `she`, `it`, `they`, etc.) — a routine step that is no more troublesome than the move from ordinary (unsorted) predicate logic up to many-sorted predicate logic.

Turning to lexical ambiguity, one might apply $\bullet$ to capture two of the common readings of the noun `bank` as

$$\lambda z \quad (\mathsf{bank}_{\$\$}(z) \;\bullet\; \mathsf{bank}_{\approx}(z)) \ ,$$

and characterize ambiguity in terms of meta-variables, following the approach taken in Alshawi and Crouch [**2**]. Rather than lifting types by λ-abstraction and passing to meta-variables, we can also adopt a first-order combinatory approach, turning a predicate symbol such as $\mathsf{bank}_{\$\$}$ into a constant symbol $bank_{\$\$}$, and reformulating the formula $\mathsf{bank}_{\$\$}(z)$ into the term $APP(bank_{\$\$}, z)$, for some (partial) binary function symbol APP.[5] An occurrence of the ambiguous un-indexed word `bank` can then be formalized as a sorted variable, whose sort restricts it to range over the denotations of $bank_{\$\$}$, $bank_{\approx}$, plus whatever other senses the word `bank` has. In this way, arguably all forms of ambiguity admit a uniform characterization in terms of unresolved variables (complicating, as required, the combinatory manipulations to cover various instances of structural ambiguity).

While the connective $\bullet$ may provide a list of the unambiguous formulas that an ambiguous expression can be reduced to, it is not quite as explicit

[5] The formal approach referred to here goes back to Feferman [**8**], though is perhaps more commonly associated in the linguistic community with *property theory*, and appears (in some form) in Asher [**4**]. What matters is not the technical difference between combinators and λ-abstracts, but rather the move to an expressive *un*typed theory, supporting self-reference. Furthermore, the formalization within first-order (as opposed to higher-order) logic follows the spirit of the "dynamic" shift from function application to relational (sequential) composition as the paradigmatic function mapping, according to the principle of compositionality, the meaning of the parts to the meaning of the whole.

about how to choose between these unambiguous formulas. This is problematic say, for words which are their own antonyms, constituting, as these do, distinct expressions e and $\neg e$ that may both be formulated as $\varphi \bullet \neg\varphi$ (for some φ). To avoid such mix-up's, a different $\bullet$ might be introduced, along with its interpretation $\prec$, for every disambiguation "method." Just how many of these there are is unclear to me. While it would seem that one $\bullet$ should suffice for pronominal resolution, and perhaps another for structural ambiguity, cases such as lexical ambiguity look potentially more problematic (an extreme measure for which would be to have one $\bullet$ for each ambiguous word). In any event, for the remainder of the paper, let us keep matters simple and stick to Definition D, in the hope the more general case of multiple $\bullet$'s is illuminated by the case of a single $\bullet$.

Next, let us return to the issue of compositionality. This has figured prominently in motivating Definition D, although we have yet to supply a compositional interpretation of the expressions specified in Definition D. But there exists, by abstract nonsense worked out in Fernando [**10**] (Theorem 6), a largest equivalence relation $\equiv_a$ on E that (i) is a congruence relative to $(\cdot,\cdot,\cdot)_a$ and to each connective of E, and (ii) satisfies

$$\text{for all } \varphi, \psi \in \Phi, \qquad \varphi \equiv_a \psi \quad \text{implies} \quad [\![\varphi]\!] = [\![\psi]\!] \ .$$

It follows that the (term model) interpretation $[\![\cdot]\!]_a : E \to \mathsf{Pow}(E)$ given by

$$[\![e]\!]_a \quad = \quad \{e' \in E : e \equiv_a e'\}$$

constitutes a compositional interpretation of E, under Definition D, respecting the meanings $[\![\varphi]\!]$ of unambiguous formulas φ ($\in \Phi$). What, however, is perhaps unsatisfying about $[\![\cdot]\!]_a$ is that it is thoroughly "representational" (in that it builds the meaning of an expression from other expressions), and applies to any system of interpretations $[\![\varphi]\!]$ of unambiguous formulas φ, whether or not it has anything at all to do with some conception of truth. Moreover, there is also the presupposition of a disambiguation strategy a mentioned in (†), about which the reader is entitled to inquire.

3. Reasoning with partial disambiguations

There is no denying that Definition D does not reduce the burden of specifying (intuitively) a strategy for disambiguation. The definition of $(\cdot,\cdot,\cdot)_a$ by induction on the second argument simply shoves the complexity in that argument over to the first and third arguments, forcing a particular interpretation of ; (the interpretation of the constructs $\bullet$ and $x =?$ being given freely by a).[6] In order to analyze the first and third arguments, it will prove convenient to adjoin an element $\top$ to Φ (corresponding to $\emptyset$ in E^*), postulating that for

[6] $(\cdot,\cdot,\cdot)_a$ is essentially a repackaging of a, and should not, despite appearances, be mistaken for a context-sensitive grammar.

every $e \in E$,

$$\begin{aligned} \top\,;\,e &= e \\ &= e\,;\,\top \end{aligned}$$

and setting $[\![\top]\!] = [\![true]\!]$ (where *true* is a tautology in Φ). Now, a reduction in the complexity of the first argument of $(\cdot,\cdot,\cdot)_a$ is effected by imposing on a the (*conservative*) constraint that for all $e, e' \in E$,

$$(e, e', \top)_a \quad = \quad ((\top, e, e')_a, e', \top)_a\ ,$$

meaning that $(\top, e, e')_a$ is, for the purpose of left disambiguation, no less informative than e. Similarly, for the third argument (i.e., right side),

$$(\top, e, e')_a \quad = \quad (\top, e, (e, e', \top)_a)_a\ .$$

An "incremental" processing of the sequence $e; e'$ would pre-process the expression e into $(\top, e, \top)_a$, before e' is known, whereupon e' is attached to the right of $(\top, e, \top)_a$. Now, in order that

$$(\top,\ e; e',\ \top)_a \quad = \quad (\top,\ (\top, e, \top)_a; e',\ \top)_a\ ,$$

it suffices that a validate the equations

$$\begin{aligned} (\top, e, e')_a &= (\top,\ (\top, e, \top)_a,\ e')_a \\ (e, e', \top)_a &= ((\top, e, \top)_a,\ e',\ \top)_a\ , \end{aligned}$$

in which case a can be called *incremental* (for e, e'). Next, suppose that an expression e is disambiguated fully into an unambiguous expression $(l, e, e')_a \in \Phi$, under the assumption that l occurs to its left and e' to its right. Then the same principle of incrementality above would suggest that for every possible succeding expression r, $(l,\ e,\ e'; r)_a = (l, e, e')_a$. That is, a might be required to be *right monotonic* in the sense that for all $l, e, r \in E$,

$$(l, e, r)_a \in \Phi \quad \text{implies} \quad (\forall e' \in E)\ \ (l,\ e,\ r; e')_a = (l, e, r)_a\ .$$

The corresponding notion of left monotonicity is that for all $l, e, r \in E$,

$$(l, e, r)_a \in \Phi \quad \text{implies} \quad (\forall e' \in E)\ \ (e'; l,\ e,\ r)_a = (l, e, r)_a$$

(which would be of interest if the directional bias of processing from left to right is weakened).

Observe that the compositional interpretation $[\![\cdot]\!]_a$ of E in §2 does not depend on a respecting any of the constraints mentioned in the preceding paragraph. The justification for these constraints must lie instead with operational intuitions concerning the process of disambiguation. To identify "reasonable" strategies of disambiguation, another source of intuitions that we might tap is truth or logical consequence. Towards that end, rather than proceeding, as in §2, from some completely abstract interpretation $[\![\varphi]\!]$ of unambiguous formulas φ, let us suppose we are given some binary relation $\vdash\ \subseteq \Phi \times \Phi$ of logical consequence between unambiguous formulas, and consider how we might extend it to ambiguous expressions e and e'. Clearly, if we can disambiguate e and e' to unambiguous formulas φ and ψ respectively, then it is reasonable to call e'

a logical consequence of e exactly when $\varphi \vdash \psi$. So suppose we have (— never mind, from where —) some disambiguating information θ that tells us, given any two expressions e and e' for which we wish to know if e entails e', that e is to be disambiguated as the formula $\theta_-(e, e') \in \Phi$, and e' as the formula $\theta_+(e, e') \in \Phi$. Then θ suggests an extension $\vdash_\theta$ of $\vdash$ to E, defined by

$$e \vdash_\theta e' \quad \text{iff} \quad \theta_-(e, e') \vdash \theta_+(e, e')$$

for all $e, e' \in E$. Now, rather than specifying a particular θ, suppose instead Θ were the set of all such θ's. Then it is reasonable to assert that e logically entails e' iff $\{\theta \in \Theta : e \vdash_\theta e'\}$ is a large subset of Θ. Insisting that Θ is the one "large subset of Θ" amounts to a *supervaluation* approach (van Fraasen [**12**]). But we might explore other possibilities, subject to minimal conditions such as

($*$) whenever A is a large subset of Θ, every subset B of Θ for which $A \subseteq B$ is also a large subset of Θ.

Then, writing $\mathcal{L}$ for the large subsets of Θ, it is natural to extend $\vdash$ to a relation $\vdash_\mathcal{L}$ on E, defined by

$$e \vdash_\mathcal{L} e' \quad \text{iff} \quad \{\theta \in \Theta : e \vdash_\theta e'\} \in \mathcal{L}$$

for all $e, e' \in E$. To guarantee that $\vdash_\mathcal{L}$ is the same as $\vdash$ on Φ, it suffices to require of every $\theta \in \Theta$ that for all $\varphi, \psi \in \Phi$

$$\theta_-(\varphi, \psi) \;=\; \varphi \;=\; \theta_+(\psi, \varphi) \;,$$

an eminently plausible assumption even if ψ is in $E - \Phi$. That assumption corresponds to clause (12) in Definition D. While the parameters Θ and $\mathcal{L}$ are assumed simply to be given, just as a is for Definition D, we may nonetheless investigate how these might relate to $(\cdot, \cdot, \cdot)_a$.

As a first attempt, why not take $\theta_-(e, e') = (\top, e, e')_a$ and $\theta_+(e, e') = (e, e', \top)_a$? The snag here is that whereas $\theta_-(e, e')$ and $\theta_+(e, e')$ are assumed to have unambiguous values, neither $(\top, e, e')_a$ nor $(e, e', \top)$ need be in Φ. (Indeed, the definition of $\vdash_\mathcal{L}$ above appeals not to just one θ, but to a collection Θ of these; it is with Θ that $(\cdot, \cdot, \cdot)_a$ should be related.) So suppose either $(\top, e, e')_a$ or $(e, e', \top)_a$ is not in Φ. We could try to append some expression l to the left of e such that $(l, e, e')_a \in \Phi$ and $(l; e,\ e',\ \top)_a \in \Phi$, and then set $\theta_-(e, e')$ to $(l, e, e')_a$ and $\theta_+(e, e')$ to $(l; e,\ e',\ \top)_a$. More generally, any number of θ's might be obtained by attaching expressions simultaneously to the left, right and in between e and e' so as to totally disambiguate e and e'. As for the family $\mathcal{L}$ of large subsets of Θ, that notion might be defined from the preference relations $\prec_a^{l,r}$ on E (mentioned in (†) and in the interpretation (11) of $\bullet$) as follows. Writing $\preceq_a^{l,r}$ for the reflexive closure of $\prec_a^{l,r}$, i.e.,

$$e \preceq_a^{l,r} e' \quad \text{iff} \quad e \prec_a^{l,r} e' \text{ or } e = e' \;,$$

let $\mathcal{L}$ consist of every subset A of Θ such that for every disambiguation, there is a $\preceq_a^{l,r}$-preferred one in A,

$$\begin{aligned} A \in \mathcal{L} \quad \text{iff} \quad & (\forall \theta \in \Theta)(\forall e, e' \in E)(\exists \theta' \in A) \\ & \theta'_-(e,e') \preceq_a^{\top,e'} \theta_-(e,e') \;\text{ and} \\ & \theta'_+(e,e') \preceq_a^{e,\top} \theta_+(e,e') \;. \end{aligned}$$

Observe that such an $\mathcal{L}$ verifies condition $(*)$ above, and that the conservative case $\mathcal{L} = \{\Theta\}$ of supervaluation arises from the absence of any preference (i.e., for all $l, r \in E$, $\prec_a^{l,r}$ is the empty relation $\emptyset$). Consequence relations $\mid\!\sim$ involving a preference order are familiar in the literature (e.g., Hobbs, Stickel, Appelt and Martin [**16**] and Beaver [**5**]), a wide variety of which (considered in Kraus, Lehmann and Magidor [**20**]) can be characterized by introducing an ambiguous expression c subject to a similar scheme $\mathcal{L}$ of preferential contextual disambiguation, with

$$\varphi \mid\!\sim \psi \quad \text{iff} \quad \varphi; c \vdash_{\mathcal{L}} \psi$$

for all unambiguous formulas φ, ψ. A precise statement and proof are given in Fernando [**11**], although much more work remains to be done on extending $\mid\!\sim$ to ambiguous expressions.[7] Suffice it, for now, to say that the rogue expression c represents an indeterminate leap in context (implicated, for example, in presupposition) that can serve as a fix, loosening up the tight constraint imposed on ; by (9).

The passage from a disambiguation strategy and an unambiguous consequence relation $\vdash$ to some ambiguous consequence relation $\vdash_{\mathcal{L}}$ suggests a *logical* notion of monotonicity (distinct from the aforementioned *operational* notions of left and right monotonicity), requiring that for all $e, e' \in E$,

$$\begin{aligned} e \vdash_{\mathcal{L}} e' \quad \text{implies} \quad & (\forall l \in E) \;\; l; e \vdash_{\mathcal{L}} e' \\ & \text{and/or} \;\; (\forall r \in E) \;\; e; r \vdash_{\mathcal{L}} e' \;. \end{aligned}$$

Personally, I find the previous operational notions of monotonicity more plausible requirements than a logical one, and see no reason (at the moment) why the slogans "monotonic interpretation" (from Alshawi and Crouch [**2**]) and "disambiguation as defeasible reasoning" (from Poesio [**23**]) must be irreconcilable. It is difficult to dispute the non-monotonicity in our everyday reasoning with ambiguous expressions. But, as a design principle, it is sensible to construe the *un*ambiguous character of formulas in Φ as the requirement that whenever $(l, e, r)_a$ *does* happen to be in Φ, it can*not* be re-interpreted, whatever further contextual information to its left or right there might be;

[7] An essential difference, for instance, is that even the principle (LLE) of left logical equivalence in Kraus, Lehmann and Magidor [**20**] becomes questionable under a formalization in which disambiguation need not respect truth-conditional equivalence (and the assumption that $[\![\varphi]\!]_a = [\![\varphi]\!]$ for unambiguous formulas φ must be weakened). On the other hand, lest we think that logic is completely irrelevant to disambiguation, it is perhaps worth citing the Strongest Meaning Hypothesis of Dalrymple, Kanazawa, Mchombo and Peters [**6**] as an important thesis linking logical entailment to disambiguation.

otherwise, such an $(l,e,r)_a$ should have been set to an element of $E-\Phi$. The flaw in this argument is that it may have the absurd(?) consequence that $(l,e,r)_a \in \Phi$ only when $e \in \Phi$ (the presumption being that ambiguity can never be resolved unambiguously inasmuch as any disambiguation can be reversed with further discourse); however, matters need not get this far, as long as we come to terms with the idea that any fixed disambiguation strategy a is bound to break down on some discourse (contrived pathologically perhaps for that very purpose). In any case, my hope is that the division between operational and logical approaches can serve as a tool for sorting out conflicting intuitions. And while I have no opinion as to how questions of priority and equivalence between operational and logical conceptions should intuitively be resolved, the distinction between them can be characterized quite crisply, I think, in very plain terms. Those terms have to do with how the partiality of disambiguation is formulated — either by delaying disambiguation, as in (11), or (following the time-honored practice of modelling a partial object by the set of its completions) by passing to some set Θ of total disambiguations (not to mention, some notion $\mathcal{L}$ of large subset of Θ).

4. Conclusion

The present paper proceeds from the premiss that ambiguous expressions require disambiguation for an unambiguous interpretation, but that ultimately the essence of ambiguity lies in the variety of notions of disambiguation from which to choose. To analyze that variety, two recipes are provided above, the first of which (in §2) reformulates a partial disambiguation strategy a into a ternary disambiguation function $(\cdot,\cdot,\cdot)_a$ that

(i) supports an abstract compositional interpretation of expressions on which $(\cdot,\cdot,\cdot)_a$ is defined,

(ii) provides a natural means in which to state operational constraints on a,

but

(iii) need not have anything to do with models or truth (on which a non-representational semantics is based).

Towards a non-representational semantics for ambiguous expressions, the second recipe extends (in §3) an unambiguous consequence relation $\vdash$ to ambiguous expressions, assuming certain ingredients Θ and $\mathcal{L}$. The problem arises as to how to transform the ingredients for one recipe into ingredients for the other.[8] Indeed, precisely what ingredients should be used in either recipe is an open matter left for experimentation, the hope being that "good" disambiguation strategies can be identified by studying what these recipes yield separately and, via some transformation between ingredients, jointly. Admittedly, the present work is very programmatic. In its defense, its "definition to

[8] We have only touched on one direction above — i.e., from a to $\Theta, \mathcal{L}$ — but clearly the opposite direction cries out for just as much attention. Such labor will presumably bear fruit when suitable complementary transformations are composed.

theorem" ratio is not quite as bad as it may appear, assuming one takes into account related theorems from Fernando [**10, 9, 11**], which have not been properly stated above (for the sake of readability).

Finally, a few closing words about a question, some answer to which the patient reader may well demand.

Why might the constructs $\bullet$, ; and $x =?$ (analyzed in Definition D) be so special?

From the point of view of logical semantics, it would be natural to investigate a calculus of sequents validating

$$e_1, \ldots, e_n \vdash e'_1, \ldots, e'_m \quad \text{iff} \quad e_1; \ldots; e_n \vdash e'_1 \bullet \cdots \bullet e'_m$$

(concerning which recall, in §3, (i) the appeal to the relations $\prec_a^{l,r}$ interpreting $\bullet$, for a notion $\mathcal{L}$ inducing $\vdash_{\mathcal{L}}$, and (ii) the use of ; to inject a rogue context c into the antecedent). As for the DRT expression $x =?$, it is arguably the ultimate underspecification construct, assuming the claim that ambiguity can be characterized uniformly in terms of unresolved variables. Within computational linguistics, this claim has been tested extensively by work on (revised) *Quasi Logical Form* (QLF, Alshawi and Crouch [**2**]), and related formalisms (van Genabith and Crouch [**13**], Pinkal [**22**]). All that is offered here is a semantics and formal setting, within which to state and study discourse effects on disambiguation.

Acknowledgments

My thanks to Nicholas Asher, Hans Kamp, Peter Krause and Carl Vogel for helpful discussions.

References

[1] Peter Aczel. *Non-Well-Founded Sets.* CSLI Lecture Notes Number 14, Stanford, 1988.

[2] H. Alshawi and R. Crouch. Monotonic semantic interpretation. In *Proc. 30th Annual Meeting of the Association for Computational Linguistics*, 1992.

[3] N. Asher and T. Fernando. Labeling representations for effective disambiguation. To be presented in the Second International Workshop on Computational Semantics (IWCS II). Tilburg, The Netherlands, January 1997.

[4] Nicholas Asher. *Reference to Abstract Objects in Discourse.* Kluwer Academic Publishers, Dordrecht, 1993.

[5] David Ian Beaver. Presupposition and assertion in dynamic semantics. Dissertation, University of Edinburgh, 1995.

[6] M. Dalrymple, M. Kanazawa, S. Mchombo, and S. Peters. What do reciprocals mean? In *Proc. Fourth Annual Semantics and Linguistic Theory Conference.* Cornell University, May 1994.

[7] Kees van Deemter. Towards a logic of ambiguous expressions. In K. van Deemter and S. Peters, editors, *Semantic Ambiguity and Underspecification.* CSLI Lecture Notes Number 55, Stanford, 1996.

[8] Solomon Feferman. A language and axioms for explicit mathematics. In J.N. Crossley, editor, *Algebra and Logic*, LNM 450. Springer-Verlag, Berlin, 1975.

[9] Tim Fernando. A logical connective for ambiguity requiring disambiguation. In J. Groenendijk, editor, *Ellipsis, Underspecification, Events and More in Dynamic Semantics.* Dyana deliverable R2.2.B, 1995.

[10] Tim Fernando. Ambiguity under changing contexts. Manuscript, 1996. Preliminary version presented in *Mathematics of Language 4* (Philadelphia, October 1995), and submitted for publication.

[11] Tim Fernando. Non-monotonic consequences of preferential contextual disambiguation. In P. Dekker and M. Stokhof, editors, *Proc. Tenth Amsterdam Colloquium.* ILLC, University of Amsterdam, 1996.

[12] B.C. van Fraassen. Presupposition, implication and self-reference. *Journal of Philosophy*, 65, 1968.

[13] J. van Genabith and R. Crouch. Direct and underspecified interpretations of LFG f-structures. In *COLING-96.* Copenhagen, 1996.

[14] J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14, 1991.

[15] Irene Heim. The semantics of definite and indefinite noun phrases. Dissertation, University of Massachusetts, Amherst, 1982.

[16] J. Hobbs, M. Stickel, D. Appelt, and P. Martin. Interpretation as abduction. *Artificial Intelligence*, 63, 1993.

[17] J. Jaspars and M. Kameyama. Preferences in dynamic semantics. In P. Dekker and M. Stokhof, editors, *Proc. Tenth Amsterdam Colloquium.* ILLC, University of Amsterdam, 1996.

[18] H. Kamp and U. Reyle. *From Discourse to Logic.* Kluwer Academic Publishers, Dordrecht, 1993.

[19] E. König and U. Reyle. A general reasoning scheme for underspecified representations. Preprint, 1996.

[20] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44, 1990.

[21] Reinhard Muskens. Combining Montague semantics and discourse representation. *Linguistics and Philosophy*, 19(2), 1996.

[22] Manfred Pinkal. Radical underspecification. In P. Dekker and M. Stokhof, editors, *Proc. Tenth Amsterdam Colloquium.* ILLC, University of Amsterdam, 1996.

[23] Massimo Poesio. Disambiguation as (defeasible) reasoning with underspecified representations. In P. Dekker and M. Stokhof, editors, *Proc. Tenth Amsterdam Colloquium.* ILLC, University of Amsterdam, 1996.

[24] Rob A. van der Sandt. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9(4), 1992.

9

Labeling representations for effective disambiguation

Nicholas Asher and Tim Fernando

Abstract. The effects of discourse on disambiguating (sub)expres-sions and the problem of computing those effects are treated formally by imposing a labeling structure over discourse representation structures, leading to barebones forms of Segmented and Underspecified Discourse Representation Theory.

Keywords: Ambiguity, underspecified semantic representations, information packaging, the semantics-pragmatics interface, dynamic interpretation in text

1. Introduction

From a formal semantic perspective, interpreting natural language utterances can be broken down into two tasks:

(I) Constructing logical forms or semantic representations of the utterances and

(II) Manipulating well-formed semantic representations.

Assuming semantic representations are drawn from a formal language, the problem (I) of *formalization* must be addressed before reasoning (II) *within* a formal system can commence. In *Discourse Representation Theory* (DRT, Kamp and Reyle [**10**]), the point is simply that a *Discourse Representation Structure* (DRS) must be associated with a natural language statement before one can speak of model-theoretic interpretations that lie behind the manipulations in (II), and which, in turn, provide the reason for bothering with (I).

Experience has shown that picking out a formal semantic representation faithful to a natural language statement is extremely difficult, resulting in pressure to blur the distinction between (I) and (II). Hence, the interest in *ambiguous* and *underspecified* semantic representations, which reduce the distance between natural language utterances and semantic representations. The price for shifting some of the burden of formalization (I) over to (II) is a gap within the formal system between on, the one hand, ambiguous and underspecified representations, and, on the other hand, unambiguous and fully specified representations (required to make sense of ambiguity and underspecification). The present paper is an attempt to combine two extensions of DRT, *Underspecified Discourse Representation Theory* (UDRT, Reyle [**13**]) and *Segmented Discourse Representation Theory* (SDRT, Asher [**1**], Lascarides and

Asher [**11**]), that concentrate on two complementary aspects of this gap: the first, on representing ambiguity; the second, on doing away with it. An example of an ambiguous expression excluded from Kamp and Reyle [**10**] is $x =?$, described in Asher [**1**] (following DRT folklore) as "an incomplete condition" that "anaphora resolution will make complete" (p. 73).[1] It is to complete such conditions (by supporting anaphora resolution and various other processes involved in discourse interpretation) that a system of *discourse relations* and *segmented discourse representation structures* (SDRSs) is introduced in SDRT. As shown in Asher [**2**], this extension can be presented perspicuously through a labeling system, similar to that used in UDRT for scope ambiguities. The present paper extends the use of labels to different levels (viz., word, sentence, discourse), with the intuition that the colon : between the label l and the expression e in $l : e$ serves as a "fence" for keeping the uncomputable content of e from spoiling the computational demands on disambiguation. Exactly what we mean by this will occupy much of the paper. At the end, we sketch some ideas on how to deal with ambiguity with a mix of "wait and see" and non-monotonic strategies.

2. From truth conditions to information packaging

To motivate our approach, we begin by recalling a formal distinction drawn in DRT between *conditions* and *discourse representation structures* (DRSs). A condition φ can be thought of as an ordinary "static" (truth-conditional) formula, interpreted (semantically), relative to a fixed set *MEP* of *model-embedding pairs*, as the set $[\![\varphi]\!] \subseteq \mathit{MEP}$ of model-embedding pairs at which it is *true* (à la Tarski, where an embedding assigns variables objects in the model paired with it):

$$[\![\varphi]\!] \;=\; \{s \in \mathit{MEP} \mid \varphi \text{ is true at } s\}\ .$$

By contrast, a DRS is a pair (U, C) consisting of finite sets U and C of discourse referents (or variables) and of conditions respectively, which is "dynamic" insofar as its meaning is taken to be a binary relation $[\![U, C]\!] \subseteq \mathit{MEP} \times \mathit{MEP}$ on model-embedding pairs, specifying its *input/output* behavior:

$$\begin{aligned}[\![U, C]\!] \;=\; \{(s, s') \in \mathit{MEP} \times \mathit{MEP} \mid \text{on input } s,\\ (U, C) \text{ can output } s'\}\ .\end{aligned}$$

As the set C of conditions in a DRS (U, C) can be conjoined[2] into a single condition φ_C so that $[\![U, C]\!] = [\![U, \{\varphi_C\}]\!]$, and as $[\![\emptyset, \{\varphi_C\}]\!] = \{(s, s) \mid s \in [\![\varphi_C]\!]\}$, the difference between conditions and DRSs is that a DRS may have a non-empty set of discourse referents (i.e., U need not be $\emptyset$).

[1] The string $x =?$ should *not* be confused with the perfectly unambiguous random assignment $x :=?$, employed in Groenendijk and Stokhof [**7**] to assign an indeterminate value to a variable introduced by an indefinite, or with the corresponding DRS $(\{x\}, \emptyset)$.

[2] If necessary (following the strict conventions of DRT, according to which logical connectives are defined on DRSs) by first turning a condition φ into a DRS $(\emptyset, \{\varphi\})$.

The set U of discourse referents in a DRS (U, C) makes a genuine difference in discourse interpretation inasmuch as it provides some information concerning what the discourse (represented by (U, C)) is *about*. In particular, U supplies possible referents for anaphora resolution. Clearly, however, U need not encode sufficient information either to make a choice, in cases such as (1) where more than one candidate presents itself, or to construct the required referent, in say bridging examples like (2).

(1) Ernest has a brother. He feeds him.

(2) Ernest has a bike. The seat is bent out of shape.

One might turn to the set C of conditions for help — but again without any guarantee of finding it. At least two problems plague a truth-conditional approach to discourse interpretation:

(P1) the prohibitive costs of extracting truth-conditional information

and

(P2) the coarseness of grain of what can be extracted truth-conditionally.

By (P1), we have in mind, for instance, the undecidability of first-order logic, rendering consistency checks hopeless (non-r.e.). As for (P2), the difficulty is that the information typically encoded into C is, if abstracted from all matters but truth, not quite fine enough, even when supplemented with U; an example is anaphora resolution in (3), for which let us agree that `having a brother` is reduced to the condition of being in the domain of a symmetric binary relation.

(3) Ernest has a brother. He also has a pigeon.

To overcome obstacles (P1) and (P2), we will magnify, rather than minimize, the distinction between DRSs and conditions, by enriching the set U of discourse referents in a DRS (U, C) with more structure. Insofar as truth occupies a privileged place in semantics, it makes sense to dedicate C solely to truth conditions, consigning to our enrichment of U other matters, which may nonetheless bear on C, through a subsequent process of discourse interpretation. In particular, the aforementioned expressions $x =?$ (which are not among the conditions or DRSs assigned an unambiguous semantics via $[\![\cdot]\!]$) can be collected to pick out a set $U_? \subseteq U$ of discourse referents x requiring anaphora resolution; and the set U can then be expanded into the structure $\langle U, U_? \rangle$. Of course, in practice, it may be convenient to write $x =?$ among the other conditions in C (in the lower portion of the box). But even then, we ought not to forget that quite unlike conditions in C, the expression $x =?$ is merely a signal to some anaphora resolution module for attention, to be discarded as soon as it has spawned a genuine (if not true) condition. But as we have already mentioned, anaphora resolution may require much more than just U and C, suggesting that the structure $\langle U, U_? \rangle$ be enriched further. Without spelling out just now what that is, let us write P for that enrichment, the idea being that P describes information *p*ackaging, to complement the information *c*ontent kept in C.

Having passed from the DRS (U, C) to the pair (P, C), let us next relate the information packaging/content dichotomy to ambiguity. If any consensus has developed at all about formalizing ambiguity, it is the necessity of locating the uncertainty characteristic of ambiguity at a meta-level, removed from the object level where unambiguous expressions lie. Inasmuch as the conditions in C are unambiguous, it is natural to equate that meta-level with P, and to think of $x =?$ as an ambiguous expression subject to a form of disambiguation called anaphora resolution. The meta/object-level division between P and C is complex and involves rules not only for

(a) discriminating between information packaging (P) and information content (C),

but also for

(b) communication between P and C.

The motivation for item (a) comes from the problem (P2) of coarseness of grain in C, and also from the costs, (P1), of retrieving information from C, which is, in addition, crucial for item (b). The idea, put in operational terms, is that the process of discourse interpretation should have immediate unrestricted two-way access to P, but not to C, for which it must, in view of (P1), submit itself to a rigorous border control regime. By "two-way access to P", we mean that the discourse interpretation process should not only be able to feed information into P (thereby updating it, as well as C), but also request information from P. But then if the discourse interpretation process is to be, in some sense, "effectively computable," then so too must the logic for P. Hence, another reason for separating out P from C is that it would be wishful thinking to expect the logic for C to be effectively computable. By contrast, insofar as information packaging must necessarily be, in some sense, finite (if only because of the infeasibility of unpacking something infinite, let alone the problem of storing an infinite amount of packaging material), it is more plausible to assume that the logic for P would be more tractable. What makes the entire business rather delicate, however, is that it is not at all clear that discourse interpretation can get by without occasionally requesting information from C. Consider, for instance, the connection argued in van der Sandt [**14**] between anaphora resolution and presupposition projection. In effect, van der Sandt [**14**] supplements quasi-conditions like $x =?$ with restrictions of the kind in Barwise and Cooper [**4**] that must somehow be accomodated within (the representation that defines) C. This poses considerable computational challenges through its unrestricted use of a logic for C (including consistency checks). As an initial step to addressing this problem (something one of us has done elsewhere), our notational system exposes it. More specifically, our proposal for item (b) amounts to a labeling system, similar to that used in Reyle [**13**], with the crucial additional idea of employing the colon : in $l : e$ as a fence separating the label l (belonging to P) from the expression e (belonging to C) which (in contrast to l) is potentially arbitrarily complex. As the expressions e in Reyle [**13**] are treated only as pieces of syntactic material, our use of : is essentially different. On the other hand, it does share with Reyle [**13**] the aim of dealing with ambiguity in a computationally effective manner — an aim that requires integrating different levels of structure.

3. Between discourse macro-structure and micro-structure

It is one thing to isolate an instance of ambiguity within a word (as in the case of lexical ambiguity) or say, a sentence (as in the case of quantifier scope ambiguity); it is quite another matter to resolve that ambiguity. The former suggests that the problem is a local one; whereas experience with the latter teaches us that the solution often involves a more global consideration of the discourse. Consider the discourse (4) below, from Asher and Lascarides [**3**], where the intervening sentence (4b) makes all the difference in disambiguating *bar*.

(4)
a. The judge asked where the defendant was.
b. The barrister apologised, and said he was at the pub across the street.
c. The court bailiff found him slumped underneath the <u>bar</u>.

To account for (4), we must recognize that a discourse is, as argued in Grosz and Sidner [**8**], composed of *segments*, which means we must determine

(T1) what those boundaries are

and

(T2) how the segments are joined.

In SDRT, the tasks (T1) and (T2) are intertwined within a hierarchical structure of *segmented DRSs*, defined inductively from DRSs through *discourse relations* modeled after Hobbs [**9**], Polanyi [**12**] and Thompson and Mann [**15**]. Under the perspective within which we have been proceeding, taking DRSs as atomic forms of discourse segments is perhaps natural enough — assuming that these atomic forms can, in fact, be decomposed at a lower (e.g., intra-sentential) level. What may require more motivation is the choice of discourse relations that join segments so as to form larger segments. Here we will steer clear of this controversy, and concentrate (in a neutral vein) on more abstract matters.

The proposal that there is some structure to a multi-sentence discourse that must, if the discourse is to be interpreted, be uncovered (along the lines suggested by (T1) and (T2)) is essentially no different from what is by now the widely held view that some form of syntactic analysis of a sentence is a prerequisite to interpreting that sentence. The only difference is the step up in scale from a sentence to a piece of discourse, plus the possibility that the larger discourse structure may not only presuppose structure at the lower (intra-sentential) level, but, in fact, determine what that structure is. Given this two-way interaction between macro- and micro-levels, one might be tempted to do away altogether with the separation into levels. But just as we have (in the previous section) been at pains to maintain a distinction between P and

C (even though discourse interpretation may require both), let us insist on differentiating between macro- and micro-levels (noting parenthetically that the division between macro- and micro-levels is orthogonal to that between object- and meta-levels, underlying C and P, respectively). The next three sections flesh out the preceding ideas in somewhat precise terms.

4. Labeled representations and links (without holes)

To pick out a specific occurrence of an expression e, one might, as in Reyle [**13**], introduce a label l and, applying it to that occurrence, write $l : e$. In order to carry this out for an occurrence of e within a larger expression, it is useful to consider how expressions are generated. An expression, such as the formula $(\forall x)(x = y \ \vee \ f(x) < z)$ from predicate logic, is the product of *constructors*, such as $\forall$, x, $=$, y, $\vee$, f, $<$ and z, each of which is assigned an *arity* indicating the number of arguments it takes. For example, x, y and z each have arity 0 (taking, as they do, no argument), while $\forall$ has arity 2 (one for a variable, and another for a formula), as do $=$, $\vee$ and $<$ (though these require different *sorts* of arguments, which we will, for the sake of simplicity, ignore[3]). Now, to accomodate labels l within an expression e, we can simply increase the arity of each constructor by 1, using that additional argument place for a label (which can remain implicit in case that occurrence is of no interest). In UDRT and in Bos [**5**], the other ("genuine") arguments of a constructor are often left as "holes" representing scope ambiguity, the idea being that disambiguation consists, as Bos [**5**] puts it, of "plugging" these holes with labels. We will attend to holes in the next section, but keep things in this section whole in order to describe another application of labeling, for which holes need not enter (at least not immediately).

The application we have in mind is discourse macro-structure, of the kind described in the preceding section. Without spelling out what exactly discourse segments and discourse relations are, suffice it to say that a discourse relation is a constructor that takes n discourse segments as arguments. For instance, in Asher [**1**], $n = 2$ and

(i) a discourse segment is either a DRS or a *segmented DRS* (SDRS), which, like a DRS, is a pair $(\mathcal{U}, \mathcal{C})$ of sets consisting of a finite set $\mathcal{U}$ of discourse segments, and a finite set $\mathcal{C}$ of SDRS conditions on $\mathcal{U}$, where

(ii) an *SDRS condition on* $\mathcal{U}$ is an expression obtained by applying a discourse relation to discourse segments from $\mathcal{U}$.

We will exploit the parallel between an SDRS and discourse micro-structure given by a DRS (U, C) with a view towards effective disambiguation (unhindered, to the extent possible, by the problems (P1) and (P2) noted in §2). The step from (U, C) to $(\mathcal{U}, \mathcal{C})$ will be reformulated as one from (U, C) to (P, C), where P represents information packaging (that can come at either the macro-

[3]A sorted version would take an arity to be a sequence of sorts indicating the sort required at each argument place. (E.g., $\forall$ has arity (*variable,formula*).) This complicates the notation, but can otherwise be incorporated in a straightforward way to what follows.

or micro-level). Towards that end, observe that a discourse relation R holds between discourse segments $x_1, \ldots, x_n$ *relative to some context* y, which itself constitutes a discourse segment.[4] That is, a discourse relation R is more properly conceived as having arity not n, but rather $n+1$, as it would under a labeling system, where $R(x_1, \ldots, x_n)$ is transformed to $y : R(x_1, \ldots, x_n)$. To bring this observation out into the notation, it is natural to define discourse relations not on SDRSs directly, but rather on instances, or labels, of them as in Asher [**2**].[5] Hence, the elements in the set $\mathcal{U}$ forming an SDRS $(\mathcal{U}, \mathcal{C})$ can be regarded simply (and more generally) as labels.

To capture the labeling structure shared by both macro- and micro-levels of discourse structure, let us formulate the notion of labeled representation as a relational structure (in the sense of predicate logic). The idea is, given some set of constructors, to associate with a constructor f of arity n an $(n+1)$-ary relation symbol R_f, so that for a constructor e with arity 0,

$$l : e \quad \text{becomes} \quad R_e(l)$$

and a constructor f of arity 2,

$$l : f(e, e') \quad \text{becomes} \quad R_f(l_e, l_{e'}, l)$$

where l_e labels the occurrence of e in $l : f(e, e')$ and $l_{e'}$ does the same for e'. For the record,

Definition. Given a set Σ of constructors f with specified arities $n_f \in \{0, 1, \ldots\}$, a *labeled* Σ*-representation* is a pair $\langle U, I \rangle$ where

(L1) U is a non-empty set

(L2) I is an (interpretation) function with domain Σ, such that for every $f \in \Sigma$ of arity n, $I(f)$ is an $(n+1)$-ary relation on U (i.e., $I(f) \subseteq U^{n+1}$)

and (take a deep breath)

(L3) the binary relation *Succ* on U is well-founded, where (by definition) *Succ* consists of exactly the pairs $(l, l') \in U \times U$ such that for some $f \in \Sigma$ of positive arity $n > 0$, and some $(n-1)$-tuple $l_1, \ldots, l_{n-1}$ from

---

[4] More concretely, for instance, the sentence *John pushed him* can be said to "explain" *Max fell* within the discourse (5), analyzed in Lascarides and Asher [**11**], but (presumably) not within (6).

(5) Max fell. John pushed him.

(6) Max fell. He got up. John pushed him.

[5] In Asher [**1**] and other previous work, the contextual nature of the semantics of discourse relations was captured by the fact that each DRS generated as part of an SDRS had a distinct set of discourse referents. Accordingly, it was perfectly consistent to maintain that the relation *Explanation* held between the two DRSs in the SDRS for (5) in footnote 5 but not between the two (different) DRSs generated by the same sentences in (6). The label notation is an improvement in that it allows one to ignore details about the DRS construction procedure.

U,

$$(l, l_1, \ldots, l_{n-1}, l') \in I(f) \ \text{ or } \ (l_1, l, l_2 \ldots, l_{n-1}, l') \in I(f)$$
$$\text{or} \ \cdots \ \text{or} \ (l_1, \ldots, l_{n-1}, l, l') \in I(f) \ .$$

In item (L3), the binary relation *Succ* is what is sometimes referred to in the literature as a "successor" or immediate dominance relation, the transitive closure of which is subordination. To see why it should be well-founded, let us define the notion that an element $l \in U$ *labels* a Σ-expression e (built recursively from the constructors in Σ) by induction on e as follows.

> If e is an atomic expression — i.e., a constructor in Σ of arity 0 — then l labels e iff $l \in I(e)$. For the inductive step, l labels $f(e_1, \ldots, e_n)$ iff $(l_1, \ldots, l_n, l) \in I(f)$ for some $l_1, \ldots, l_n \in U$ such that l_1 labels e_1 and ... and l_n labels e_n.

Returning to (L3), observe that *Succ* traces the inductive generation of an expression, and must thus be well-founded, provided that

(a) for every $l \in U$, l cannot be re-used as a label within the same occurrence of an expression

and

(b) for some fixed number i, every $l \in U$ can label at most i occurrences of expressions.[6]

Part (a) is implied already by (L3), whereas (b) will follow from some finiteness assumptions that will be imposed in §6. For now, let us isolate a notion that will prove useful for relating labeled Σ-representations induced by discourse micro- and macro-structure, respectively. Given a labeled Σ-representation $\langle U, I\rangle$, and the binary relation *Succ* defined on U as in (L3) above, call an element $l \in U$ *atomic* if there is *no* $l' \in U$ for which $Succ(l', l)$. That is, $l \in U$ is atomic precisely if l is minimal relative to subordination.

Next, let us associate labeled Σ-representations with discourse micro- and macro-structure by choosing $\Sigma_\circ$ and Σ° respectively as follows.

> **Micro-structure$_\circ$.** Let $\Sigma_\circ$ be a set of constructors, yielding (as expressions) discourse referents and conditions. More precisely, the constructors in $\Sigma_\circ$ of arity 0 are discourse referents. Constructors in $\Sigma_\circ$ of arity > 0 return conditions, when applied to either conditions or discourse referents. That is, constructors of arity > 0 include not only atomic relation symbols, but also logical connectives, quantifiers, etc.[7]

[6] To the extent that a label merely signifies an occurrence, it is natural to require that a label apply to at most one occurrence (i.e., $i = 1$). Recall, however, that for applications to discourse macro-structure, a label also serves as a "context," relative to which a discourse relation joins certain discourse segments.

[7] Recall that we have been ignoring sorting of expressions to simplify the notation. There is also the complication that in DRT, the logical connectives and quantifiers operate not on conditions but on DRSs. It is easy enough, however, to get the constructors in the form mentioned above, by defining, for example, the operations on conditions φ according to the

> **Macro-structure$^\circ$.** Constructors in Σ° of arity 0 are discourse segments; constructors of arity > 0 are discourse relations.

Beyond the fact that discourse segments have arity 0, and discourse relations have certain positive arities, the precise nature of discourse segments and discourse relations can, for the purpose of inducing a notion of labeled Σ°-representation, be left open. The force or "semantic content" of these (syntactic) constructors can be brought out elsewhere, through, for example, the following link with micro-structure. Given

> a labeled $\Sigma_\circ$-representation $\langle U_\circ, I_\circ \rangle$ (for micro-structure), the atomic elements of which form the set $A_\circ$

and

> a labeled Σ°-representation $\langle U^\circ, I^\circ \rangle$ (for macro-structure), the atomic elements of which form the set A°,

let $\in_U$ and $\in_C$ be two relations $\in_U \subseteq A_\circ \times A^\circ$ and $\in_C \subseteq (U_\circ - A_\circ) \times A^\circ$, supporting the intuition that for every $u \in A_\circ$ and $k \in A^\circ$,

$$u \in_U k \quad \text{iff} \quad u \text{ labels a discourse referent in the (universe of the) DRS labeled by } k$$

and for every $c \in U_\circ - A_\circ$,

$$c \in_U k \quad \text{iff} \quad c \text{ labels a condition in the DRS labeled by } k \ .$$

That is, $\in_U$ and $\in_C$ associate with every atomic element $k \in A^\circ$ (of macro-structure) the DRS (U_k, C_k), where

$$\begin{aligned} U_k &= \{ f \in \Sigma_\circ \ : \ (\exists u \in A_\circ)\ u \in_U k \text{ and } u \in I_\circ(f) \} \\ C_k &= \{ \varphi \ : \ (\exists c \in U_\circ)\ c \in_C k \text{ and } c \text{ labels } \varphi \\ & \qquad \text{(relative to } \langle U_\circ, I_\circ \rangle) \} \ . \end{aligned}$$

Clearly, a relation $\in_U$ can be hardwired as a relation between constructors in $\Sigma_\circ$ of arity 0 (i.e., discourse referents) and constructors in Σ° of arity 0 (i.e., discourse segments) by encoding a set of discourse referents into the notion of a discourse segment (as in the case of a DRS). On the other hand, the relation $\in_C$ can*not* be hardwired (in the same way) as a relation between constructors in $\Sigma_\circ$ of positive arity, and constructors in Σ° of arity 0, since a constructor in $\Sigma_\circ$ of positive arity does not constitute a (completed) condition, requiring, as it were, a labeled $\Sigma_\circ$-representation for completion.

Turning to the non-atomic elements of U°, an $l \in U^\circ - A^\circ$ has discourse macro-structure given by I°, and inherits the micro-structure of its atomic constituents (i.e., the atomic elements related to it by subordination). As described in §6, the discourse relations connecting discourse segments may impose certain constraints on the micro-structure of those discourse segments. Focussing for now on macro-structure, let us mention a simple "connectedness" constraint on $\langle U^\circ, I^\circ \rangle$ amounting to a very primitive notion of *coherence*;

effects of the DRT operations on the DRSs $(\emptyset, \{\varphi\})$; similarly, when discourse referents as well as conditions are involved.

namely, that there is a label $Top \in U^\circ$ such that for every $l \in U^\circ - \{Top\}$, (l, Top) is in the transitive closure of the binary relation *Succ* mentioned in (L3) above — i.e., every $l \in U^\circ - \{Top\}$ is subordinate to *Top*. Note that by (L3), there is at most one such label *Top*. The reader accustomed to drawing a DRS box, with one condition written on top of another, might flatten *Top* into such a box, where micro-structure is horizontal (in the sense that it describes a particular condition occupying a row in the box), and macro-structure is vertical (insofar as discourse relations connect conditions listed down the box).

Collecting together notions from this section, let us define a $\Sigma_\circ, \Sigma^\circ$-*packet* $\hat{P}$ to be a 5-tuple specifying a particular choice of

$$\langle U_\circ, I_\circ \rangle,\ \langle U^\circ, I^\circ \rangle,\ \in_U,\ \in_C,\ Top\quad .$$

We will often drop the parameters $\Sigma_\circ, \Sigma^\circ$ (when they can be ignored in the background), and speak simply of packets.

5. Discourse interpretation and underspecification

What, one might ask, have we gained by reformulating (in the previous section) the step from discourse micro-structure (U, C) to macro-structure $(\mathcal{U}, \mathcal{C})$ as one from (U, C) to $(\hat{P}, C)$? The process of discourse interpretation can, as in DRT, be conceived as an *updating of discourse contexts*, which, as every programmer knows, requires some coding or representation if the process is to be realized computationally. Now, the point behind spelling out a notion of a labeled representation is to lay down a solid basis for integrating SDRT and UDRT. But to actually carry out such an integration, we will need to bore holes into our labeled representations.

Let us explain. The notion of a *discourse context* given in Asher [**2**] specifies a particular packet $\hat{P}$, as well as (i) an element $Current \in U^\circ$ that labels the last discourse segment analyzed, and (ii) a certain function (we will get to in the next section) assigning discourse segments *main eventualities*. But confining our attention at present to packets $\hat{P}$, a crucial question to ponder is

(Q) should the initial discourse context in a discourse interpretation process, in fact, *single out a particular* packet $\hat{P}$, and, even if it did, should the output of a discourse interpretation process also specify exactly one packet?

Notice that we have smuggled into (Q) the presupposition of a definite output. Both the uniqueness and existence of an output might very well be challenged, under an approach to ambiguity based on non-determinism. Be that as it may, the widespread interest in underspecification suggests that the answer "no" to both parts of (Q) is worth exploring.

The idea behind underspecified representations is simply to leave things out. We have gone through great lengths to formulate labeled $\Sigma_\circ$-representations as relational structures so that, for instance, the first-order language of these relational structures can be used to describe families of labeled $\Sigma_\circ$-representations. Writing R_f for the $(n+1)$-ary relation symbol induced by a constructor f in $\Sigma_\circ$ of arity n, and adding a binary relation symbol $\leq$ for the reflexive-transitive closure of the relation *Succ* mentioned in (L3), the generalization of UDRT in Bos [**5**] is obtained by considering atomic formulas of the form $R_f(x_1, \ldots, x_n, l)$ and $x \leq y$, where holes are nothing more than variables (at this meta-level of packaging). Other forms of ambiguity (e.g., lexical) can be captured if the constructors can be replaced by variables, as in higher-order logic. SDRT invites similar investigations of labeled Σ°-representations, separate from or in conjunction with labeled $\Sigma_\circ$-representations via fixed or varying choices of links $\in_U$ and $\in_C$. Doubts (such as that expressed in Wilson [**16**]) that discourse relations of the sort used in SDRT must be identified in order to interpret a discourse suggest that perhaps R_f ought, at least in some cases, to be left underspecified (i.e., as a hole). An essential consideration that must be brought into the picture is, as Wilson [**16**] argues, "processing effort." It is precisely that issue which the next section tries to address, in formal terms.

6. Finitely packaging infinite content

Having been concerned, in §§4 and 5, almost exclusively with information packaging, let us now peek at the content inside. But as we will be interested in just how much effort that takes, let us define a labeled Σ-representation $\langle U, I\rangle$ to be a *finitely labeled* Σ*-representation* if, in addition to (L1), (L2), and (L3), it meets the following requirement (reflecting the finiteness of discourse)

(L4) U is finite, and $I(f) \neq \emptyset$ for at most finitely many f's in Σ.

Henceforth, all labeled Σ-representations (in particular, those in packets $\hat{P}$) will be assumed to be finitely labeled. It follows that a packet is a perfectly finite object that can readily be Gödel-numbered, and fed to a computer.

Writing Φ for the collection of conditions, let us assume also that we are given some notion $\vdash \subseteq \mathsf{Pow}(\Phi) \times \Phi$ of logical entailment on these conditions (by virtue of which the conditions can be said to have *content*[8]). Now, fix a packet $\hat{P}$. Recall from §4 that every atomic element k of the macro-structure $\langle U^\circ, I^\circ\rangle$ is assigned a set $C_k \subseteq \Phi$ of conditions. The key to the content encoded by a packet $\hat{P}$ is the relation $:_\vdash$ between such elements k and conditions φ, defined by

$$k :_\vdash \varphi \quad \text{iff} \quad C_k \vdash \varphi \ .$$

The relation $:_\vdash$ can be extended to arbitrary (non-atomic) elements in the macro-structure $\langle U^\circ, I^\circ\rangle$, including *Top*, for which

$$Top :_\vdash \varphi \quad \text{iff} \quad \Big(\bigcup_{k\in A^\circ} C_k\Big) \vdash \varphi \ .$$

[8] "Content" here can be understood in the model-theoretic terms mentioned in §2, assuming $\vdash$ is connected to that by a completeness theorem, as in first-order logic.

The computational costs of $:_{\vdash}$ can be traced to $\vdash$, for which the best we can hope for is that it be recursively enumerable (as in the case of first-order logic). It is for this reason that we have described the relation : earlier as a "fence" for keeping discourse interpretation computationally effective.

As we have also mentioned, however, the pressure to jump the fence can be irresistible when theorizing about discourse without regard for computation. To determine a range of temporal relations, Lascarides and Asher [**11**] make do with the trivial "inspection" notion $:_I$ of inference relation, given by

$$k :_I \varphi \quad \text{iff} \quad \varphi \in C_k \ ,$$

together with a function Ev picking out the *main eventuality* $Ev(k)$ associated with the label k. The idea, in general, is to enrich a packet with some finite structure $\mathcal{E}$ around the image of Ev, approximating the worlds picked out by $\{\varphi \in \Phi \mid Top :_{\vdash} \varphi\}$ (according to the model-theoretic interpretation of conditions mentioned in §2 above). An additional feature of SDRT is the use of a non-monotonic logic to determine discourse relations between discourse segments, and (for instance) the temporal relations between the eventualities. The non-monotonicity here reflects a human tendency to disambiguate discourse without necessarily incontrovertible evidence. But the computational cost of nonmonotonically inferring the appropriate SDRS is high even in the best of cases, involving only $:_I$ and quantification over the aforementioned finite structure, and hopeless in the more realistic case, where the use of $:_{\vdash}$ is less restricted.

In addition to making nonmonotonic leaps in discourse interpretation, however, humans also live with ambiguity. Indeed, part of the interest in underspecification is the possibility of filling in holes monotonically as the discourse progresses. To what degreee can the non-monotonicity in SDRT be reduced through underspecification? Without attempting to settle this question now once and for all, let us try to be more precise about what we mean by a discourse context. An *idealized discourse context* Ω is a tuple consisting of a (finite) packet $\hat{P}$, a label $Current \in U^{\circ}$ (pointing to the last discourse segment analyzed), a function Ev with domain U°, and possibly some additional finite structure built around the image of Ev.

To turn an idealized discourse context Ω into a *discourse context*, two ingredients are needed:

(a) a finite set $U_?$ of *pieces* of Ω (to be treated as holes)

and

(b) a set Γ of *constraints* on Ω (specifying requirements on how to fill the holes in $U_?$).

By "pieces" of Ω, we mean not only

(I) labels $l \in U_{\circ} \cup U^{\circ}$ in the packet specified by Ω

but also

(II) constructors $f \in \Sigma_{\circ} \cup \Sigma^{\circ}$ (buried) in the domain of $I_{\circ}$ and of I°.

Confining our attention first to micro-structure, scope ambiguities represented in UDRT and in Bos [**5**] can be captured by

(a) including among $U_?$ labels from $U_\circ$ (as allowed under (I)), and

(b) encoding into Γ the requirement that such holes be filled (in a bijective manner, if we wish) from some set of pieces of Ω disjoint from $U_?$, in accordance with certain subordination constraints (expressed through an auxiliary symbol $\leq$).[9]

In addition, cases of lexical and pronominal ambiguity can be represented by

(a) throwing into $U_?$ constructors from $\Sigma_\circ$ (as allowed under (II)),

although for these cases,

(b) it is not at all clear that Γ should require that such holes be plugged with material from Ω.

That is to say, a crucial difference between the structural ambiguities represented in UDRT and the lexical and pronominal ambiguities disambiguated in SDRT is that in the latter, specifying the material with which to plug the holes can be highly non-trivial (as illustrated by bridging phenomena). In particular, it is doubtful whether Γ should require that those holes be filled by pieces in Ω.

At the discourse macro-level, pieces in item (I) specify *where* to attach a discourse segment, while pieces in item (II) specify *what* discourse relation to link discourse segments with. In SDRT, Γ would constrain all labels in $U^\circ - \{Top\}$ to be subordinate to *Top*, and relate the main eventualities of discourse segments linked by particular discourse relations according to the rules set out, for instance, in Lascarides and Asher [**11**]. Now, by including into $U_?$ discourse macro-items from (I) and (II), we allow for the possibility of underspecifying the structure of discourse, according to considerations centering around "processing effort" (Wilson [**16**]). That is, suppose we have a discourse segment K (possibly underspecified) and a label l that is to be incorporated into some (initial) discourse context. One option — a "lazy" and trivially monotonic one — is to simply add the label l to $U_?$ and any other underspecified elements in K to $U_?$. Another option is to resolve some of the underspecification by using various bits of the SDRS update procedure of Asher [**1, 2**]. Which bits we want to resolve will depend, as relevance theorists have pointed out, on how much computational cost we think is associated with the resolution and the amount of information the interpreter hopes to gain from the computation. For instance, the presence of a clue

[9] The reader familiar with the *underspecified representations* $\langle H, L, C\rangle$ of Bos [**5**] should be able to extract such triples as special instances of the notion of a discourse context above, noting the formal (but otherwise inessential) difference that holes are treated (for the sake of convenience) as labels above.

By introducing discourse structure into the representations, traces of syntax of the kind left, for example, in the notion of a plugging (present in the revised semantics of URs in Definition 8 of Bos [**5**]) can be analyzed semantically, yielding a fully compositional interpretation of ambiguous expressions, relative to some (possibly partial) notions of disambiguation (Fernando [**6**]).

word or conditions in the look-up may make it almost effortless to compute the discourse relation through which to attach the new label l. On the other hand, although it may be quite expensive to compute the discourse relation, the resolution of other underspecifications — both lexical and pronominal — may depend on this computation (see Asher and Lascarides [**3**]). In this case too, it might seem appropriate for the interpreter to calculate the discourse relation. If, however, the other underspecified elements do not rely on a choice of discourse relation, the interpreter might leave the choice of R unresolved (if the computation of R is itself not trivial). Similarly, computing an attachment site might, in some cases, be very easy (especially if there is only one possible site), but, in other cases, quite expensive.

7. Conclusion

There is a striking logical contrast in disambiguation between the adventurous non-monotonicity in SDRT and the cautious stillness of UDRT. This is, no doubt, due in no small measure to the very different directions in which SDRT and UDRT have been pushed — SDRT towards the semantics/pragmatics interface, and UDRT towards computational work. We have tried above to outline a level of representations at which such a connection might be established. Marrying SDRT with UDRT has dramatic consequences for discourse interpretation. By allowing underspecification at all levels, we can reduce the requirements on contextual update, disambiguating only when we think the gains are worth the computational effort — a point we hope to take up elsewhere.

References

[1] Nicholas Asher. *Reference to Abstract Objects in Discourse*. Kluwer, Dordrecht, 1993.
[2] Nicholas Asher. Mathematical treatments of discourse contexts. In P. Dekker and M. Stokhof, editors, *Proc. Tenth Amsterdam Colloquium*. ILLC, University of Amsterdam, 1996.
[3] N. Asher and A. Lascarides. Lexical disambiguation in a discourse context. *J. Semantics*, 12, 1995.
[4] J. Barwise and R. Cooper. Extended Kamp notation: a graphical notation for situation theory. In P. Aczel, et al., editors, *Situation Theory and Its Applications*, volume 3. CSLI, Stanford, 1993.
[5] Johan Bos. Predicate logic unplugged. In P. Dekker and M. Stokhof, editors, *Proc. Tenth Amsterdam Colloquium*. ILLC, University of Amsterdam, 1996.
[6] Tim Fernando. Ambiguity under changing contexts. Manuscript, 1996. Preliminary version presented in *Mathematics of Language 4* (Philadelphia, October 1995), and submitted for publication.
[7] J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14, 1991.
[8] B. Grosz and C. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3), 1986.
[9] Jerry R. Hobbs. On the coherence and structure of discourse. Technical Report 85-37, Center for the Study of Language and Information, Stanford, 1985.
[10] H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer, Dordrecht, 1993.

[11] A. Lascarides and N. Asher. Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and Philosophy*, 16(5), 1993.

[12] Livia Polanyi. A theory of discourse structure and discourse coherence. In P.D. Kroeberger, et al., editors, *General Session, 21st Regional Meeting of the Chicago Linguistics Society*. Chicago, 1985.

[13] Uwe Reyle. Dealing with ambiguities by underspecification: construction, representation and deduction. *J. Semantics*, 10(2), 1993.

[14] Rob A. van der Sandt. Presupposition projection as anaphora resolution. *J. Semantics*, 9(4), 1992.

[15] S. Thompson and W. Mann. Rhetorical structure theory: a framework for the analysis of texts. In *IPRA Papers in Pragmatics*, volume 1. 1987.

[16] Deirdre Wilson. Truth, coherence, relevance: recent approaches to pragmatics. Invited talk, ITALLC 96, London, July 1996.

10

In conjunction with qualitative probability

Abstract. Numerical probabilities (associated with propositions) are eliminated in favor of qualitative notions, with an eye to isolating what it is about probabilities that is essential to judgments of acceptability. A basic choice point is whether the conjunction of two propositions, each (separately) acceptable, must be deemed acceptable. Concepts of acceptability closed under conjunction are analyzed within Keisler's weak logic for generalized quantifiers — or more specifically, filter quantifiers. In a different direction, the notion of a filter is generalized so as to allow sets with probability non-infinitesimally below 1 to be acceptable.

1 Introduction: weighing the evidence. Let $\mathcal{L}$ be a set of "formulas" $(\varphi, \psi, \dots)$, and V be a set of "possibilities" $(a, b, \dots)$, connected to $\mathcal{L}$ by a relation $\models$, capturing the intuition that for all $a \in V$ and $\varphi \in \mathcal{L}$,

$$a \models \varphi \quad \text{iff} \quad a \text{ "supports" } \varphi \ .$$

We can, for concreteness, take the case of predicate logic, although we can also proceed more abstractly, assuming only some relation $\models \subseteq V \times \mathcal{L}$. In any case, the point is to pick out a set $\mathcal{A} \subseteq \mathcal{L}$ of "*a*cceptable" formulas on the basis of $\models$, so that for every $\varphi \in \mathcal{L}$,

$$\varphi \text{ is } \textit{acceptable} \quad \text{iff} \quad \begin{array}{l}\text{the "evidence" } \{a \in V : a \models \varphi\} \\ \text{supporting } \varphi \text{ "has enough weight"}\end{array}$$

or, in other words,

$$\varphi \in \mathcal{A} \quad \text{iff} \quad \{a \in V : a \models \varphi\} \in \mathcal{H} \tag{1}$$

for some family $\mathcal{H} \subseteq \mathsf{Pow}(V)$ of "*h*eavy" subsets of V. Clearly, the safest choice is to equate $\mathcal{A}$ with the $\models$-validities (i.e., the formulas φ such that for every $a \in V$, $a \models \varphi$), which is to say, to set $\mathcal{H} = \{V\}$. But suppose we took a *chance* on some other choice of $\mathcal{A}$ (and $\mathcal{H}$) that tolerates failures or exceptions. What are the rules for doing so *rationally*?

"Logic" suggests the closure condition

$$\text{(Up)} \qquad \frac{\varphi \in \mathcal{A} \qquad \varphi \vdash \psi}{\psi \in \mathcal{A}} \qquad\qquad \frac{A \in \mathcal{H} \qquad A \subseteq B\ (\subseteq V)}{B \in \mathcal{H}}$$

where $\varphi \vdash \psi$ means that for every a, if $a \models \varphi$ then $a \models \psi$. (Up) alone introduces no element of risk into $\mathcal{A}$ (or $\mathcal{H}$) insofar as (Up) does not yield, from the assumption that $V \in \mathcal{H}$, any more elements of $\mathcal{H}$. On the other hand, (Up) does suggest how to proceed — namely, by weakening $\vdash$ to some binary relation $\mid\!\sim$ on $\mathcal{L}$ or (turning to $\mathcal{H}$) $\subseteq$ to some binary relation $\preceq$ on $\mathsf{Pow}(V)$

$$(\mathrm{Up})_{\mid\sim} \ \frac{\varphi \in \mathcal{A} \qquad \varphi \mid\!\sim \psi}{\psi \in \mathcal{A}} \qquad\qquad (\mathrm{Up})_{\preceq} \ \frac{A \in \mathcal{H} \qquad A \preceq B}{B \in \mathcal{H}} \ .$$

Although the correspondence between $\mathcal{A}$ and $\mathcal{H}$ described by (1) extends to pairs $\mathcal{A}$ and $\mathcal{H}$ closed respectively under $(\mathrm{Up})_{\mid\sim}$ and $(\mathrm{Up})_{\preceq}$, for some relations $\mid\!\sim$ and $\preceq$, the intuitions that most naturally come to play with the rule $(\mathrm{Up})_{\mid\sim}$ differ from those behind $(\mathrm{Up})_{\preceq}$ — the former being syntactic (or proof-theoretic), the latter, semantic. An example where this distinction matters concerns the operation $\wedge$ of conjunction on $\mathcal{L}$, for which it is understood that for every $a \in V$, $a \models \varphi \wedge \psi$ iff $a \models \varphi$ and $a \models \psi$. It is largely on this simple example that the present paper turns.

1.1 Content of paper. There is a certain plausibility to asserting

(And) if φ and ψ are both acceptable, then $\varphi \wedge \psi$ is acceptable

if only because

(i) it takes a bit of sophistication to even sense the difference between "φ and ψ" and "$\varphi \wedge \psi$"

and

(ii) after such sophistication is acquired, we learn that the difference does not (in a sense) matter, if acceptability is construed as validity (semantic or syntactic).

(And) becomes problematic, however, as soon as we accept some exceptions. This is brought out most clearly perhaps by H. Kyburg's "lottery paradox": the proposition that one in say, a million tickets in a lottery will win is acceptable, as are each of a milion propositions asserting that a particular ticket will not win. The "paradox" vanishes after a moment's thought on the underlying semantics: were we to agree that

$$\mathcal{A} \ = \ \{\varphi \in \mathcal{L} \ : \ \text{there is at most one } a \in V \text{ for which } a \not\models \varphi\} \ ,$$

we may run up against a pair φ and ψ of formulas in $\mathcal{A}$ such that the one counter-example to φ is different from the one counter-example to ψ, whence $\varphi \wedge \psi \notin \mathcal{A}$.

But as long as we concentrate on the syntactic side $\mathcal{A}$, choosing the semantics $\mathcal{H}$ to support the manipulations on $\mathcal{A}$ that we have decided to legitimize, there is hope for (And). In this regard, it is worthwhile noting what Pearl 1994 calls a "long-standing tension between the logical and probabilistic approaches to dealing with such exceptions": whereas the former is "prescriptive" (insofar as logic is simply a record of "conversational conventions"), probabilities are "descriptive" (be they measures of objective frequencies or subjective beliefs[1]).

[1]In the case of objective probabilities, it is useful to equate $\mathcal{L}$ with the set of *terms* of a formal language (ideally: typed), whereas in the case of subjective probabilities, it is

Now, the question is how could probabilities *describe* the *logical* rule (And)? A natural way to proceed is to accept precisely the formulas with probability greater than some fixed threshold $\alpha \in [0,1]$ (say, .999), given a probability function pr from $\mathcal{L}$ to the unit interval $[0,1]$

$$\mathcal{A}_{pr,\alpha} \quad = \quad \{\varphi \in \mathcal{L} \; : \; pr(\varphi) > \alpha\} \; .$$

The hitch is that exceptions add up: from $pr(\varphi) = 1 - \delta$ and $pr(\psi) = 1 - \epsilon$, one cannot, in general, do better than predict $pr(\varphi \wedge \psi) \geq 1 - (\delta + \epsilon)$. This suggests that we

(I) replace the condition that $pr(\varphi) > \alpha$ by the requirement that $1 - pr(\varphi)$ be *infinitesimal*, where infinitesimals are assumed to be closed under addition: if δ and ϵ are infinitesimals, then so is $\delta + \epsilon$.

The notion of an infinitesimal here is exactly that introduced by A. Robinson in his non-standard reconstruction of calculus. Reversing chronological order, we could, as an alternative to (I),

(II) explain away infinitesimals by $\epsilon\delta$-type limits (à la Bolzano-Weierstrass).

Pearl 1994 describes the work of Adams and Spohn in much this way, though with*out* the emphasis on (And). I have decided here to focus on (And) because, together with (Up), it supports a very direct and general analysis of approaches (I) and (II), using no more structure than that implicated by line (1) above. This is made precise by Theorems 1 and 2 in §§2.1 and 2.2 respectively, where, in particular, no appeal is made to numbers, be they in the standard unit interval $[0,1]$, or some non-standard copy thereof. This is not to say that the analysis given is incompatible with a numerical approach; only that it allows us to avoid all kinds of arithmetical complications — not to mention the somewhat embarrassing question: what probability function?

Without specifying a particular probability function, we return, in §3, to probabilities, beefing up the rule (Up) to a rule $(\mathrm{Up})_{\preceq}$, where $\preceq$ means "at least as probable." In this section, we refrain from making any commitment to the soundness or, for that matter, unsoundness of (And). $(\mathrm{Up})_{\preceq}$ applies not only to models of (And), but also to $\mathcal{A}_{pr,\alpha} = \{\varphi \in \mathcal{L} : pr(\varphi) > \alpha\}$, where pr is a probability function, and α is some number in $[0,1]$. (It is easy enough to introduce a rule restricting α say, to be above $\frac{1}{2}$). As weak as the rule $(\mathrm{Up})_{\preceq}$ might be, it nevertheless constitutes a step to understanding what is involved *qualitatively* in accepting formulas with probabilities that fall non-infinitesimally short of 1.

1.2 Related work. There is an evidently widespread belief that it is a non-trivial (if not hopeless) enterprise to reason *qualitatively* about formulas with probabilities greater than some fixed threshold non-infinitesimally short of 1. Pearl 1988 puts the matter as follows

---

convenient to take $\mathcal{L}$ to be the language's set of formulas (V the language's collection of models, and $\models$ its satisfaction relation).

> Probabilities that are infinitesimally close to 0 and 1 are very rare in the real world. Most default rules used in ordinary discourse maintain a certain percentage of exceptions, simply because the number of objects in every meaningul class is finite. Thus, a natural question to ask is, why study the properties of a logic that applies only to extreme probabilities? Why not develop a logic that characterizes moderately high probabilities, say probabilities higher than 0.5 or 0.9 — or more ambitiously, higher than α, where α is a parameter chosen to fit the domains of the predicates involved?
>
> The answer is that any such alternative logic would be extremely complicated and probably would need to invoke many axioms of arithmetic. [pages 493, 494]

A similar view can be found in Halpern and Rabin 1987, where "a logic to reason about likelihood" is developed relative to a *non*-probabilistic semantics. The probabilistic approach to likelihood presented in §3 (below) is not only qualitative but quite possibly simpler than might have been feared. It proceeds along lines similar to Scott 1964 and Segerberg 1971, as discussed further below.

The literature on non-monotonic formalisms that incorporate (And) as a basic or derived rule is vast and, for the uninitiated, downright bewildering. (See, for instance, Pearl 1994 and the references cited therein, plus Kyburg 1970 for earlier work.) What the present paper offers is a logical approach that departs minimally (if at all) from standard practice in classical mathematical logic. The one possible point of departure is the appeal to the *weak logic of generalized quantifiers* in Keisler 1970 (which arguably belongs to the mainstream of logic) for formalizing line (1) above,[2] and even then, the introduction of generalized quantifiers can be eliminated according to Theorems 1 and 2 below, resulting in ordinary predicate logic. Some specialists in the logic of generalized quantifiers seem to consider Theorem 1 part of the subject's folklore. It is implicit in van Lambalgen 1991, and appears in a disguised form as Theorem 5 of Alechina and van Lambalgen 1996, the inessential notational differences being due to that work's somewhat novel syntax (involving modality) and semantics (motivated by proof theory).

A crucial syntactic point that ought to be stressed is the expulsion of numbers from formulas below — in contrast, that is, to the quantitative approaches in Keisler 1985 and Halpern 1990, where numbers appear explicitly in formulas. The idea behind minimizing (explicit) reference to probabilities is to isolate what it is about probabilities that is essential to judgments of acceptability; but by opening the door to alternative non-probabilistic interpretations of formulas, the challenge then becomes showing that only the

[2]A fine point about (1) and weak logic is that the extension of $\mathcal{A}$ to a binary predicate $\mid\!\sim$ on $\mathcal{L}$ can be treated by passing from unary to binary generalized quantifiers, at the cost only of notational clutter; see §2.3.

probabilistic semantics need matter. (More concretely, the problem in establishing completeness is how to define a probability measure from syntactic entities that do not mention numbers.)

2 Reasoning according to preference: filters. A straightforward formalization of line (1) is provided by the *weak logic* $L(Q)$ for generalized quantifiers of Keisler 1970 where

(i) L is a first-order language and Q is a generalized quantifier symbol, inducing formulas $Qx\varphi$ (in addition to the usual closure rules on first-order L-formulas)

and

(ii) an $L(Q)$-model is a pair (M, q) consisting of a first-order L-model M and a family $q \subseteq \mathsf{Pow}(|M|)$ of subsets of the universe $|M|$ of M, so that (relativizing (1) to (M, q))

$$(M, q) \models Qx\varphi[f] \quad \text{iff} \quad \{a \in |M| : (M, q) \models \varphi[f^x_a]\} \in q$$

for every function f mapping variables to objects in $|M|$ (and where f^x_a is the function that maps x to a, but is otherwise identical to f).

2.1 Filter quantifiers. Given a set V, a *filter on* V is a non-empty family $\mathcal{H}$ of subsets of V satisfying (Up) and closed under intersections — viz., $V \in \mathcal{H}$; whenever $A \in \mathcal{H}$ and $A \subseteq B \subseteq V$, $B \in \mathcal{H}$; and for all $A \in \mathcal{H}$ and $B \in \mathcal{H}$, $A \cap B \in \mathcal{H}$ (thereby supporting (And)). These properties translate in $L(Q)$ to the (filter) schemes

$$\begin{array}{ll} \text{(Q1)} & Qx\ x = x \\ \text{(Q2)} & \forall x(\varphi \supset \psi)\ \supset\ (Qx\varphi \supset Qx\psi) \\ \text{(Q3)} & Qx\varphi\ \wedge\ Qx\psi\ \supset\ Qx(\varphi \wedge \psi)\ . \end{array}$$

Let us write $F[Q]$ for the $L(Q)$-theory induced by (Q1), (Q2) and (Q3), where φ and ψ are $L(Q)$-formulas with the same set of free variables. Note that $F[Q]$ holds for $Q = \forall$, or restricted universal quantification. The converse is not quite true: consider the standard model M of arithmetic, and let q be the family

$$\{A \subseteq \{0, 1, \dots\}\ :\ \{0, 1, \dots\} - A \text{ is finite}\}$$

of *co-finite* sets of natural numbers; then (M, q) is a model of $F[Q]$, even though q is a *non-principal* filter. (A filter q on V is *principal* if $\bigcap q \in q$, in which case $\bigcap q$ is said to *generate* q.) Nevertheless, the converse can be approximated through elementary extensions and the following bit of bookkeeping due to van Lambalgen 1991. Fix a relation symbol R not in L that accepts any finite positive number of arguments, and call an $L(Q)$-model (M, q) *relatively principal* if under some expansion of (M, q) to R,

$$Qx\varphi \quad \equiv \quad \forall x(R(x, \overline{y}) \supset \varphi) \tag{2}$$

holds for every $L(Q)$-formula φ with free variables $x, \overline{y}$ (where $\overline{y}$ is say, ordered according to some fixed well-ordering of variables). (2) says that Q can be taken to be universal quantification $\forall$ restricted to some set R of "generic" (or "normal") elements (modulo $\overline{y}$). These generic elements are "transcendental" in $\overline{y}$: assuming $\forall y Qx(x \neq y)$, they cannot be named by L-terms with free variables drawn from $\overline{y}$. (Hence, the necessity of adding $\overline{y}$ to R.)

Theorem 1.[3] *Every $L(Q)$-model of $F[Q]$ can be elementarily extended to a relatively principal $L(Q)$-model.*

Proof. Fix an $L(Q)$-model (M, q) of $F[Q]$, and a finite set Φ_0 of instances of (2). By the compactness theorem of weak logic (Keisler 1970), it suffices to show how to expand (M, q) to a model of Φ_0. The idea is to interpret R as the set

$$\{(F_\psi(\overline{a}), \overline{a}) \ : \ \psi \text{ is an } L(Q)\text{-formula}, \ \overline{a} \in dom(F_\psi)\} \ ,$$

for certain partial functions F_ψ (to be defined presently) from the set of finite sequences of (the universe) $|M|$ (of M) to $|M|$. Let n be the number of free variables in $Qx\psi$. The domain of F_ψ consists exactly of the n-tuples $\overline{a} \in |M|^n$ such that

$$(*) \qquad (M, q) \ \not\models \ Qx\psi[\overline{a}] \ .$$

Given such a sequence $\overline{a}$, let φ be the conjunction

$$\bigwedge\{\varphi_0 \ : \ Qx\varphi_0 \text{ has the same set of free variables as } Qx\psi, \\ \text{and `}Qx\varphi_0\text{'} \in \Phi_0, \text{ and } (M, q) \models Qx\varphi_0[\overline{a}]\}$$

(appealing here to the finiteness of Φ_0). It is understood that an empty conjunction is some tautology. Using (Q3) in case the set is non-empty, it follows that

$$(M, q) \ \models \ Qx\varphi[\overline{a}] \ .$$

Hence, by assumption $(*)$ and $Qx\varphi \wedge \neg Qx\psi \supset \exists x(\varphi \wedge \psi)$, to which (Q2) can, as pointed out to me by N. Alechina, be rewritten,

$$(M, q) \ \models \ \exists x(\varphi \wedge \neg\psi)[\overline{a}] \ .$$

It remains to choose some such witness for the value of $F_\psi(\overline{a})$. □

Remark. The interpretation of R suggested in the proof of Theorem 1 can be described roughly as follows: for every $L(Q)$-formula ψ such that $(M, q) \not\models Qx\psi$, throw in a witness to the set

$$\{\neg\psi\} \ \cup \ \{\varphi \in L(Q) \ : \ (M, q) \models Qx\varphi\}$$

of formulas. What is "rough" about this description is that (a) we should be more careful to specify what variables and constants to allow in the formulas, and (b) the required witnesses may not exist in M. The first point is a simple matter of bookkepping, while the second can be handled by appealing to the existence of elementary extensions that are *ω-saturated* — suggesting a restatement of Theorem 1 as

Proposition 1$'$. *Every ω-saturated $L(Q)$-model of $F[Q]$ is relatively principal.*

[3]See the note in §1.2 for bibliographic information.

In fact, Proposition $1'$ can be sharpened to so-called *recursively saturated* models (Barwise 1975), as the sets of formulas that must be realized can be given effectively as follows

$$\{(\neg Qx\psi) \supset \neg\psi\} \quad \cup \quad \{(Qx\varphi) \supset \varphi \ : \varphi \in L_n(Q)\}$$

for every $\psi \in L_n(Q)$, where L_n is an expansion of the language L to n fresh constants (abusing notation in identifying $L_n(Q)$ with the set of $L_n(Q)$-formulas with one free variable x). Note the similarity of $(\neg Qx\psi) \supset \neg\psi$ to Henkin expansions that witness existential statements. The next section adapts notions from a celebrated method, the *force* of which is to *omit* rather than to *realize types*.

2.2 Graded normality. The proper extensions mentioned in Theorem 1 can be avoided, provided (2) is weakened to allow for varying grades (rather than an absolute, either/or, notion) of genericity: add an argument place to R, for a relation symbol $\sqsubset$ to be used with the intuition that

$$u \sqsubset v \quad \text{iff} \quad u \text{ is "more generic" than } v \ .$$

More precisely, let L be a countable first-order language, $\sqsubset$ be a fresh binary relation symbol (not in L), and let $y \sqsubseteq z$ abbreviate $(y \sqsubset z) \vee (y = z)$. Given an L-model M, let L_M be L together with a fresh constant symbol for every object in M. To simplify the notation, we will identify an object a in M with its constant symbol in L_M, and write M for the L_M-model obtained by expanding the L-model M to L_M, with every a in M interpreted as a.

Theorem 2. *For every countable (or finite) $L(Q)$-model (M,q) of $F[Q]$, there is a transitive binary relation $\prec \ \subseteq |M| \times |M|$ such that for every $L_M(Q)$-formula φ with exactly one free variable x,*

$$(3) \qquad Qx\varphi \quad \equiv \quad (\forall z)(\exists y \sqsubseteq z)(\forall x \sqsubset y)\ \varphi$$

holds in $(M, q, \prec)$.

Remarks.

1. (3) describes an $\epsilon\delta$-limit/asymptotic/cofinal-type quantification, with (2) falling out as the special case given by
$$u \sqsubset v \quad \equiv \quad R(u)$$
(i.e., the second argument v in $\sqsubset$ is vacuous).
2. The converse of Theorem 2 (the soundness of $F[Q]$ under (3)) is trivial: the schemes (Q1) and (Q2) follow from (3) alone, while (Q3) is a consequence of the transitivity of $\prec$ (as well as (3)). (I have not investigated what generality (3) buys beyond that of (2), in the absence of the assumption that $\sqsubset$ is transitive.)
3. The restriction to one free variable is inessential, and is made only to simplify notation, allowing us to suppress the subscripts $\overline{y}$ on $\sqsubset$ in (3).
4. It is considerably more complicated to describe an interpretation of $\sqsubset$ validating (3) in Theorem 2 than an interpretation of R supporting (2) in Theorem 1. In this connection, it is interesting to note the sentiment

any fool can realize a type, but it takes a model-theorist to omit one

expressed in Sacks 1972. The twist in Theorem 2 is that the "ontological promiscuity" in saturation arguments is avoided by a purely combinatorial argument (with*out* resorting to any of the model-theorist's tools, such as completeness).

Proof of Theorem 2. Fix an $L(Q)$-model (M,q) of $F[Q]$, and partition the set Φ of $L_M(Q)$-formulas with exactly one free variable x as follows

$$\begin{aligned}\Phi_+ &= \{\varphi \in \Phi \ : \ (M,q) \models Qx\varphi\} \\ \Phi_- &= \{\psi \in \Phi \ : \ (M,q) \not\models Qx\psi\} \ .\end{aligned}$$

If for every $\varphi \in \Phi_+$, $(M,q) \models \forall x\varphi$, then we can set $\prec$ to $\{(a,a) : a \in |M|\}$ and we are done. Otherwise, choose a $\varphi_0 \in \Phi_+$ and $a_0 \in |M|$ such that

$$(M,q) \quad \not\models \quad \varphi_0\,[a_0] \ ,$$

and let $\{\varphi_0, \varphi_1, \varphi_2, \varphi_3, \ldots\}$ be an enumeration of Φ. We will define $\prec$ by finite approximations $\prec^i$ (for $i \geq 0$), with

$$\prec \quad = \quad \bigcup_{i \geq 0} \prec^i \ .$$

The plan is to construct for each $i \geq 0$, a finite transitive relation $\prec^i$ such that the following four conditions hold, with the understanding that

$$\begin{aligned}\Phi_+^i &= \Phi_+ \cap \{\varphi_j : j \leq i\} \\ \Phi_-^i &= \Phi_- \cap \{\varphi_j : j \leq i\} \\ E^i &= \{a \in dom(\prec^i) : (\forall b \prec^i a)\ a \prec^i b\} \qquad (= \text{ the set of } \prec^i\text{-minimal points}).\end{aligned}$$

(C1) $\prec^i \subseteq \prec^{i+1}$.
(C2) $dom(\prec^{i+1} - \prec^i) \subseteq E^{i+1}$.
(C3) For every $a \in E^i$, $(M,q) \models \bigwedge \Phi_+^i[x/a]$.
(C4) There is a "witness" map $w_i : \Phi_-^i \to E^i$ such that for every $\psi \in \Phi_-^i$,

$$(M,q) \quad \models \quad \neg\psi\ [x/w_i(\psi)] \ ,$$

and for every $j > i$, and every $a \in |M|$,

$$a \preceq^j w_i(\psi) \quad \text{implies} \quad w_j(\psi) \prec^j a \ .$$

What makes (C1)-(C4) interesting is

Lemma A. *If $\prec$ is the union $\bigcup_i \prec^i$ of finite transitive relations $\prec^i$ satisfying (C1), (C2), (C3) and (C4), then the required equivalence (3) holds.*

Proof. Suppose (first) that $(M,q) \models Qx\varphi$. Given $a \in |M|$, either there is some $a' \in |M|$ such that $a' \prec a$, or not. If not, then

$$(M,q,\prec) \quad \models \quad (\exists y \sqsubseteq a)(\forall x \sqsubset y)\ \varphi$$

holds vacuously (by the definition of $\sqsubseteq$ as the disjunction of $\sqsubset$ with equality). Otherwise, choose an i such that $\varphi \in \Phi^i_+$ and appeal to (C1), (C2) and (C3) for an $a' \in E^i$ such that $a' \prec^i a$.

Next, assume $(M, q) \not\models Qx\varphi$. Choose an i such that $\varphi \in \Phi^i_-$, and conclude from (C1) and (C4) that

$$(M, q, \prec) \quad \models \quad (\forall y \sqsubseteq z)(\exists x \sqsubset y)\ \neg\varphi[z/w_i(\varphi)] \ .$$

$\Box$ (Lemma A)

To push the construction of $\prec$ through, the following will be useful.

Lemma B. *For all $\varphi \in \Phi_+$, $\psi \in \Phi_-$, and every finite subset A_0 of $\{a \in |M| : |M| - \{a\} \in q\}$,*

$$\{a \in |M| : (M, q) \models (\varphi \wedge \neg\psi)[x/a]\} \quad \not\subseteq \quad A_0 \ .$$

Proof. Repeated applications of (Q3) give

$$(M, q) \quad \models \quad Qx(\varphi \wedge \bigwedge_{a \in A_0} x \neq a) \ ,$$

and therefore, if $\psi \in \Phi_-$ then (Q2) implies

$$(M, q) \quad \not\models \quad \forall x \quad ((\varphi \wedge \bigwedge_{a \in A_0} x \neq a) \supset \psi) \ ,$$

i.e.,

$$(M, q) \quad \not\models \quad \forall x \quad ((\varphi \wedge \neg\psi) \supset \bigvee_{a \in A_0} x = a) \ ,$$

as required. $\Box$ (Lemma B)

Looking more closely at $\{a \in |M| : |M| - \{a\} \in q\}$, observe that (Q2) implies

$$\{a \in |M| : |M| - \{a\} \notin q\} \quad = \quad \bigcap q$$

and

Lemma C. *For every $a \in \bigcap q$ and every $\varphi \in \Phi_+$, $(M, q) \models \varphi[x/a]$.*

Proof. If $(M, q) \not\models \varphi[x/a]$, then

$$\{b \in |M| : (M, q) \models \varphi[x/b]\} \quad \subseteq \quad |M| - \{a\} \ ,$$

so that $\varphi \in \Phi_+$ and (Q2) imply $|M| - \{a\} \in q$ (i.e., $a \notin \bigcap q$). $\Box$ (Lemma C)

Next, let I^i be the image of $\prec^i$ minus E^i

$$I^i \quad = \quad \{a \in |M| - E^i : (\exists b \in |M|)\ b \prec^i a\}$$

and add to the list (C1)-(C4)

(C5) For every $a \in I^i$, there is a $\varphi \in \Phi^i_+$ such that $(M, q) \not\models \varphi[a]$.

(C6) The witness map w_i mentioned in (C4) is surjective (*on*to E^i) and has the additional property that for all $\psi, \psi' \in \Phi^i_-$ such that $\psi \neq \psi'$, if $w_i(\psi) = w_i(\psi')$, then $w_i(\psi) \in \bigcap q$.

Let us turn finally to the definition of $\prec^i$ (and w_i). The initial stage $i = 0$ is trivial: since $\varphi_0 \in \Phi_+$, we can set $\prec^0 = \emptyset$ (whence $E^0 = \emptyset = w_0$). Now, consider stage $i+1$.

Case 1: $\varphi_{i+1} \in \Phi_+$. Let $N = \{a \in E^i : (M,q) \not\models \varphi_{i+1}[x/a]\}$. By Lemma C, for every $a \in N$, $a \notin \bigcap q$. Hence by (C6), each $a \in N$ has a unique $\psi_a \in \Phi^i_-$ such that $w_i(\psi_a) = a$. Next, apply (Q3) and Lemma B to define a function $new : N \to |M|$ such that for every $a \in N$,

$$(M,q) \quad \models \quad ((\neg\psi_a) \wedge \bigwedge \Phi^{i+1}_+)[x/new(a)]$$

(whence $new(a) \notin I^i$ by (C5)) and

$$new(a) \in (E^i - N) \cup \{new(a') : a' \in N - \{a\}\} \quad \text{implies} \quad new(a) \in \bigcap q \ .$$

Then set $\prec^{i+1}$ to be the transitive closure of

$$\prec^i \ \cup \{(new(a), a) : a \in N\} \ \cup \ \{(new(a), new(a)) : a \in N\}$$

and define $w_{i+1} : \Phi^{i+1}_- \to E^{i+1}$ by

$$w_{i+1}(\psi) \quad = \quad \begin{cases} new(w_i(\psi)) & \text{if } w_i(\psi) \in N \\ w_i(\psi) & \text{otherwise} \end{cases}$$

for every $\psi \in \Phi^{i+1}_-$ ($= \Phi^i_-$ by the case assumption).

Case 2: $\varphi_{i+1} \in \Phi_-$. Writing ψ for φ_{i+1}, choose, by (Q3) and Lemma B, an a such that

$$(M,q) \quad \models \quad ((\neg\psi) \wedge \bigwedge \Phi^i_+)[x/a]$$

and

$$a \in \{w_i(\psi) : \psi \in \Phi^i_-\} \quad \text{implies} \quad a \in \bigcap q \ .$$

Then set

$$\begin{aligned} \prec^{i+1} \quad &= \quad \prec^i \ \cup \{(a, a_0)\} \\ w_{i+1} \quad &= \quad w_i \cup \{(\psi, a)\} \end{aligned}$$

(recalling that a_0 was the element chosen at the outset satisfying $\neg\varphi_0$, and that $a \notin I^i$, by (C5)).

Verifying conditions (C1) through (C6) is routine. $\Box$

2.3 The binary case (from $\mathcal{A}$ to $\mid\!\sim$). Theorems 1 and 2 generalize to binary quantifiers (with only notational complications) as follows. To step from a unary quantifier up to a binary quantifier, an L-model M is paired with a binary relation $q \subseteq \mathsf{Pow}(|M|) \times \mathsf{Pow}(|M|)$ such that

$$(M,q) \models Qx(\varphi,\psi)[f] \quad \text{iff} \quad \{a \in |M| : (M,q) \models \varphi[f^x_a]\} \ q \ \{a \in |M| : (M,q) \models \psi[f^x_a]\} \ .$$

The weak completeness and compactness theorems of Keisler 1970 lift immediately to this setting (as worked out, for instance, in Westerståhl 1989). The filter schemes (Q1), (Q2) and (Q3) turn into

$$\begin{aligned}&Qx(\varphi,\varphi)\\&\forall x(\psi\supset\psi') \;\supset\; (Qx(\varphi,\psi)\supset Qx(\varphi,\psi'))\\&Qx(\varphi,\psi)\wedge Qx(\varphi,\psi') \;\supset\; Qx(\varphi,\psi\wedge\psi')\end{aligned}$$

respectively, with line (2) becoming

$$Qx(\varphi,\psi) \quad\equiv\quad \forall x((\varphi\wedge R_{\varphi,x}(x,\overline{y}))\supset\psi) \tag{4}$$

where $\overline{y}$ lists the free variables $\neq x$ in φ as well as ψ. Line (4) supports a reading of the formula $Qx(\varphi,\psi)$ as "for all relevant φ-x's, ψ." Theorem 1 can be lifted to these binary forms, under the additional condition that R is "extensionalized" so that

$$\forall x\overline{y}(\varphi\equiv\varphi') \;\supset\; \forall x\overline{y}(R_{\varphi,x}(x,\overline{y})\equiv R_{\varphi',x}(x,\overline{y}))$$

for all $L(Q)$-formulas φ and φ' with the same free variables $x,\overline{y}$. Similar remarks apply to Theorem 2. (I.e., the relation symbol $\sqsubset$ must also be relativized to the antecedent φ and the quantified variable x, although its extension can be arranged to depend on φ only up to $\equiv$.)

3 Between preferences and probabilities: quasi-filters. Having upgraded a "normality" predicate R into a "preference" relation $\sqsubset$, let us proceed further, into probability measures, concentrating on $\mathcal{H}$ (rather than on $\mathcal{A}$). Fix a non-empty set V, and a field $\mathcal{F}$ of subsets of V (containing V and closed under $\cup$ and $\bar{\cdot}$). Recall that a *(finitely additive) probability function on* $\mathcal{F}$ is a function $\mu:\mathcal{F}\to[0,1]$ such that $\mu(V)=1$, and for all $A,B\in\mathcal{F}$,

$$\mu(A\cup B) \;=\; \mu(A-B)+\mu(B-A)+\mu(A\cap B)\ .$$

3.1 Weakening filters non-conservatively. A family $\mathcal{H}\subseteq\mathcal{F}$ of sets in $\mathcal{F}$ is *sizable* if there is a probability function μ on $\mathcal{F}$ such that for every $A\in\mathcal{F}$,

$$A\in\mathcal{H} \quad\text{iff}\quad \mu(A)>\mu(\overline{A}) \quad \left(\text{i.e., } \mu(A)>\frac{1}{2}\right)\ .$$

Given $\alpha\in[0,1]$, call a family $\mathcal{H}\subseteq\mathcal{F}$ *α-sizable* if there is a probability function μ on $\mathcal{F}$ such that for every $A\in\mathcal{F}$,

$$A\in\mathcal{H} \quad\text{iff}\quad \mu(A)>\alpha\ .$$

Let us record some properties of α-sizable families differentiating them from filters.

Proposition 3.

(i) *Sizable families verify*

$$\text{(Half)}\qquad \frac{\overline{A}\notin\mathcal{H} \qquad A\cup\overline{B}\notin\mathcal{H}}{A\cup B\in\mathcal{H}}\ .$$

(ii) *For* $n \geq 1$, *let*

$$(\mathrm{Cov}_n) \qquad \bigvee_{\delta:\{1,2,\ldots,n\}\to\{+,-\}} \Big((\bigcup_{i=1}^{n} A_i^{\delta(i)}) \in \mathcal{H} \Big)$$

where $A_i^+ = A_i$ *and* $A_i^- = \overline{A}_i$. *Then* (Cov_n) *is valid for* α*-sizable families iff* $\alpha < 1 - 2^{-n}$.

Proof. Part (i) is trivial: $\mu(\overline{A}) \leq \frac{1}{2}$ and $\mu(A \cup \overline{B}) \leq \frac{1}{2}$ imply $\mu(A \cup B) = 1$, since

$$\begin{aligned} \mu(A \cup B) &= \mu(A - B) + \mu(B) \\ &= \mu(\overline{B}) - \mu(\overline{B} - A) + \mu(B) \\ &= 1 - \mu(\overline{B} - A) \end{aligned}$$

where $\mu(\overline{B} - A) = 0$ (because $\mu(\overline{A}) \leq \frac{1}{2}$ and $\mu(A \cup \overline{B}) \leq \frac{1}{2}$).

Part (ii) requires a little more work, the point being to prove that for every $n \geq 1$, and sequence $A_1, \ldots, A_n \in \mathcal{F}$,

$$(5) \qquad \max_{\delta:\{1,2,\ldots,n\}\to\{+,-\}} \mu(\bigcup_{i=1}^{n} A_i^{\delta(i)}) \;\geq\; 1 - 2^{-n}$$

where equality can be arranged for suitable $A_1, \ldots, A_n$. To establish the equality in (5), we induct on n, with the induction hypothesis strengthened as follows (taking μ to be the uniform distribution over the power set of $V = \{1, 2, \ldots, 2^n\}$)

IH_n: there is a sequence $A_1, \ldots, A_n$ of subsets of $\{1, 2, \ldots, 2^n\}$ and an enumeration $\delta_1, \delta_2, \ldots, \delta_{2^n}$ of functions from $\{1, 2, \ldots, n\}$ to $\{+, -\}$ such that for each $j \in \{1, 2, \ldots, 2^n\}$,

$$\{1, 2, \ldots, 2^n\} - \bigcup_{i=1}^{n} A_i^{\delta_j(i)} \;=\; \{j\}\ .$$

The base case $n = 1$ is immediate. Next, assuming IH_n, define, for each $i \in \{1, 2, \ldots, n\}$,

$$B_i \;=\; A_i \;\cup\; \{2^n + j \;:\; j \in A_i\}$$

and $B_{n+1} = \{1, 2, \ldots, 2^n\}$. Then for each $j \in \{1, 2, \ldots, 2^n\}$,

$$\{1, 2, \ldots, 2^{n+1}\} - (\overline{B}_{n+1} \cup \bigcup_{i=1}^{n} B_i^{\delta_j(i)}) \;=\; \{j\}$$

$$\{1, 2, \ldots, 2^{n+1}\} - (B_{n+1} \cup \bigcup_{i=1}^{n} B_i^{\delta_j(i)}) \;=\; \{2^n + j\}$$

from which IH_{n+1} easily follows, and hence, the equality in (5). As for the inequality in (5), a function $\delta : \{1, 2, \ldots, n\} \to \{+, -\}$ verifying it (given any fixed sequence $A_1, \ldots, A_n$) can be constructed by choosing $\delta(1)$ so that

$\mu(A_1^{\delta(1)}) \geq \frac{1}{2}$, and then $\delta(2)$ so that $\mu(A_2^{\delta(2)} - A_1^{\delta(1)}) \geq \frac{3}{4} - \mu(A_1^{\delta(1)})$, and so forth, the general pattern being that for every $k \in \{1, 2, \ldots, n\}$, $\delta(k)$ satisfies

$$\mu(A_k^{\delta(k)} - \bigcup_{i=1}^{k-1} A_i^{\delta(i)}) \ \geq \ 1 - 2^{-k} - \mu(\bigcup_{i=1}^{k-1} A_i^{\delta(i)}) \ .$$

□

3.2 A generalization of filters. To avoid (Half) and (Cov_n), we quantify away α as follows. Call $\mathcal{H}$ *additive* if it is α-sizable for some α. That is, $\mathcal{H}$ is additive iff for some probability function $\mu : \mathcal{F} \to [0,1]$, and $\alpha \in [0,1]$, $\mathcal{H} = \{A \in \mathcal{F} : \mu(A) > \alpha\}$. Rewriting $(\mathrm{Up})_{\preceq}$ (from §1), with $\mu(A) \leq \mu(B)$ in place of $A \preceq B$, we get

$$(\mathrm{Up})_\mu \quad \frac{A \in \mathcal{H} \qquad \mu(A) \leq \mu(B)}{B \in \mathcal{H}} \ .$$

Note, however, that unless we know what μ is, we cannot assert $(\mathrm{Up})_\mu$. Could it be then that the best we could do to characterize additive families is to assert (Up)?

The following counter-example shows that there is more structure in additive families to account for.

(†) Let $V \supseteq \{1,2,3,4\}$ and $\mathcal{H} = \{A \subseteq V : \{1,3\} \subseteq A \text{ or } \{2,4\} \subseteq A\}$. Although $\mathcal{H}$ verifies (Up), it is not additive: were it induced by μ, then as $\{1,3\} \in \mathcal{H}$ and $\{1,2\} \notin \mathcal{H}$, $\mu(\{3\}) > \mu(\{2\})$; but as $\{2,4\} \in \mathcal{H}$ and $\{3,4\} \notin \mathcal{H}$, $\mu(\{2\}) > \mu(\{3\})$.

To strengthen (Up), some notation is useful saying (roughly) that a sequence $A_1, A_2, \ldots, A_n$ of sets in $\mathcal{F}$ is heavier than a sequence $B_1, B_2, \ldots, B_n$ (of equal length). With that in mind, let us write

$$\sum_{i=1}^{n} A_i \ \geq \ \sum_{i=1}^{n} B_i$$

to mean that for every $a \in V$,

$$\sum_{i=1}^{n} \chi^{A_i}(a) \ \geq \ \sum_{i=1}^{n} \chi^{B_i}(a)$$

where χ^A is the characteristic function of A (mapping elements of A to 1, and elements of $V - A$ to 0). Now, consider the condition

(σ) for all sequences $A_1, \ldots, A_n$ and $B_1, \ldots, B_n \in \mathcal{F}$,

$$\sum_{i=1}^{n} A_i \geq \sum_{i=1}^{n} B_i \quad \text{implies} \quad (\exists i \in \{1, 2, \ldots, n\})$$
$$A_i \in \mathcal{H} \text{ or } B_i \notin \mathcal{H} \ .$$

(Up) is just the case $n = 1$ (as $A_1 \geq B_1$ just means $A_1 \supseteq B_1$). Example (†) above is handled by

$$\frac{A \cup C \notin \mathcal{H} \qquad B \cup (C - A) \in \mathcal{H} \qquad A \cup (D - B) \in \mathcal{H}}{B \cup D \in \mathcal{H}}$$

which follows from the case $n = 2$.

Theorem 4. *Assume $\mathcal{F}$ is finite. Then a family $\mathcal{H} \subseteq \mathcal{F}$ is additive iff it validates (σ).*

Proof. $\mathcal{H}$ is additive iff the inequality $<_{\mathcal{H}}$ given by $(\mathcal{F} - \mathcal{H}) \times \mathcal{H}$ can be translated into a satisfiable system of linear inequalities as follows. Without loss of generality, let $\mathcal{F} = \mathsf{Pow}(V)$ and $V = \{1, 2, \ldots, \hat{n}\}$. Fix an enumeration $\{(\hat{A}_1, \hat{B}_1), \ldots, (\hat{A}_{\hat{m}}, \hat{B}_{\hat{m}})\}$ of $(\mathsf{Pow}(V) - \mathcal{H}) \times \mathcal{H}$, and observe that $\mathcal{H}$ is additive iff

(P) there exist real numbers $x_1, \ldots, x_{\hat{n}} \geq 0$ such that

$$\sum_{j=1}^{\hat{n}} \alpha_{ij} x_j \quad > \quad 0 \qquad \text{for every } i \in \{1, \ldots, \hat{m}\}$$

where for every $i \in \{1, \ldots, \hat{m}\}$ and $j \in \{1, \ldots, \hat{n}\}$,

$$\alpha_{ij} \quad = \quad \begin{cases} 1 & \text{if } j \in \hat{B}_i - \hat{A}_i \\ -1 & \text{if } j \in \hat{A}_i - \hat{B}_i \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathbf{A}$ be the $\hat{m} \times \hat{n}$ matrix (α_{ij}), and appeal to the following well-known fact from linear algebra (e.g., Strang 1980, p. 333)

> Either $\mathbf{Ax} \geq \mathbf{b}$ has a nonnegative solution $\mathbf{x}$, or else there is a $\mathbf{y}$ such that $\mathbf{yA} \geq \mathbf{0}, \mathbf{yb} < \mathbf{0}$ and $\mathbf{y} \leq \mathbf{0}$.

Passing from $\geq$ to $>$, (P) has the "dual"

(D) there exist real numbers $y_1, \ldots, y_{\hat{m}} \geq 0$, not all 0, such that

$$\sum_{i=1}^{\hat{m}} y_i \alpha_{ij} \quad \leq \quad 0 \qquad \text{for every } j \in \{1, \ldots, \hat{n}\} \ .$$

The reals in (P) and (D) can be assumed to be rationals, and after clearing the denominators to make the y_i's in (D) positive integers, y_i can be taken to be the number of copies of the pair $\hat{A}_i, \hat{B}_i$ in

(σ') for all sequences $A_1, \ldots, A_n$ and $B_1, \ldots, B_n \in \mathcal{F}$,

$$(\forall i \in \{1, 2, \ldots, n\}) \ A_i <_{\mathcal{H}} B_i \quad \text{implies}$$
$$\text{not} \ \sum_{i=1}^{n} A_i \geq \sum_{i=1}^{n} B_i \ .$$

Now, (σ') is just the contrapositive of (σ). □

Remarks.

1. Theorem 4 is reminiscent of a theorem due to Kraft, Pratt and Seidenberg 1959, characterizing binary relations on finite Boolean algebras that can be read "at least as probable as." That characterization is established again in Scott 1964, employing methods similar to that used in the proof of Theorem 4 above, although I do not see how to reduce Theorem 4 to the Kraft, Pratt, Seidenberg/Scott result, as the notion of an additive family quantifies away even more information than the comparative relation "at least as probable as." Also, the "primal/dual" argument above applies more generally than "the general method" (Theorem 1.1) of Scott 1964.
2. Clearly, (σ) is satisfied by every filter $\mathcal{H}$: taking the contrapositve (σ') of (σ), suppose that for every $i \in \{1, 2 \ldots, n\}$, $A_i \notin \mathcal{H}$ and $B_i \in \mathcal{H}$; then $\bigcap_i B_i \in \mathcal{H}$, and for every $k \in \{1, 2, \ldots, n\}$, $\bigcap_i B_i \not\subseteq A_k$, whence $\sum_i A_i \not\geq \sum_i B_i$.
3. To force $\alpha \geq \frac{1}{2}$, it suffices to add a rule of non-empty intersection (considerably weaker than (And))

$$\text{(NEI)} \qquad \frac{A \in \mathcal{H} \qquad B \in \mathcal{H}}{A \cap B \neq \emptyset} \ .$$

   For $\alpha = \frac{m}{n}$, with $m < n$, add the rule that for any sequence $A_1, \ldots, A_n$ that partitions V, there is a subset I of $\{1, \ldots, n\}$ with cardinality m such that

$$\bigcup_{i \in I} A_i \ \in \ \mathcal{H} \ .$$

   In yet another direction, the rule $(\mathrm{Cov})_n$, from Proposition 3(ii), adds the restriction that $\alpha < 2^{-n}$.
4. To see that each case $n - 1$ of (σ) is insufficient, let $V = \{1, 2, \ldots, n^2\}$, and, recalling the definition of $<_{\mathcal{H}}$ in the proof above of Theorem 4, consider

$$\begin{array}{rl}
\{1, 2, 3, \ldots, n\} & <_{\mathcal{H}} \\
\{1, 2n+1, 3n+1, \ldots, (n-1)n+1\} & \\
\{2n+1, 2n+2, 2n+3, \ldots, 2n+n\} & <_{\mathcal{H}} \\
\{2, 2n+2, 3n+2, \ldots, (n-1)n+2\} & \\
\{3n+1, 3n+2, 3n+3 \ldots, 3n+n\} & <_{\mathcal{H}} \\
\{3, 2n+3, 3n+3, \ldots, (n-1)n+3\} & \\
\vdots & \\
\{(n-1)n+1, (n-1)n+2, \ldots, n^2\} & <_{\mathcal{H}} \\
\{n, 2n+n, 3n+n, \ldots, n^2\} \ . &
\end{array}$$

   (Note that if the left hand sides are packaged as an $n \times n$ matrix $\mathbf{A}$, then the right hand sides form the transpose $\mathbf{A}^T$ of $\mathbf{A}$.)

4 Conclusion: directions from and applications to natural language. Although the approaches pursued in §§2 and 3 are compatible (insofar as (Up) and (And) imply (σ)), the thrust of §2 — viz., extending (And) — runs opposite that of §3, the canonical models of which, $\mathcal{A}_{pr,\alpha}$, fail to verify (And). Were we to adopt (And), then §3 would become uninteresting; were we to reject it, then §2 would become pointless. My reason for keeping the two sections in one paper is that I see no compelling argument that can (once and for all) settle the matter for or against (And). Intuitions about acceptability simply differ — intuitions expressed in languages, begging to be investigated. Among these are natural languages such as English, the semantics of which might be explored in the hope, for example, of shedding light on the acceptability of (And). The skeptical reader is bound to protest at such naiveté: natural languages are not formal; surely!

There is no denying that first-order logic is formal in a way that English is not. Even so, there is a growing body of work inspired by R. Montague's slogan "English as a formal language" (1970/4), rephrased in more recent years as "Natural language as a programming language" (motivated by investigations into so-called "dynamic semantics," going back to, among other papers, Kamp 1981). More concretely, *generic* sentences, such as "Birds fly," provide a natural testing ground for questions about acceptability. And, in a slightly different direction, the very idea of formalizing, if not mechanizing, natural language leads to the problem of accepting exceptions. Briefly put, there is, I think, room for both §§2 and 3 in exploring the semantics of natural language.

4.1 Defeasibility in natural language interpretation. The transformation of "English as a formal language" into "natural language as a programming language" can be depicted by the equation

$$\begin{aligned} \textit{Natural language interpretation} \quad &= \\ \textit{logic}\ \text{[e.g. model-theoretic} \\ \text{semantics in Montague grammar]} \\ +\ \ \textit{control}\ \text{[e.g. choice of} \\ \text{formula/disambiguation]} \quad &, \end{aligned}$$

the point being that some notion of "control" is required to formalize a natural language utterance before it can be subjected to a logical analysis. Moreover, that logical analysis may, in turn, feed back into how to formalize the natural language utterance (and so on). This suggests a broader construal of "logic" (as applied to natural language) than that traditionally associated with model theory. For instance, the resulting logic should, many argue, allow for defeasibility, a particular preferential approach to which, verifying (And), is pursued in Asher 1993 and Lascarides and Asher 1993. Integrating this with statistical/probabilistic information is an obvious task crying out for attention. Stepping back a bit, basic logical issues raised by the problem of control are discussed in Fernando 1996, where the challenge to compositionality posed by the context-dependence of disambiguation is taken up, and a logical approach based on generalized quantifiers is outlined.

4.2 A case study: back to many (naturally). Having derived much inspiration from Keisler 1970, let us close by noting that a linguistically motivated analysis of *many* not only calls the filter properties (sound for *uncountably many*) into question, but also introduces (at least) two other dimensions to the classical picture in Keisler 1970 (or, for that matter, in Westerståhl 1989): vagueness and intensionality. Passing from cardinalities to (vague subjective) probabilities over "possible worlds", Fernando and Kamp 1996 propose to define there to be *many* φ-x's iff the number of φ-x's is more than what might be expected:

$$(\mathsf{Many}\text{-}x)(\varphi) \quad \text{iff} \quad \bigvee_{n\geq 1} \Big((\exists_{\geq n} x)(\varphi) \ \wedge \ \mathsf{Prob}((\exists_{<n} x)(\varphi)) \Big) \tag{6}$$

where Prob is to be interpreted as an additive family $\mathcal{H}$. A formal study of (6) within an intensionalized form of weak logic is currently being written up (with H. Kamp), drawing in no small measure on Theorem 4 above (which can be encoded within modal logic, in the manner of Segerberg 1971).

Acknowledgments. I am very grateful to N. Alechina, H. Kamp and M. van Lambalgen for helpful discussions.

References

N. Alechina and M. van Lambalgen. Generalized quantification as substructural logic. *Journal of Symbolic Logic*, 61(3), 1996.

Nicholas Asher. *Reference to Abstract Objects in Discourse.* Kluwer Academic Publishers, Dordrecht, 1993.

Jon Barwise. *Admissible Sets and Structures.* Springer-Verlag, Berlin, 1975.

T. Fernando and H. Kamp. Expecting many. In *Proc. Semantics and Linguistic Theory VI.* Cornell University, 1996.

Tim Fernando. Ambiguity under changing contexts. Manu-script, 1996. Preliminary version presented in *Mathematics of Language 4* (Philadelphia, October 1995), and submitted for publication.

J.Y. Halpern and M.O. Rabin. A logic to reason about likelihood. *Artificial Intelligence*, 32, 1987.

Joseph Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46, 1990.

Hans Kamp. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof, editors, *Formal Methods in the Study of Language.* Mathematical Centre Tracts 135, Amsterdam, 1981.

H. Jerome Keisler. Logic with the quantifier "there exist uncountably many". *Annals of Mathematical Logic*, 1, 1970.

H. Jerome Keisler. Probability quantifiers. In J. Barwise and S. Feferman, editors, *Model-Theoretic Logics*, Perspectives in Mathematical Logic. Springer-Verlag, 1985.

C.H. Kraft, J.W. Pratt, and A. Seidenberg. Intuitive probability on finite sets. *Ann. Math. Statist.*, 30, 1959.

Henry E. Kyburg, Jr. Conjunctivitis. In M. Swain, editor, *Induction, Acceptance, and Rational Belief.* D. Reidel, Dordrecht, 1970.

Michiel van Lambalgen. Natural deduction for generalized quantifiers. In J. van der Does and J. van Eijck, editors, *Generalized Quantifier Theory and Applications.* Dutch Network for Language, Logic and Information (Amsterdam), 1991. To appear in CSLI Lecture Notes.

A. Lascarides and N. Asher. Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and Philosophy*, 16(5), 1993.

Richard Montague. English as a formal language. In R. Thomason, editor, *Formal philosophy.* Yale University Press, New Haven, 1974.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Mateo, California, 1988.

Judea Pearl. From Adams' conditionals to default expressions, causal conditionals and counterfactuals. In E. Eells and B. Skyrms, editors, *Probability and Conditionals*, Cambridge Studies in Probability, Induction and Decision Theory. Cambridge University Press, 1994.

Gerald E. Sacks. *Saturated Model Theory.* W.A. Benjamin, Reading, Mass., 1972.

Dana Scott. Measurement structures and linear inequalities. *Journal of Mathematical Psychology*, 1:233–247, 1964.

Krister Segerberg. Qualitative probability in a modal setting. In J.E. Fenstad, editor, *Second Scandinavian Logic Symposium*, pages 341–352. North-Holland, 1971.

Gilbert Strang. *Linear Algebra and Its Applications.* Academic Press, Orlando, 1980.

Dag Westerståhl. Quantifiers in formal and natural languages. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume IV. D. Reidel, Dordrecht 1989.

11

Processing ambiguity non-deterministically: A modal logic for disambiguation

March 11, 1997

1. Introduction

An obvious step in interpreting a set E of (possibly) ambiguous expressions is to associate with each expression $e \in E$ a set $\delta(e) \subseteq \Phi$ of unambiguous formulas from some set Φ, for which

$$\delta(e) \quad \approx \quad \{\varphi \in \Phi \ : \ e \text{ can be disambiguated as } \varphi\} \ .$$

Such a function $\delta : E \to \mathsf{Pow}(\Phi)$ does not, however, take context-sensitivity into account: $\delta(e)$ says only what formulas (in Φ) e *can* be disambiguated to, and nothing about the contexts under which e is disambiguated to a particular $\varphi \in \delta(e)$. Accordingly, functions $[\cdot,\cdot,\cdot] : (E \times E \times E) \to E$, with $\Phi \subset E$, are considered in Fernando [**3**], subject to the intuition that for all expressions $l, e, r \in E$,

$$\begin{aligned} [l,e,r] \quad \approx \quad & \text{an expression disambiguating } e \text{, assuming} \\ & l \text{ occurs to the left (or past) of } e \ , \\ & \text{and } r \text{ to the right (or future) of } e \ . \end{aligned}$$

(As with δ, different choices of $[\cdot,\cdot,\cdot]$ are possible, the point being to study the consequences of such choices.) The disambiguation of e by $[l,e,r]$ is, unlike $\delta(e)$, not only

(a) sensitive to the context l, r

but also

(b) deterministic, insofar as $[l,e,r]$ is a specific expression, rather than a set of unambiguous formulas

and

(c) partial, in the sense that $[l,e,r]$ may (assuming $e \in E - \Phi$) well be an element of $E - \Phi$.

In particular, if the context l, r provides no information to disambiguate an ambiguous expression $e \in E - \Phi$, it is natural to expect $[l,e,r] = e$. In this sense, $[l,e,r]$ allows for *lazy* disambiguation, the strategy being to *wait* for an extension l', r' of the context l, r that would disambiguate e. But what if no such extension exists? Then it would make sense to *guess* some value from $\delta(e)$. Now, having introduced non-determinism, the present paper goes whole hog, by considering a "total" disambiguation strategy in which an unambiguous formula can be extracted from $[l,e,r]$ by guessing an extension r' of r for which $[l,e,r'] \in \Phi$. To keep things from getting out of hand, these guesses are recorded in states, structured by a labeled transition system, with a modal logic.

1.1. Transitions disambiguating expressions. More formally, let us assume that the set E of expressions is closed under a concatenation connective ; and write ee' for $e;e'$. To describe cases of deadlock mentioned above in disambiguating an expression e, with l occuring to its left and r to its right, let us define e to be l,r-*stuck* if for every $r' \in E$, $[l,e,r] = [l,e,rr']$. We will assume that

(A0) for all $l, r \in E$ and $\varphi \in \Phi$, $[l,\varphi,r] = \varphi$

from which it follows that every $\varphi \in \Phi$ is l,r-stuck (for all $l,r \in E$). Next, let us weaken the function $[\cdot,\cdot,\cdot]$ into a relation $D(l,e,r,\varphi) \subseteq E \times E \times E \times \Phi$, given by

$$\begin{aligned} D(l,e,r,\varphi) \quad \text{iff} \quad & [l,e,r] = \varphi \ \text{ or} \\ & (e \text{ is } l,r\text{-stuck and } \varphi \in \delta([l,e,r])) \ . \end{aligned}$$

Note that the first disjunct would be subsumed by the second (and can therefore be eliminated) if $\delta(\varphi) = \{\varphi\}$ for every $\varphi \in \Phi$ (complementing (A0)), and $[\cdot,\cdot,\cdot]$ is *right-monotonic* in the sense that for all $l,e,r \in E$,

$$[l,e,r] \in \Phi \quad \text{implies} \quad (\forall r' \in E)\ [l,e,r] = [l,e,rr'] \ .$$

For now, however, let us refrain from making such potentially problematic assumptions about $[\cdot,\cdot,\cdot]$.

Next, let us pair up sequences of expressions in E with unambiguous formulas in Φ disambiguating those expressions as follows. Let us call a finite sequence $\varphi_1 \cdots \varphi_n \in \Phi^*$ a *fully disambiguated alternative (fda) to* a sequence $e_1 \cdots e_m$ in E^* if $n = m$ and there is some $\overline{r} \in E^*$ such that for every $i \in \{1, \ldots, n\}$,

$$D(e_1 \cdots e_{i-1},\ e_i,\ e_{i+1} \cdots e_n \overline{r},\ \varphi_i) \ ,$$

where the empty sequence of expressions in E is, by convention, identified with a distinguished element $\top$ in Φ ($\subseteq E$), and the parenthesization in $e_1 e_2 \cdots e_n$ is to the left:

$$(\cdots (e_1 e_2) \cdots e_n).$$

Note that the non-determinism in fda's arises not only through δ (via D), but also through guessing a continuation $\overline{r}$. Now, let State be the set of all pairs $(\overline{e}, \overline{\varphi})$ such that $\overline{\varphi}$ is an fda to $\overline{e}$. These states undergo transitions labeled by expressions $e \in E$ as follows:

$$\begin{aligned} (\overline{e},\overline{\varphi}) \xrightarrow{e} (\overline{e'},\overline{\varphi'}) \quad \text{iff} \quad & \overline{e'} = \overline{e}e \text{ and } (\exists \varphi)\ \overline{\varphi'} = \overline{\varphi}\varphi \\ & [\text{and } (\overline{e},\overline{\varphi}), (\overline{e'},\overline{\varphi'}) \in \mathsf{State}] \ , \end{aligned}$$

as well as an equivalence relation

$$\begin{aligned} (\overline{e},\overline{\varphi}) \sim (\overline{e'},\overline{\varphi'}) \quad \text{iff} \quad & \overline{e'} = \overline{e} \\ & [\text{and } (\overline{e},\overline{\varphi}), (\overline{e'},\overline{\varphi'}) \in \mathsf{State}] \ , \end{aligned}$$

supporting a jump to a different interpretation $\overline{\varphi'}$ of $\overline{e}$. In addition, some "logic" can be injected through a logical consequence relation $\rhd$ on Φ — i.e., a relation on $\Phi^* \times \Phi$, capturing the intuition that $\varphi_1 \cdots \varphi_n \rhd \varphi$ iff φ is a consequence of $\varphi_1, \ldots, \varphi_n$. Further assumptions are introduced below, as needed to push certain lines of investigation through.[1]

[1] As to exactly what end, the reader familiar with *Discourse Representation Theory* (DRT, Kamp and Reyle [**5**]) may, for orientation, wish to identify Φ with the set of DRSs, and $E - \Phi$ with a set of English sentences (or perhaps some more formal system of underspecified representations). The transitions $\xrightarrow{e}$ can then be seen as describing a DRS

1.2. Contents of the present paper. This paper explores a certain (meta-)strategy for non-determi-nistically processing ambiguous expressions, with particular attention to relating states, subject to transitions defined from possibly different choices, $\delta, [\cdot,\cdot,\cdot], \rhd$ and $\delta', [\cdot,\cdot,\cdot]', \rhd'$, of primitives (the first two ingredients of which, δ and $[\cdot,\cdot,\cdot]$, are particular disambiguation functions). The leap to modal logic in §2 is not surprising, given well-known modal characterizations of the notion of *bisimulation* equivalence (Park [**6**], Hennessy and Milner [**4**]), recalled in §2.2. This "logical" shift brings with it questions about valid formulas, investigated in §3, where a decidability result is established. Directions for further work are discussed in §4.

2. A language for disambiguating transitions

When are two states of the form outlined in §1.1 "equivalent"? The present section considers two approaches that are shown to converge, the first of which builds on a formal (modal) language.

2.1. The modal language $\mathcal{L}(E,\Phi)$. Let $\mathcal{L}(E,\Phi)$, or $\mathcal{L}$ for short, consist of $\mathcal{L}$-formulas A generated from formulas $\varphi \in \Phi$ and expressions $e \in E$ as follows

$$(\mathcal{L}\text{-gen}) \qquad A \quad ::== \quad \varphi \mid \langle e\rangle A \mid \Diamond A \mid \neg A \mid A \wedge B\ .$$

A pair of functions, $\delta : E \to \mathsf{Pow}(\Phi)$ and $[\cdot,\cdot,\cdot] : (E \times E \times E) \to E$, induces, in the manner of §1.1, a set State of states $(\overline{e},\overline{\varphi})$, relative to which, let us define, given a relation $\rhd\ \subseteq \Phi^* \times \Phi$,

$$\begin{array}{rll}
(\overline{e},\overline{\varphi}) \models \varphi & \text{iff} & \overline{\varphi} \rhd \varphi \\
(\overline{e},\overline{\varphi}) \models \langle e\rangle A & \text{iff} & \text{for some } \varphi \text{ such that } (\overline{e}e, \overline{\varphi}\varphi) \in \mathsf{State}, \\
 & & (\overline{e}e,\overline{\varphi}\varphi) \models A \\
(\overline{e},\overline{\varphi}) \models \Diamond A & \text{iff} & \text{for some } \overline{\varphi'} \text{ such that } (\overline{e},\overline{\varphi'}) \in \mathsf{State}, \\
 & & (\overline{e},\overline{\varphi'}) \models A \\
(\overline{e},\overline{\varphi}) \models \neg A & \text{iff} & \text{not } (\overline{e},\overline{\varphi}) \models A \\
(\overline{e},\overline{\varphi}) \models A \wedge B & \text{iff} & (\overline{e},\overline{\varphi}) \models A \text{ and } (\overline{e},\overline{\varphi}) \models B\ .
\end{array}$$

An infinitary extension $\mathcal{L}^\infty$ of $\mathcal{L}$ is easily described by allowing an infinitary conjunction $\bigwedge$ on a set $\mathbf{A}$ of $\mathcal{L}^\infty$-formulas —

$$(\mathcal{L}^\infty\text{-gen}) \qquad A \quad ::== \quad \varphi \mid \langle e\rangle A \mid \Diamond A \mid \neg A \mid \bigwedge \mathbf{A}$$

with

$$(\overline{e},\overline{\varphi}) \models \bigwedge \mathbf{A} \quad \text{iff} \quad (\forall A \in \mathbf{A})\ (\overline{e},\overline{\varphi}) \models A\ .$$

2.2. Transitions underlying $\mathcal{L}$. To relate the language $\mathcal{L}$ to the transitions mentioned in §1.1, let us be a bit more careful to specify when the primitives $\delta, [\cdot,\cdot,\cdot]$ and $\rhd$ are "good." In compliance with standard practice in modal logic, we set apart the first two primitives from the third ($\rhd$). A *g-pair* (for "good" or "generating" pair) is a pair of functions δ and $[\cdot,\cdot,\cdot]$ such that

(A1) for all $\varphi \in \Phi$ and $l, r \in E$, $\delta(\varphi) = \{\varphi\}$ and $[l,\varphi,r] = \varphi$

---

construction algorithm, in which the previous English sentences processed and the DRSs constructed for them are recorded as states. Backtracking from guesses is allowed through the transitions $\sim$. More in §4.2 below.

and

$$\text{(A2)} \quad \text{for all } e \in E, \quad \{[l,e,r] : l,r \in E\} \cap \Phi \ \subseteq \ \delta(e) \quad .$$

(A1) says that unambiguous formulas should be left alone, while (A2) says that for all $l, r \in E$, disambiguation by $[l,\cdot,r]$ sharpens that by δ.[2] For the record, let us define the $\mathcal{L}$*-frame induced* by a g-pair $\delta, [\cdot,\cdot,\cdot]$ to be the pair consisting of the set State (as defined in §1.1) and the transition relation $\rightarrow \subseteq \mathsf{State} \times (E + \{\Diamond\}) \times \mathsf{State}$ satisfying

$$\begin{array}{lll} (\overline{e},\overline{\varphi}) \stackrel{e}{\rightarrow} (\overline{e'},\overline{\varphi'}) & \text{iff} & \overline{e'} = \overline{e}e \text{ and } (\exists \varphi \in \Phi)\ \overline{\varphi'} = \overline{\varphi}\varphi \\ (\overline{e},\overline{\varphi}) \stackrel{\Diamond}{\rightarrow} (\overline{e'},\overline{\varphi'}) & \text{iff} & \overline{e} = \overline{e'} \end{array}$$

for all $(\overline{e},\overline{\varphi}), (\overline{e'},\overline{\varphi'}) \in \mathsf{State}$ (writing $a \stackrel{e}{\rightarrow} b$ to mean (a,e,b) belongs to $\rightarrow$). (The connection with $\mathcal{L}$ should be evident: $\stackrel{e}{\rightarrow}$ supports the modal operator $\langle e \rangle$, while $\stackrel{\Diamond}{\rightarrow}$ supports $\Diamond$.) Now, to ground the language $\mathcal{L}$ in Φ, it remains to equip an $\mathcal{L}$-frame with a *valuation* $V : \Phi \rightarrow \mathsf{Pow}(\mathsf{State})$, defined from a suitable relation $\rhd$ as

$$V(\varphi) \quad = \quad \{(\overline{e},\overline{\varphi}) \in \mathsf{State} : \overline{\varphi} \rhd \varphi\} \ .$$

We shall refer to the resulting structure $\langle \mathsf{State}, \rightarrow, V \rangle$ as the $\mathcal{L}$*-model induced by the g-triple* $\delta, [\cdot,\cdot,\cdot], \rhd$.

Putting the language $\mathcal{L}$ aside for a moment now, let us fix two ternary relations $\rightarrow \subseteq S \times L \times S$ and $\rightarrow' \subseteq S' \times L \times S'$. Let $\cdot^{bf}$ be the "back-and-forth" operator that maps a binary relation $R \subseteq S \times S'$ to the binary relation R^{bf} consisting of all pairs $(a, a') \in S \times S'$ such that for every $l \in L$,

["forth"] for every $b \in S$ such that $a \stackrel{l}{\rightarrow} b$, there exists $b' \in S'$ such that $a' \stackrel{l}{\rightarrow}{}' \, b'$ and bRb'

and

["back"] for every $b' \in S'$ such that $a' \stackrel{l}{\rightarrow}{}' \, b'$, there exists $b \in S$ such that $a \stackrel{l}{\rightarrow} b$ and bRb'.

A *bisimulation between* $\langle S, \rightarrow \rangle$ *and* $\langle S', \rightarrow' \rangle$ is a binary relation $R \subseteq S \times S'$ for which $R \subseteq R^{bf}$. It is well-known that the $\subseteq$-largest bisimulation $\underline{\leftrightarrow}$ between $\langle S, \rightarrow \rangle$ and $\langle S', \rightarrow' \rangle$ exists, and can be calculated (co-inductively) as $\bigcap_\alpha \underline{\leftrightarrow}_\alpha$, where $\underline{\leftrightarrow}_0 = S \times S'$, $\underline{\leftrightarrow}_{\alpha+1} = {\underline{\leftrightarrow}_\alpha}^{bf}$, and $\underline{\leftrightarrow}_\lambda = \bigcap_{\alpha<\lambda} \underline{\leftrightarrow}_\alpha$ for limit ordinals λ.

Returning to $\mathcal{L}$, note that an $\mathcal{L}$-model $\langle \mathsf{State}, \rightarrow, V \rangle$ can be repackaged as $\langle \mathsf{State}, \rightarrow + \rightsquigarrow \rangle$, where $\rightsquigarrow \subseteq \mathsf{State} \times \Phi \times \mathsf{State}$ is defined by

$$(\overline{e},\overline{\varphi}) \stackrel{\varphi}{\rightsquigarrow} (\overline{e'},\overline{\varphi'}) \quad \text{iff} \quad (\overline{e},\overline{\varphi}) = (\overline{e'},\overline{\varphi'}) \in V(\varphi) \ .$$

With this understanding, we can, given $\mathcal{L}$-models M and M' and states $(\overline{e},\overline{\varphi})$ and $(\overline{e'},\overline{\varphi'})$ in M and M' respectively, write $M, (\overline{e},\overline{\varphi}) \ \underline{\leftrightarrow}\ M, (\overline{e'},\overline{\varphi'})$ to mean that the largest bisimulation between M and M' relates $(\overline{e},\overline{\varphi})$ to $(\overline{e'},\overline{\varphi'})$. Next, define $M, (\overline{e},\overline{\varphi}) \ \equiv_{\mathcal{L}} \ M', (\overline{e'},\overline{\varphi'})$ to hold if for all $A \in \mathcal{L}$,

$$(\overline{e},\overline{\varphi}) \models_M A \quad \text{iff} \quad (\overline{e'},\overline{\varphi'}) \models_{M'} A \ .$$

Similarly, for $\equiv_{\mathcal{L}^\infty}$ (replacing $\mathcal{L}$ by $\mathcal{L}^\infty$).

[2]The possibility that an expression might get stuck in the sense of §1.1 precludes a straightforward converse to (A2) for cases other than in (A1).

Theorem 1. *Let* $M = \langle \mathsf{State}, \rightarrow, V \rangle$ *and* $M' = \langle \mathsf{State}', \rightarrow', V' \rangle$ *be the* $\mathcal{L}$*-models induced by g-triples* $\delta, [\cdot,\cdot,\cdot], \rhd$ *and* $\delta', [\cdot,\cdot,\cdot]', \rhd'$*, respectively. For all* $(\overline{e}, \overline{\varphi}) \in \mathsf{State}$ *and* $(\overline{e'}, \overline{\varphi'}) \in \mathsf{State}'$,

$$M, (\overline{e}, \overline{\varphi}) \underline{\leftrightarrow} M', (\overline{e'}, \overline{\varphi'}) \qquad \text{iff} \qquad M, (\overline{e}, \overline{\varphi}) \equiv_{\mathcal{L}^\infty} M', (\overline{e'}, \overline{\varphi'}) \ .$$

If for every $e \in E$*,* $\delta(e)$ *and* $\delta'(e)$ *are finite sets, then the equivalence holds with* $\mathcal{L}^\infty$ *replaced by* $\mathcal{L}$.

Proof. A direct consequence of $\underline{\leftrightarrow} = \bigcap_\alpha \underline{\leftrightarrow}_\alpha$. The additional assumption that $\delta(e) \cup \delta'(e)$ is finite for every $e \in E$ implies, by (A2), that $\underline{\leftrightarrow} = \underline{\leftrightarrow}_\omega$. ⊣

3. $\mathcal{L}$-VALIDITIES UNDER MINIMAL ASSUMPTIONS ON E

The present section is concerned with capturing $\mathcal{L}$-formulas A that are $\mathcal{L}$*-valid* in the sense that for any g-triple, A holds at every state of the $\mathcal{L}$-model induced by that triple. Given a specific relation $\rhd$ of interest, it is natural to specialize the notion of validity to that choice, and call A $\mathcal{L}(\rhd)$*-valid* if for every g-pair $\delta, [\cdot,\cdot,\cdot]$, A holds at every state of the $\mathcal{L}$-model M induced by the g-triple $\delta, [\cdot,\cdot,\cdot], \rhd$. In this connection, it is useful to state a few rules sound for many choices of $\rhd$.

$$\text{(Ref)} \quad \frac{}{\varphi \rhd \varphi} \qquad\qquad \text{(Weak)} \quad \frac{\overline{\varphi} \rhd \varphi}{\overline{\varphi}\psi \rhd \varphi} \quad \frac{\overline{\varphi} \rhd \varphi}{\psi\overline{\varphi} \rhd \varphi}$$

$$\text{(Cut)} \quad \frac{\overline{\varphi} \rhd \psi \qquad\qquad \overline{\varphi}\psi \rhd \varphi}{\overline{\varphi} \rhd \varphi}$$

The rules (Weak) and (Cut) are formulated in a slightly non-standard way[3] to simplify the proof of the following encoding of $\rhd$ into $\mathcal{L}$ via $\wedge$ and $\neg$.

Proposition 2. *If* $\rhd$ *satisfies (Ref), (Weak) and (Cut), then for all* $\varphi_1, \ldots, \varphi_n, \varphi \in \Phi$,

$$\varphi_1 \cdots \varphi_n \rhd \varphi \qquad \text{iff} \qquad (\varphi_1 \wedge \cdots \wedge \varphi_n) \supset \varphi \ \textit{is}\ \mathcal{L}(\rhd)\textit{-valid}$$

(where, as usual, $A \supset B$ *abbreviates* $\neg(A \wedge \neg B)$*).*

Proof. The direction $\Rightarrow$ follows from repeated applications of (Cut) and both sides of (Weak). As for $\Leftarrow$, check satisfaction at the state $(\varphi_1 \cdots \varphi_n, \varphi_1 \cdots \varphi_n)$, appealing to (Ref) and both sides of (Weak). ⊣.

Proposition 2 implies that the complexity of $\mathcal{L}(\rhd)$-validities could not possibly be less than that of $\rhd$, if the rules (Ref), (Weak) and (Cut) are sound in $\rhd$. (And this is to say nothing of modal $\mathcal{L}$-formulas that follow from (Ref), (Weak), and (Cut); more in §3.1 below.) A general decidability result for $\mathcal{L}(\rhd)$-validities is thus out of the question, unless the notion of computability is relativized to an oracle for $\rhd$ (in uncomputable cases such as that of first-order logic) — throwing in, whenever $\varphi_1 \cdots \varphi_n \rhd \varphi$, the $\mathcal{L}$-formula $(\varphi_1 \wedge \cdots \wedge \varphi_n) \supset \varphi$ as an axiom into a proof system for $\mathcal{L}(\rhd)$-validities (assuming $\rhd$ verifies (Ref), (Weak) and (Cut)).

[3] (Ref), (Weak) and (Cut) together imply contraction and exchange

$$\text{(Con)} \quad \frac{\overline{\varphi}\varphi\varphi \rhd \psi}{\overline{\varphi}\varphi \rhd \psi} \qquad\qquad \text{(Exc)} \quad \frac{\overline{\varphi}\varphi\psi\overline{\psi} \rhd \varphi'}{\overline{\varphi}\psi\varphi\overline{\psi} \rhd \varphi'}$$

— which may seem surprising until one notices that not only is an application of (Exc) implicit in doubling (Weak), but so is an application of (Con) in halving the conclusion of (Cut) from $\overline{\varphi}\,\overline{\varphi} \rhd \varphi$ to $\overline{\varphi} \rhd \varphi$.

For the moment, let us set our sights on $\mathcal{L}$-validities rather than on some fixed choice of $\rhd$, while at the same time being sensitive to the sort of consequence relation $\rhd$ that we work with. More specifically, we will attack the problem of a completeness theorem for $\mathcal{L}$-validities, by building models for an $\mathcal{L}$-formula A based on relations $\rhd$ familiar from ordinary propositional logic, except that the internal structure of every formula $\varphi \in \Phi - \{\top\}$ occurring in A is abstracted away. Notice that all formulas in Φ (however complex) are atomic relative to the generation ($\mathcal{L}$-gen) of $\mathcal{L}$-formulas.[4] In particular, it is implicit in ($\mathcal{L}$-gen) that for all $A, B \in \mathcal{L}$, none of $\langle e\rangle A, \Diamond A, \neg A$ and $A \wedge B$ is in E. Whatever connectives E or Φ has ought not to be confused with these. If we require, however, that the relation $\rhd$ in a g-triple satisfy $\overline{\varphi} \rhd \top$ (for every $\overline{\varphi} \in \Phi^*$, where $\top$ is the formula in Φ mentioned previously as, by convention, the representative in E of the empty sequence with respect to ;), then $\top$ is $\mathcal{L}$-valid. Throughout this section, our definition of a g-triple will be amended to incorporate the requirement

(A3) the rules (Ref), (Weak) and (Cut) are sound in $\rhd$, as is

$$\overline{\emptyset \rhd \top} \ .$$

3.1. A proof system for $\mathcal{L}$. It is easy enough to list a number of proof principles sound for $\mathcal{L}$-models:

(i) tautologies (relative to $\wedge$ and $\neg$)

(ii) the rule of modus ponens

(iii) the rule of "uniform substitution" (of arbitrary formulas for atomic formulas)

(iv) for every $e \in E$, the normal logic K for $\langle e\rangle$, plus for every $\varphi \in \Phi$,

$$\langle\varphi\rangle A \quad \equiv \quad [\varphi]A$$

(where, as usual, $A \equiv B$ abbreviates $(A \supset B) \wedge (B \supset A)$, while $[e]$ and $\Box$ abbreviate $\neg\langle e\rangle\neg$ and $\neg\Diamond\neg$, respectively)

(v) the modal system S5 for $\Diamond$, plus

$$\begin{array}{rcl} \langle e\rangle\Diamond A & \supset & \Diamond\langle e\rangle A \\ \Diamond\langle e\rangle A & \supset & [e]\Diamond A \end{array}$$

(vi) consequences of $\rhd$ under (A3)

$$\begin{array}{rcll} & \top & & \\ \langle\varphi\rangle\varphi & \wedge & [\varphi][e]\varphi & \text{[by (Ref) and (Weak)]} \\ \varphi & \supset & [e]\varphi & \text{[by (Weak)]} \\ (\varphi \wedge \langle\varphi\rangle\psi) & \supset & \psi & \text{[by (Cut)]} \ . \end{array}$$

But what else should we add?

An instructive test for a complete proof system is provided by the notion of an $(n, E_\circ)$-*diagram*, where n is some natural number, and $E_\circ$ is a finite subset of E. This is defined by induction on n.

$n = 0$: A $(0, E_\circ)$-diagram is an $\mathcal{L}$-formula of the form

$$\bigwedge \quad (\Phi_\circ \cup \{\neg\varphi : \varphi \in (\Phi \cap E_\circ) - \Phi_\circ\})$$

for some $\Phi_\circ \subseteq \Phi \cap E_\circ$.

[4] Hence, the care taken in avoiding mention of Φ-connectives in Proposition 2 (and in the rules (Ref), (Weak) and (Cut)).

$n+1$: An $(n+1, E_\circ)$-diagram is an $\mathcal{L}$-formula of the form

$$A \;\wedge\; \bigwedge \{([e]\bigvee \mathbf{A}_e) \;\wedge\; \bigwedge \langle e\rangle \mathbf{A}_e \;: e \in E_\circ\}$$

where A is an $(n, E_\circ)$-diagram and for every e belonging to $E_\circ$, $\mathbf{A}_e$ is a set of $(n, E_\circ)$-diagrams (and it is understood that $\bigvee \emptyset$ is a contradictory $\mathcal{L}$-formula, while $\bigwedge \emptyset$ is $\top$).

By a routine induction on n, it is easy to prove that for every n and finite set $E_\circ \subseteq E$,

$$\bigvee \quad \{A \in \mathcal{L} \;:\; A \text{ is an } (n, E_\circ)\text{-diagram}\}$$

is $\mathcal{L}$-valid. In fact, we can trim this disjunction, by adding a bit more structure to the notion of a diagram, reflecting (A3).

Towards that end, let us assume $E_\circ$ contains $\top$, and define an $(n, E_\circ)$*-path* to be a function from $\bigcup_{i=0}^n E_\circ^i$ to $\mathsf{Pow}(\Phi \cap E_\circ)$ such that for every $\overline{e} \in dom(p)$,

$$p(\overline{e}) \neq \emptyset \quad \text{implies} \quad \top \in p(\overline{e}) \ ,$$

and for every $\overline{e}e \in dom(p)$ and $\varphi \in \Phi \cap E_\circ$,

$$\begin{aligned} p(\overline{e}e) \neq \emptyset \quad &\text{implies} \quad p(\overline{e}) \neq \emptyset \text{ and } p(\overline{e}) \subseteq p(\overline{e}e) \\ p(\overline{e}) \neq \emptyset \quad &\text{implies} \quad p(\overline{e}\varphi) = p(\overline{e}) \cup \{\varphi\} \ . \end{aligned}$$

The intuition is that an $(n, E_\circ)$-path p assigns to every $\overline{e} \in \bigcup_{i=0}^n E_\circ^i$ the set $p(\overline{e})$ of formulas in Φ that hold (along the path) after asserting $\overline{e}$. This is formalized by associating with every $(n, E_\circ)$-path p and $\overline{e} \in dom(p)$ the $\mathcal{L}$-formula

$$A_{p,\overline{e}} \;\;=\;\; \bigwedge \,(p(\overline{e}) \cup \{\neg\varphi : \varphi \in (\Phi \cap E_\circ) - p(\overline{e})\}) \ .$$

Now, for branching, let us introduce the notion of a tree as follows. A set P of $(n, E_\circ)$-paths defines a function t from $\bigcup_{i=0}^n E_\circ^i$ to $\mathsf{Pow}(\mathsf{Pow}(\Phi \cap E_\circ))$ given by

$$t(\overline{e}) \;\;=\;\; \{p(\overline{e}) : p \in P\} \ .$$

If $|t(\emptyset)| = 1$ (i.e., for all $p, p' \in P$, $p(\emptyset) = p'(\emptyset)$), then the function t, so defined, is said to be an $(n, E_\circ)$*-tree*. Let us say an $(n, E_\circ)$-path p is a *path through* t if for all $\overline{e} \in dom(p)$, $p(\overline{e}) \in t(\overline{e})$. For every $\overline{e} = e_1 \cdots e_i \in dom(t) - \{\emptyset\}$, let $A_{t,\overline{e}}$ be the $\mathcal{L}$-formula

$$\begin{aligned} &[e_1]\cdots[e_i] \bigvee \{A_{p,\overline{e}} : p \text{ is a path through } t \text{ and } p(\overline{e}) \neq \emptyset\} \\ &\wedge \bigwedge \{\langle e_1\rangle \cdots \langle e_i\rangle A_{p,\overline{e}} : p \text{ is a path through } t \text{ and } p(\overline{e}) \neq \emptyset\}. \end{aligned}$$

Collecting all such e's together, and writing Φ_t for the unique element of $t(\emptyset)$, let A_t be the $\mathcal{L}$-formula

$$\bigwedge \quad (\Phi_t \cup \{A_{t,\overline{e}} : \overline{e} \in dom(t) - \{\emptyset\}\})$$

encoding t.

Proposition 3. *For every natural number n and finite set $E_\circ \subseteq E$,*

$$\bigvee \quad \{A_t \;:\; t \text{ is an } (n, E_\circ)\text{-tree}\}$$

is $\mathcal{L}$-valid.

Proof. Induct on n. $\dashv$

In addition to (i)-(vi) above, let us throw in all the $\mathcal{L}$-valid formulas mentioned in Proposition 3 as axioms in our proof system,[5] and, as usual, write $\vdash A$ to mean that A can be derived in it. Call A *consistent* if not $\vdash \neg A$.

3.2. Completeness and decidability. Given a consistent $\mathcal{L}$-formula A, how can we construct a g-triple inducing an $\mathcal{L}$-model for A? We have no choice but to set $[l, \varphi, r] = \varphi$, $\delta(\varphi) = \{\varphi\}$ for all $l, r \in E$ and $\varphi \in \Phi$. As A is a finite string, it should not be surprising that we need only worry about a finite set $E_\circ$ of expressions in E (including formulas in Φ) that contains the set $E(A)$ of expressions occurring in A. Accordingly, it will be useful to pick out a "rogue" expression $@ \in E - (\Phi \cup E(A))$, take $\delta(@) = \emptyset$, and then set $[l, e, r] = @$ for all but finitely many choices of arguments $l, e, r \in E$. Just what those choices are will depend on some set s of $\mathcal{L}$-formulas that will be required to meet certain properties, from which it follows that s yields, through a certain recipe, a g-triple inducing an $\mathcal{L}$-model with a state in which A holds. Putting off what that recipe is for now, let us attend to what s is. Call a set s of $\mathcal{L}$-formulas A-*full* if

(i) for every $\mathcal{L}$-formula B, $B \in s$ or $\neg B \in s$

and

(ii) for every finite sequence $B_1, \ldots, B_n$ in s, not $\vdash (B_1 \wedge \cdots \wedge B_n) \supset \neg A$.

It follows that A is an element of every A-full set. Moreover, (as will come as no surprise to the cognoscenti)

Lemma 4. *For every consistent $\mathcal{L}$-formula A, there is an A-full set s.*

Proof (folklore). Construct $s = \bigcup_{n<\omega} s_n$ in stages from an enumeration $A_0, A_1, \ldots$ of $\mathcal{L}^A$-formulas, with $s_0 = \{A\}$, and

$$s_{n+1} \quad = \quad \begin{cases} s_n \cup \{\neg A_n\} & \text{if } \vdash (\bigwedge s_n) \supset \neg A_n \\ s_n \cup \{A_n\} & \text{otherwise .} \end{cases}$$

$\dashv$

Turning next to the aforementioned recipe, we will require the following ingredients, in addition to an A-full set s, and an expression $@ \in E - (\Phi \cup E(A))$. Writing $n_\circ$ for the number of occurrences in A of modal operators of the form $\langle e \rangle$, and $E_\circ$ for $E(A) \cup \{\top\}$, fix

(a) an expression $\hat{e} \in E - (\Phi \cup E(A) \cup \{@\})$
(b) for every $(n_\circ, E_\circ)$-tree t, a distinct formula $\varphi_t \in \Phi - E(A)$
(c) for every $(n_\circ, E_\circ)$-path p, a distinct expression $e_p \in E - E_\circ$.[6]

As long as E and Φ are infinite — an assumption that we can safely take for granted — such choices are possible (owing to the finiteness of $E(A)$). Now, let Φ_0 be the finite subset $(E_\circ \cap \Phi) \cup \{\varphi_t : t \text{ is an } (n_\circ, E_\circ)\text{-tree}\}$ of Φ, and Φ_1 be some subset of Φ containing Φ_0 with cardinality $|\Phi_1| = 2^{|\Phi_0|}$. Then there is a bijection $\tilde{\bigwedge} : \mathsf{Pow}(\Phi_0) \to \Phi_1$ such that for every $\varphi \in \Phi_0$, $\tilde{\bigwedge}\{\varphi\} = \varphi$. Writing

[5] We will not worry about deriving a simple and elegant proof theory here.

[6] Whether or not $e_p \in \Phi$ is immaterial (in contrast to (b), where φ_t *must* be an element of Φ).

$\tilde{\cdot} : \Phi_1 \to \mathsf{Pow}(\Phi_0)$ for the inverse function of $\tilde{\bigwedge}$, define a relation $\rhd \subseteq \Phi^* \times \Phi$ inductively by closing the set

$$\{(\varphi_1 \cdots \varphi_n, \psi) \in \Phi_1^* \times \Phi_1 \ : \ \tilde{\psi} \subseteq \tilde{\varphi_1} \cup \cdots \cup \tilde{\varphi_n}\}$$

under the rules mentioned in (A3).

The idea then is to set for all $l, r \in E$, $(n_\circ, E_\circ)$-trees t and $(n_\circ, E_\circ)$-paths p through t,

$$[\top, \hat{e}, re_p\varphi_t] \quad = \quad \tilde{\bigwedge} \, (p(\emptyset) \cup \{\varphi_t\}) \qquad \text{if } \Diamond A_t \in s \text{ and } p(\emptyset) \neq \emptyset$$

(which is independent of the choice of the path p through t, since $|t(\emptyset)| = 1$) and

$$[\hat{e}l, e, re_p\varphi_t] \quad = \quad \tilde{\bigwedge} \, (p(le) \cup \{\varphi_t\}) \qquad \text{if } \Diamond A_t \in s \text{ and } p(le) \neq \emptyset$$

(which *is* dependent on the choice of p). Note that the conjunction above is for Φ, as opposed to that for $\mathcal{L}$. It is a routine matter to complete the definition of $[\cdot, \cdot, \cdot]$ elsewhere to @, and to arrange δ to satisfy (A2) by setting

$$\delta(e) \quad = \quad \Phi \cap \{[l, e, r] \ : \ l, r \in E\} \ .$$

Observe that this recipe makes sense, whether s is A-full or not, and indeed with A replaced by $n_\circ$ and $E_\circ$. But to simplify notation and focus on the most prominent ingredients, we will refer to the $\mathcal{L}$-model so defined as one *A-induced by s*.

Next, leaving this $\mathcal{L}$-model aside for the moment, let

(i) $T(n_\circ, E_\circ)$ be the set of all $(n_\circ, E_\circ)$-trees
(ii) for every $T \subseteq T(n_\circ, E_\circ)$, A^T be the $\mathcal{L}$-formula

$$(\bigwedge_{t \in T} \Diamond A_t) \quad \wedge \quad (\Box \bigvee_{t \in T} A_t)$$

encoding the "forest" T (where A_t is as defined in §3.1)

and

(iii) $\mathcal{L}[n_\circ, E_\circ]$ be the set of $\mathcal{L}$-formulas B with at most $n_\circ$ occurrences of modal operators $\langle e \rangle$, and where all occurrences of expressions $e \in E$ (including $\varphi \in \Phi$) belong to $E_\circ$.

Lemma 5. *For every $T \subseteq T(n_\circ, E_\circ)$, $t \in T$ and $B \in \mathcal{L}[n_\circ, E_\circ]$, either $(A^T \wedge A_t) \supset B$ is $\mathcal{L}$-valid or $(A^T \wedge A_t) \supset \neg B$ is $\mathcal{L}$-valid (or both).*

Proof. By induction on B, noting that the reason for passing to forests is to take account of $\Diamond$. ⊣

Returning now to the $\mathcal{L}$-model A-induced by s, we have

Lemma 6. *Given an A-full set s, A holds at a state in the $\mathcal{L}$-model A-induced by s.*

Proof. Let $T_s = \{t : t \in T(n_\circ, E_\circ) \text{ and } \Diamond A_t \in s\}$. $T_s \neq \emptyset$ by Proposition 3 and the A-fullness of s. So, pick a $t \in T_s$ and a path p through t, noting that

$$(\hat{e}, [\top, \hat{e}, e_p\varphi_t]) \quad \models \quad A^{T_s} \wedge A_t \ .$$

It follows by condition (ii) of A-fullness and Lemma 5 that

$$(\hat{e}, [\top, \hat{e}, e_p\varphi_t]) \quad \models \quad A \ .$$

⊣

An immediate consequence of Lemmas 4 and 6 is

Theorem 7. *For every $\mathcal{L}$-formula A, $\vdash A$ iff A is $\mathcal{L}$-valid.*

Moreover, $\mathcal{L}$ enjoys the so-called finite model property (where the finite models amount to the formulas $A^T \wedge A_t$ mentioned in Lemma 5), whence

Corollary 8. *The set of $\mathcal{L}$-valid formulas is decidable.*

Can we establish Theorem 7 and Corollary 8, with $\mathcal{L}$-validity replaced by $\mathcal{L}(\rhd)$-validity? By Proposition 2, our notion of proof must be beefed up to derive the $\mathcal{L}$-formulas

$$\varphi_1 \wedge \cdots \wedge \varphi_n \quad \supset \quad \varphi$$

whenever $\varphi_1 \cdots \varphi_n \rhd \varphi$, and our notion of computation equip-ped with an oracle for $\rhd$. Although I have yet to check all the details, I believe the argument above for Theorem 7 and Corollary 8 relativizes, provided we add to (A3) the assumption that Φ is closed under a connective $\tilde{\bigwedge}$ that behaves as conjunction (relative to $\rhd$).

4. Discussion

Although the subject of the present work is ambiguity, it is fair to say that the set $E - \Phi$ of ambiguous expressions has been kept largely in the background, and no mention made of the internal structure of these expressions. The attention instead has been on the language $\mathcal{L}$, which is perhaps most naturally construed as a metalanguage, the connectives of which should be distinguished from any within E or Φ. It is true enough that, under (A3), the formula $\top \in \Phi$ functions also as a valid $\mathcal{L}$-formula, and that the difference between $\mathcal{L}$-conjunction $\wedge$ and Φ-conjunction $\tilde{\wedge}$ (assuming Φ is closed under such a connective) is blurred by $\mathcal{L}$-axioms such as

$$\varphi \wedge \varphi' \quad \equiv \quad \varphi \,\tilde{\wedge}\, \varphi' \ .$$

Nonetheless, the distinction must be maintained, as already in the case of a formula $\bot \in \Phi$ that is "absurd" insofar as for every $\varphi \in \Phi$, $\bot \rhd \varphi$, it is, as an $\mathcal{L}$-formula, perfectly consistent: $(\bot, \bot) \models \bot$. Of course, states at which $\bot$ hold are special — special enough, in fact, to merit being collected in the set

$$\bot\text{-}\mathsf{State} \quad = \quad \{(\overline{e}, \overline{\varphi}) \in \mathsf{State} \ : \ \overline{\varphi} \rhd \bot\} \ ,$$

to which the valuation $V : \Phi \rightarrow \mathsf{Pow}(\mathsf{State})$ can be reduced, assuming

(A4) $\rhd$ is $\bot$-*reducible* in the sense that for every $\varphi \in \Phi$ there is a formula $\varphi^c \in \Phi$ such that for all $\overline{\varphi} \in \Phi^*$,

$$\overline{\varphi} \rhd \varphi \quad \text{iff} \quad \overline{\varphi}\varphi^c \rhd \bot \ .$$

Proposition 9. *If $\rhd$ satisfies (A4), then for every $\mathcal{L}$-model induced by a g-triple containing $\rhd$, and for every state $(\overline{e}, \overline{\varphi})$ of that model,*

$$\begin{aligned} (\overline{e}, \overline{\varphi}) \models \varphi \quad &\textit{iff} \quad (\overline{e}, \overline{\varphi}) \models \langle \varphi^c \rangle \bot \\ &\textit{iff} \quad (\overline{e}\varphi^c, \overline{\varphi}\varphi^c) \in \bot\text{-}\mathsf{State} \end{aligned}$$

for every $\varphi \in \Phi$, where φ^c is its complement stipulated by (A4).

Not only does Proposition 9 simplify an $\mathcal{L}$-model $\langle \mathsf{State}, \rightarrow, V \rangle$ to $\langle \mathsf{State}, \rightarrow, \bot\text{-}\mathsf{State} \rangle$, it also suggests a way of extending the language $\mathcal{L}$ to include not only formulas $\varphi \in \Phi$ but arbitrary expressions $e \in E$ —

($\mathcal{L}_+$-gen) $\quad A \quad ::== \quad e \mid \langle e \rangle A \mid \Diamond A \mid \neg A \mid A \wedge B$

with

$$(\dagger) \qquad (\overline{e}, \overline{\varphi}) \models e \quad \text{iff} \quad (\overline{e}, \overline{\varphi}) \models \langle e^c \rangle \bot \ .$$

Two problems plague (†).

(P1) Whereas the modalities $\langle \varphi \rangle$ and $[\varphi]$ are interchangable (because $\xrightarrow{\varphi}$ is functional), $\langle e \rangle$ and $[e]$ are, in general, not (especially if e is ambiguous).

(P2) Just what is the complement e^c of e?

(P1) raises the question: should we really use $\langle e^c \rangle$ rather than $[e^c]$ in formulating an expression $e \in E$ as an $\mathcal{L}$-formula $\langle e^c \rangle \bot$? Inasmuch as both $\langle e^c \rangle \bot$ and $[e^c]\bot$ are available as $\mathcal{L}$-formulas, it may seem silly to insist on choosing between the two.[7] The point, however, in explicitly introducing expressions $e \in E$ into $\mathcal{L}_+$ is to investigate the problem of logical consequence between ambiguous expressions (in terms of $\mathcal{L}_+$), and for this, some unambiguous choice would be helpful. As for (P2), there is no dismissing the importance of complementation $\cdot^c$, especially for matters such as *presupposition*, for which the so-called *negation test* suggests characterizing the presupposition of e as $e \vee e^c$ (with, as usual, $A \vee B$ abbreviating $\neg(\neg A \wedge \neg B)$). (For this purpose, e^c had better not amount to $\neg e$, or else the presupposition, $e \vee e^c$, of e would always be vacuous, whichever way e were formulated within $\mathcal{L}$.) Having said precious little about E, let us close by peeking inside E, before reflecting on strategies that map E to Φ.

4.1. Ambiguous connectives. In addition to assumptions (A1)-(A4), it is natural also to impose the constraint

$$[l,\ ee'\ ,r] \quad = \quad [l,\ e,\ e'r]\ [le,\ e',\ r]\ ,$$

leading to

$$\langle ee' \rangle A \quad \equiv \quad \langle e \rangle \langle e' \rangle A$$

(assuming the juxtaposition in $\varphi\varphi' \rhd \psi$ can be read ;), and

$$[l, e^c\ , r] \quad = \quad [l, e, r]^c\ .$$

Turning to the function $\delta : E \to \mathsf{Pow}(\Phi)$, which comes into play when an expression e gets l,r-stuck (see §1.1), an explicit handle on δ is provided by assuming E is closed under a binary connective $\bullet$, with

$$\delta(e \bullet e') \quad = \quad \delta(e) \cup \delta(e')$$

and

$$[l, e \bullet e', r] \quad = \quad \begin{cases} [l, e, r] & \text{if } e \prec^{l,r} e' \\ [l, e', r] & \text{if } e' \prec^{l,r} e \\ [l, e, r] \bullet [l, e', r] & \text{otherwise} \end{cases}$$

for some function from pairs $(l, r) \in E \times E$ to asymmetric binary relations $\prec^{l,r}$ on E. For motivation and intuitions, see Fernando [**3**]. In a different direction, ambiguous expressions can also be built through labeling and underspecification (e.g., Reyle [**8**], which is generalized in Bos [**2**], and, even further, in Asher and Fernando [**1**]).

[7] In fact, there are many more quantifiers to choose from than $\exists$ ($\langle e \rangle$) and $\forall$ ($[e]$), a general scheme for which, incorporating context-dependent disambiguation, is given in Fernando [**3**].

In any event, the question arises: what are the consequences in $\mathcal{L}$ of the internal structure of expressions?

4.2. Guessing disambiguations. Just what those consequences in $\mathcal{L}$ are will depend partially on $[\cdot,\cdot,\cdot]$, which might, for instance, be required[8] to be monotonic in the sense that whenever $[l,e,r] \in \Phi$, $[l'l,e,rr'] = [l,e,r]$. Note that the functions $[\cdot,\cdot,\cdot]$ constructed for the proof of completeness in §3 can be safely assumed to be monotonic. Also, δ is of no real significance to the $\mathcal{L}$-model induced, beyond the fact that $\delta(@) = \emptyset$. The crucial mechanism used is that of guessing a continuation $e_p\varphi_t$ to be appended to the sequence $\overline{e}$, which is then subjected to $[\cdot,\cdot,\cdot]$-disambiguation. That guess, $e_p\varphi_t$, "explains" the "abduction" of a fully disambiguated alternative to $\overline{e}$ from $[\cdot,\cdot,\cdot]$ (Peirce [**7**]). It could just as easily have been inserted to the left, rather than the right of $\overline{e}$, had the notion of a fully disambiguated alternative allowed for guessing the past (left), as opposed to the future (right). (Indeed, one might imagine inserting guesses anywhere relative to $\overline{e}$: to the left, right, middle ...) Nevertheless, some notion of direction does seem to be unavoidable in making sense of "real-life" phenomena such as presupposition, and in developing abstract tools such as the language $\mathcal{L}$ described above, the modal operators $\langle e \rangle$ of which look ahead to the future, with $\Diamond$ standing by just in case there is none.

References

[1] N. Asher and T. Fernando. Labeling representations for effective disambiguation. In *Proc. Second International Workshop on Computational Semantics (IWCS II)*. Tilburg, The Netherlands, 1997.

[2] Johan Bos. Predicate logic unplugged. In P. Dekker and M. Stokhof, editors, *Proc. Tenth Amsterdam Colloquium*. ILLC, University of Amsterdam, 1996.

[3] Tim Fernando. Ambiguity under changing contexts. Manuscript, 1996. Preliminary version presented in *Mathematics of Language 4* (Philadelphia, October 1995), and submitted for publication.

[4] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *J. Assoc. Computing Machinery*, 32(1), 1985.

[5] H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht, 1993.

[6] David Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proc. 5th GI Conference*, LNCS 104. Springer-Verlag, Berlin, 1981.

[7] Charles Sanders Peirce. Abduction and induction. In J. Buchler, editor, *Philosophical Writings of Peirce*. Dover Books, New York, 1955.

[8] Uwe Reyle. Dealing with ambiguities by underspecification: construction, representation and deduction. *Journal of Semantics*, 10(2), 1993.

Universität Stuttgart, IMS, Azenbergstrasse 12, D-70174 Stuttgart
E-mail address: fernando@ims.uni-stuttgart.de

[8] For the role of constraints in a co-inductive analysis of compositionality, see Fernando [**3**].