Bericht Nr. 31, 1993

Arbeitspapiere des
Sonderforschungsbereichs 340

ISSN 0947-6954/93

Sprachtheoretische Grundlagen für die Computerlinguistik

Paul John King and Thilo Götz:

Eliminating the Feature Introduction Condition by Modifying Type Inference

Universität Stuttgart Universität Tübingen
IBM Deutschland Informationssysteme GmbH

Eliminating the Feature Introduction Condition by Modifying Type Inference*

Paul John King† and Thilo Götz‡
Seminar für Sprachwissenschaft
Eberhard-Karls-Universität Tübingen
Wilhelmstraße 113
72074 Tübingen
Germany

7th of May, 1993

Since their inception with Kay's FUG[1], feature structures have found applications in fields as diverse as linguistics, knowledge representation and logic programming. As the applications have become ever more sophisticated, so the feature structures have become correspondingly ever more complicated, to the point that some applications require a formalism to regulate the feature structures.

[CARPENTER 1992] presents a logical formalism that comprises an algebra of feature structures and a syntax of descriptions. [CARPENTER 1992] uses feature structures to model partial information about empirical objects, and interprets descriptions as true or false of feature structures. [CARPENTER 1992] then presents a number of extensions to this formalism, and applies some of them to unification-based phrase structure grammars, definite clause programmes, and recursive type constraints.

[CARPENTER 1992] imposes a condition called feature introduction in order to well define a function called type inference. In our paper, we argue that the feature introduction condition is undesirable. We slightly modify [CARPENTER 1992]'s type inference function so that we can well define type inference regardless of the feature introduction condition, and we show that if the feature introduction condition holds then our modified type inference function reduces to [CARPENTER 1992]'s original.

We structure our paper as follows. Section 1 retrospects, expands upon and illustrates those parts of [CARPENTER 1992] relevant to our paper, section 2 criticises the feature introduction condition, and section 3 presents our type inference function. Appendix A contains the standard mathematical definitions and results needed for our paper.

---

*The research presented in this paper was sponsored by Teilprojekt B4 "Constraints on Grammar for Efficient Generation" of the Sonderforschungsbereich 340 "Sprachtheoretische Grundlagen für die Computerlinguistik" of the Deutsche Forschungsgemeinschaft. The authors wish to thank Dale Gerdemann and Bob Carpenter for their help and advice. Of course, the authors are responsible for all remaining errors.

†Email: king@sns.neuphilologie.uni-tuebingen.de
‡Email: tg@sns.neuphilologie.uni-tuebingen.de

[1] Functional unification grammar. See [KAY 1979] and [KAY 1984].

1 Preliminaries

In this section, we do three things. Firstly, we review those parts of [CARPENTER 1992] that are germane to our paper. Secondly, some of [CARPENTER 1992]'s definitions are insufficiently formal for use in some of our proofs, and we provide more formal alternatives. Finally, we use an informal semantics distilled from those of [SMOLKA 1988], [KING 1989] and [POLLARD AND MOSHIER 1990] to illustrate [CARPENTER 1992]'s definitions.

1.1 Inheritance Hierarchies

1: Definition (inheritance hierarchy)[2]

An inheritance hierarchy is a pair $\langle \mathsf{Type}, \sqsubseteq \rangle$ *such that*

$\langle \mathsf{Type}, \sqsubseteq \rangle$ *is a finite bounded complete partial order.*

For facility, we henceforth assume that $\langle \mathsf{Type}, \sqsubseteq \rangle$ is an inheritance hierarchy.

A type is a symbol that one can interpret as true or false of an object. For example, one can interpret type *married* as true of each married person, type *single* as true of each unmarried person, and type *person* as true of each person. We say that object u is of type t iff t is interpreted as true of u. Thus, Paul John King is of type *married* since Paul is married to Jennifer, Thilo Götz is of type *single* since Thilo is (as yet!) unmarried, and both Paul and Thilo are of type *person* since both Paul and Thilo are people. We call the set of objects of which an interpreted type is true the denotation of that type. Thus, Paul is in the denotation of *married*, Thilo is in the denotation of *single*, and both are in the denotation of *person*. Type is a set of types. A consequence of definition 1 is that there are finitely many types.

One often restricts the interpretation of two types so that the denotation of the first contains the denotation of the second. For example, the denotation of *person* contains the denotation of *married* since each married person is a person, and the denotation of *person* contains the denotation of *single* since each unmarried person is a person. We say that type t_1 subsumes type t_2 iff the denotation of t_1 contains the denotation of t_2. Thus, *person* subsumes both *married* and *single*. $t_1 \sqsubseteq t_2$ iff type t_1 subsumes type t_2.

Clearly, for each type t,

t subsumes t.

Equally clearly, for all types t_1, t_2 and t_3,

t_1 subsumes t_2 and t_2 subsumes t_3
$\Longrightarrow$ the denotation of t_1 contains the denotation of t_2, and
the denotation of t_2 contains the denotation of t_3
$\Longrightarrow$ the denotation of t_1 contains the denotation of t_3
$\Longrightarrow$ t_1 subsumes t_3.

Two consequences of definition 1 are that for all types t_1 and t_2,

if t_1 and t_2 subsume each other then $t_1 = t_2$,

and for all types t_1 and t_2,

if t_1 and t_2 both subsume some common type
then there is some type t such that
for each type t', t_1 and t_2 both subsume t' iff t subsumes t'.

[2] Cf. [CARPENTER 1992, definition 2.1].

1.2 Appropriateness Specifications

2: Definition (appropriateness specification)[3]

An appropriateness specification over $\langle \mathsf{Type}, \sqsubseteq \rangle$ *and finite set* Feat *is a partial function* $Approp : \mathsf{Feat} \times \mathsf{Type} \rightharpoonup \mathsf{Type}$[4] *such that for all* t_1 *and* $t_2 \in \mathsf{Type}$ *and each* $f \in \mathsf{Feat}$,

if $Approp(f, t_1)$ *is defined and* $t_1 \sqsubseteq t_2$

then $Approp(f, t_2)$ *is defined and* $Approp(f, t_1) \sqsubseteq Approp(f, t_2)$.

For facility, we henceforth assume that *Approp* is an appropriateness specification over $\langle \mathsf{Type}, \sqsubseteq \rangle$ and Feat.

A feature is a symbol that one can interpret as a partial function from objects to objects. For example, one can interpret feature SPOUSE as the partial function that maps each person to their husband or wife, if they have one.[5] Feat is a set of features. A consequence of definition 2 is that there are finitely many features.

One often restricts the interpretation of a feature and a type so that the interpretation of the feature is defined on each object of the type. For example, the interpretation of SPOUSE is defined on each object in the denotation of *married* since each married person has a spouse. We say that type t deserves feature f iff the interpretation of f is defined on each object in the denotation of t. Thus, *married* deserves SPOUSE. $Approp(f, t)$ is defined iff type t deserves feature f.

Clearly, for all types t_1 and t_2 and each feature f,

t_1 subsumes t_2 and t_1 deserves f

$\Longrightarrow$ the denotation of t_1 contains the denotation of t_2, and
the interpretation of f is defined on each object in the denotation of t_1

$\Longrightarrow$ the interpretation of f is defined on each object in the denotation of t_2

$\Longrightarrow$ t_2 deserves f.

One often further restricts the interpretation of a feature and two types so that the first type deserves the feature and the interpretation of the feature maps each object of the first type to an object of the second type. For example, *married* deserves SPOUSE and the interpretation of SPOUSE maps each object in the denotation of *married* to an object in the denotation of *married*, since each married person has a spouse and that spouse must be a married person too. We say that type t' is suitable for feature f on type t iff t deserves f and the interpretation of f maps each object in the denotation of t to an object in the denotation of t'. Thus, *married* is suitable for SPOUSE on *married*. $t' \sqsubseteq Approp(f, t)$ iff type t' is suitable for feature f on type t (sic). Thus, $Approp(f, t)$ is the most specific type (the type with the smallest denotation) that is suitable for f on t.

Clearly, for all types t, t_1 and t_2 and each feature f,

t_1 subsumes t_2 and t is suitable for f on t_1

$\Longrightarrow$ the denotation of t_1 contains the denotation of t_2, and
the interpretation of f maps each object in the denotation of t_1 to an object in the denotation of t

$\Longrightarrow$ the interpretation of f maps each object in the denotation of t_2 to an object in the denotation of t

$\Longrightarrow$ t is suitable for f on t_2.

---

[3] Cf. [CARPENTER 1992, definition 6.1].

[4] We write $\theta : X \rightharpoonup Y$ for "θ is a partial function from X to Y".

[5] Solely for the purpose of constructing a simple example, we assume all marriages are monogamous!

Equally clearly, for all types t, t_1 and t_2 and each feature f,

t_1 subsumes t_2 and t_2 is suitable for f on t

$\Longrightarrow$ the denotation of t_1 contains the denotation of t_2, and

the interpretation of f maps each object in the denotation of t to an object in the denotation of t_2

$\Longrightarrow$ the interpretation of f maps each object in the denotation of t to an object in the denotation of t_1.

$\Longrightarrow$ t_1 is suitable for f on t.

1.3 Depicting an Inheritance Hierarchy and an Appropriateness Specification

If t is a type and $\langle f_1, \ldots, f_n \rangle$ is an ordering of the features t deserves then we usually depict that part of *Approp* relevent to t as the matrix M_t, where

$$M_t = \begin{bmatrix} t & \\ f_1 & \mathit{Approp}(f_1, t) \\ \vdots & \vdots \\ f_n & \mathit{Approp}(f_n, t) \end{bmatrix}.$$

We usually depict $\langle \mathsf{Type}, \sqsubseteq \rangle$ and *Approp* as a finite directed graph G, where

M_t is a node in G iff t is a type, and

there is an arc from node M_{t_1} to node M_{t_2} in G iff t_1 subsumes t_2.

For example, figure 1 depicts our example $\langle \mathsf{Type}, \sqsubseteq \rangle$ and *Approp*.

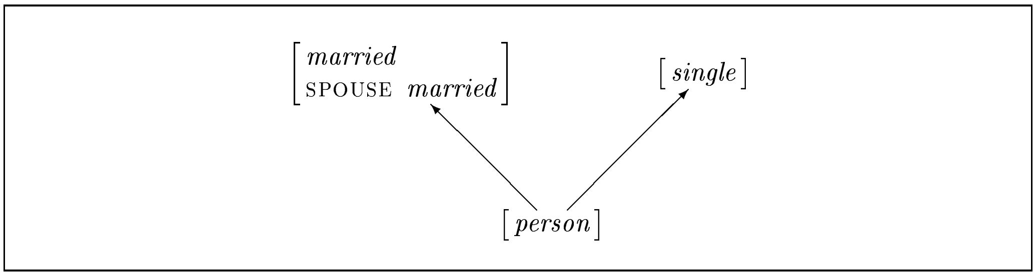


Figure 1

1.4 The Feature Introduction Condition

3: Definition (feature introduction function)[6]

A feature introduction function is a total function $\mathit{Intro} : \mathsf{Feat} \to \mathsf{Type}$[7] *such that for each* $t \in \mathsf{Type}$ *and each* $f \in \mathsf{Feat}$,

$\mathit{Approp}(f, t)$ *is defined iff* $\mathit{Intro}(f) \sqsubseteq t$.

If *Intro* is a feature introduction function then $\mathit{Intro}(f)$ is the least specific type (the type with the largest denotation) that deserves f.

[6] Cf. [Carpenter 1992, definition 6.1].

[7] We write $\theta : X \to Y$ for "θ is a total function from X to Y".

The feature introduction condition of [Carpenter 1992, definition 6.1] deems the existence of a feature introduction function compulsory. However, the purpose of our paper is to show that the feature introduction condition is unnecessary, and we accordingly deem the existence of a feature introduction function optional. We criticise the feature introduction condition in section 2.

1.5 Feature Structures and Subsumption

4: Definition (feature structure)[8]

A feature structure is a quadruple $\langle Q, \overline{q}, \theta, \delta \rangle$ *such that*

Q *is a finite set,*

$\overline{q} \in Q$,

$\theta : Q \rightarrow \mathsf{Type}$,

$\delta : \mathsf{Feat} \times Q \rightharpoonup Q$, *and*

if $q \in Q$ *then for some* $n \in \mathbb{N}$, *some* $q_0, \ldots, q_n \in Q$, *and some* $f_1, \ldots, f_n \in \mathsf{Feat}$

$q_0 = \overline{q}$,

$q_n = q$, *and*

for each $i < n$, $\delta(f_{i+1}, q_i)$ *is defined and* $\delta(f_{i+1}, q_i) = q_{i+1}$.

We write $\boldsymbol{\mathcal{F}}$ for the class of feature structures.

If $F = \langle Q, \overline{q}, \theta, \delta \rangle$ is a feature structure then we usually depict F as a finite and rooted directed graph with labelled nodes and edges, where

q is a node iff $q \in Q$

$\overline{q}$ is the root node

type t labels node q iff $\theta(q) = t$, and

feature f labels an arc from node q to node q' iff $\delta(f, q) = q'$.

We indicate the root node with a double circle. For example, if figure 2 depicts $\langle \mathsf{Type}, \sqsubseteq \rangle$

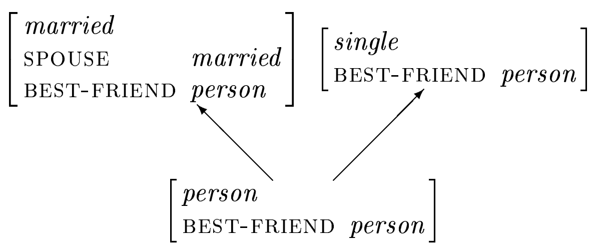


Figure 2

and *Approp* then figure 3 depicts a feature structure.

[Carpenter 1992] applies feature structures to several diverse fields. It is therefore unsurprising that [Carpenter 1992] cannot commit itself to a fixed interpretation of a feature structure. However, it seems fair to say that [Carpenter 1992] uses the modelling convention of [Pollard and Moshier 1990] to interpret a feature structure as information about objects.

[8] Cf. [Carpenter 1992, definition 3.1].

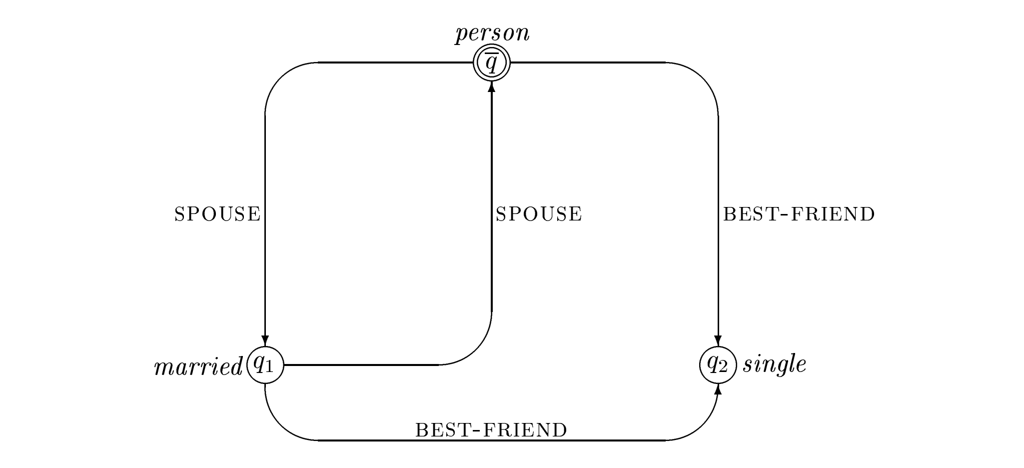


Figure 3

> "the way in which feature structures are used to model partial information about empirical objects is made specific by the so-called *modeling convention*. According to this convention, the nodes of a feature structure are taken to represent objects, and we assume that every node is labeled with a type symbol which represents the most specific conceptual class to which the object is known to belong. Furthermore, arcs come labeled with feature symbols, and an arc between two nodes indicates that the object represented by the source node has a feature, represented by a feature symbol, which has a value represented by the target node. Structure sharing is taken to model identity. That is, two arcs pointing to the same node indicate that the objects represented by the sources of the arcs have identical values for the feature represented by the arcs."
>
> [CARPENTER 1992, pages 34 and 35]

Thus, the interpretation of our example feature structure is the information that

the object is a person,

the object has a spouse and a best friend,

the object's spouse is a married person,

the object's best friend is a single person,

the object's spouse has a spouse and a best friend,

the object's spouse's spouse is the object, and

the object's spouse's best friend is the object's best friend.

Suppose Thilo is the best friend of both Paul and Jennifer. Then the interpretation of our example feature structure is true of Paul and Jennifer, but not of Thilo. (Recall that Paul and Jennifer are a married couple, but Thilo is single.) Stealing terminology from our account of types, we call the set of objects of which an interpreted feature structure is true the denotation of that feature structure. Thus, Paul and Jennifer are in the denotation of our example feature structure, but Thilo is not.

5: Definition (feature structure morphism)[9]

A feature structure morphism from feature structure $\langle Q_1, \overline{q}_1, \theta_1, \delta_1 \rangle$ *to feature structure*

[9] Cf. [CARPENTER 1992, definition 3.4].

$\langle Q_2, \overline{q}_2, \theta_2, \delta_2 \rangle$ *is a total function* $h : Q_1 \to Q_2$ *such that*

$h(\overline{q}_1) = \overline{q}_2$,

for each $q \in Q_1$, $\theta_1(q) \sqsubseteq \theta_2(h(q))$, *and*

for each $f \in \mathsf{Feat}$ *and each* $q \in Q_1$,

if $\delta_1(f, q)$ *is defined*

then $\delta_2(f, h(q))$ *is defined and* $\delta_2(f, h(q)) = h(\delta_1(f, q))$.

6: Definition (subsumption)[9]

$\sqsubseteq$ *is the binary relation on* $\boldsymbol{\mathcal{F}}$ *such that for all* F_1 *and* $F_2 \in \boldsymbol{\mathcal{F}}$,

$F_1 \sqsubseteq F_2$ *iff there is a feature structure morphism from* F_1 *to* F_2.

One can sometimes compare the explicitness of the interpretations of a pair of feature structures. For example, if figure 4 depicts feature structure F and figure 5 depicts feature

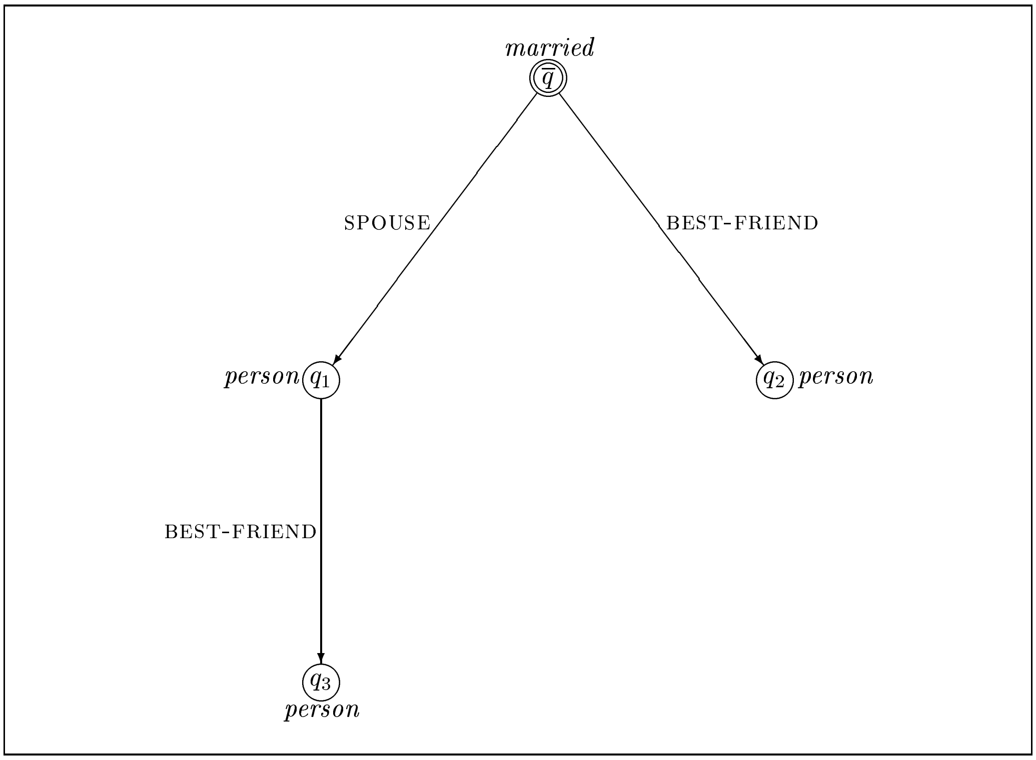


Figure 4

structure F' then the interpretation of F is less explicit than the interpretation of F'. We say that feature structure F_1 subsumes feature structure F_2 iff the interpretation of F_1 is at most as explicit as the interpretation of F_2. Thus, F subsumes F'. $F_1 \sqsubseteq F_2$ iff feature structure F_1 subsumes feature structure F_2.

7: Definition (proper subsumption)

$\sqsubset$ *is the binary relation on* $\boldsymbol{\mathcal{F}}$ *such that for each* F_1 *and* $F_2 \in \boldsymbol{\mathcal{F}}$,

$F_1 \sqsubset F_2$ *iff* $F_1 \sqsubseteq F_2$ *and* $F_2 \not\sqsubseteq F_1$.

If a first feature structure subsumes a second feature structure but the second does not subsume the first then the interpretation of the first is strictly less explicit than the

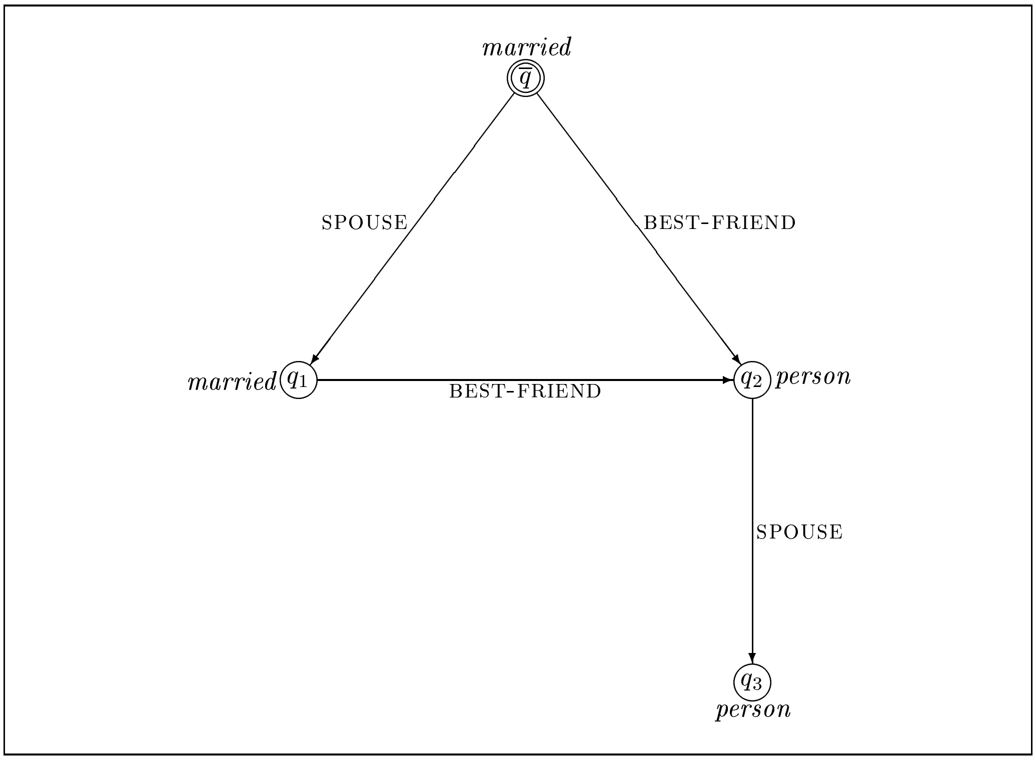


Figure 5

interpretation of the second. We say that feature structure F_1 properly subsumes feature structure F_2 iff F_1 subsumes F_2 but F_2 does not subsume F_1. Thus, F properly subsumes F'. $F_1 \sqsubset F_2$ iff feature structure F_1 properly subsumes feature structure F_2.

8: Proposition[10]

$\langle \boldsymbol{\mathcal{F}}, \sqsubseteq \rangle$ *is a preorder.*

Proof

Firstly, for each $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{F}}$,

the identity function on Q is a feature structure morphism from F to F.

Thus, for each $F \in \boldsymbol{\mathcal{F}}$,

$F \sqsubseteq F$. ...(1)

Secondly, for all F_1, F_2 and $F_3 \in \boldsymbol{\mathcal{F}}$

$F_1 \sqsubseteq F_2$ and $F_2 \sqsubseteq F_3$

$\Longrightarrow$ there is a feature structure morphism h_1 from F_1 to F_2, and

there is a feature structure morphism h_2 from F_2 to F_3

$\Longrightarrow$ $h_2 \circ h_1$ is a feature structure morphism from F_1 to F_3[11]

$\Longrightarrow$ $F_1 \sqsubseteq F_3$. ...(2)

By (1) and (2),

---

[10] Cf. [Carpenter 1992, lemma 3.5].

[11] If $\theta : X \rightharpoonup Y$ and $\varphi : Y \rightharpoonup Z$ then $\varphi \circ \theta : X \rightharpoonup Z$ such that $\varphi \circ \theta(x)$ is defined iff $\theta(x)$ and $\varphi(\theta(x))$ are defined, and if $\varphi \circ \theta(x)$ is defined then $\varphi \circ \theta(x) = \varphi(\theta(x))$.

$\langle \mathcal{F}, \sqsubseteq \rangle$ is a preorder.

□

$\langle \mathcal{F}, \sqsubseteq \rangle$ is not a partial order. For example, if feature structure F is

$$\overset{\textit{person}}{\circledcirc_{q_1}}$$

and feature structure F' is

$$\overset{\textit{person}}{\circledcirc_{q_2}}$$

then

$F \sqsubseteq F'$ and $F' \sqsubseteq F$

but

$F \neq F'$.

9: Definition (alphabetic variance)[12]

$\sim$ is the binary relation on $\mathcal{F}$ such that for each F_1 and $F_2 \in \mathcal{F}$,

$F_1 \sim F_2$ iff $F_1 \sqsubseteq F_2$ and $F_2 \sqsubseteq F_1$.

If two feature structures mutually subsume each other then the interpretations of the two are equally explicit. [Carpenter 1992] says that feature structures F_1 and F_2 are alphabetic variants iff F_1 and F_2 mutually subsume each other. Thus, F and F' are alphabetic variants. $F_1 \sim F_2$ iff feature structures F_1 and F_2 are alphabetic variants.

We complete this subsection with two useful lemmata. The first shows that if two feature structures differ only in the types that label their nodes then the only possible feature structure morphism between them is the identity function. The second shows that though subsumption does not partially order the class of feature structures, it does partially order certain finite sets of feature structures.

10: Lemma

If h is a feature structure morphism from $\langle Q, \overline{q}, \theta_1, \delta \rangle \in \mathcal{F}$ to $\langle Q, \overline{q}, \theta_2, \delta \rangle \in \mathcal{F}$ then

for each $q \in Q$, $h(q) = q$.

Proof

Suppose

$q \in Q$.

Thus, for some $n \in \mathbb{N}$, some $q_0, \ldots, q_n \in Q$, and some $f_1, \ldots, f_n \in \mathsf{Feat}$,

$q_0 = \overline{q}$,

$q_n = q$, and

for each $i < n$, $\delta(f_{i+1}, q_i)$ is defined and $\delta(f_{i+1}, q_i) = q_{i+1}$.

Thus,

$h(q_0) = h(\overline{q}) = \overline{q} = q_0$,

and, for each $i < n$,

$h(q_i) = q_i$

$\Longrightarrow h(q_i) = q_i$, $\delta(f_{i+1}, q_i)$ is defined, $\delta(f_{i+1}, q_i) = q_{i+1}$,

$\delta(f_{i+1}, h(q_i))$ is defined and $\delta(f_{i+1}, h(q_i)) = h(q_{i+1})$

[12] Cf. [Carpenter 1992, definition 3.6].

$\Longrightarrow \delta(f_{i+1}, q_i)$ is defined, $\delta(f_{i+1}, q_i) = q_{i+1}$ and $\delta(f_{i+1}, q_i) = h(q_{i+1})$

$\Longrightarrow h(q_{i+1}) = q_{i+1}$.

Thus, by induction on i, for each $i \leq n$,

$h(q_i) = q_i$.

Thus,

$h(q) = q$.

□

11: Lemma

If $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{F}}$ *then*

$\langle \{ \langle Q, \overline{q}, \theta', \delta \rangle \mid \theta' : Q \to \mathsf{Type} \}, \sqsubseteq \rangle$ *is a finite partial order.*

Proof

Firstly,

Q is finite

and

Type is finite.

Thus,

$\{ \langle Q, \overline{q}, \theta', \delta \rangle \mid \theta' : Q \to \mathsf{Type} \}$ is a finite preorder. ...(1)

Secondly, for all θ_1 and $\theta_2 : Q \to \mathsf{Type}$,

$\langle Q, \overline{q}, \theta_1, \delta \rangle \sqsubseteq \langle Q, \overline{q}, \theta_2, \delta \rangle$

$\Longrightarrow$ there is a feature structure homomorphism from $\langle Q, \overline{q}, \theta_1, \delta \rangle$ to $\langle Q, \overline{q}, \theta_2, \delta \rangle$

$\Longrightarrow$ for some $h : Q \to Q$,

for each $q \in Q$, $h(q) = q$, and by lemma 10

for each $q \in Q$, $\theta_1(q) \sqsubseteq \theta_2(h(q))$

$\Longrightarrow$ for each $q \in Q$, $\theta_1(q) \sqsubseteq \theta_2(q)$.

Thus, for all θ_1 and $\theta_2 : Q \to \mathsf{Type}$,

$\langle Q, \overline{q}, \theta_1, \delta \rangle \sqsubseteq \langle Q, \overline{q}, \theta_2, \delta \rangle$ and $\langle Q, \overline{q}, \theta_2, \delta \rangle \sqsubseteq \langle Q, \overline{q}, \theta_1, \delta \rangle$

$\Longrightarrow$ for each $q \in Q$, $\theta_1(q) \sqsubseteq \theta_2(q)$ and $\theta_2(q) \sqsubseteq \theta_1(q)$

$\Longrightarrow$ for each $q \in Q$, $\theta_1(q) = \theta_2(q)$ since $\langle \mathsf{Type}, \sqsubseteq \rangle$ is a partial order

$\Longrightarrow \theta_1 = \theta_2$

$\Longrightarrow \langle Q, \overline{q}, \theta_1, \delta \rangle = \langle Q, \overline{q}, \theta_2, \delta \rangle$. ...(2)

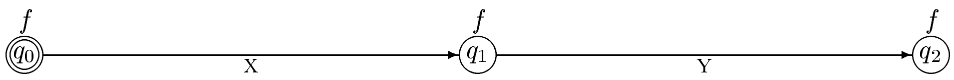
By (1) and (2),

$\langle \{ \langle Q, \overline{q}, \theta', \delta \rangle \mid \theta' : Q \to \mathsf{Type} \}, \sqsubseteq \rangle$ is a finite partial order.

□

1.6 Well-typed Feature Structures and Type Inference

Suppose figure 6 depicts $\langle \mathsf{Type}, \sqsubseteq \rangle$ and *Approp*, feature structure F_1 is



feature structure F_2 is

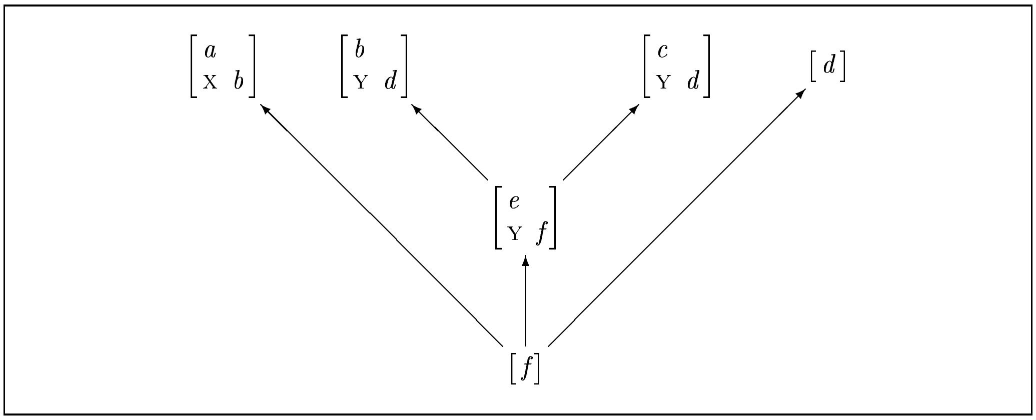
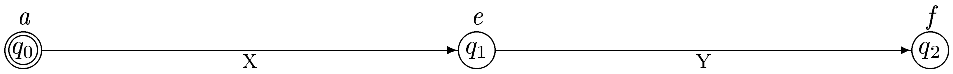
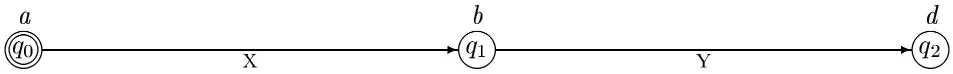


Figure 6



and feature structure F_3 is



Clearly,

$$F_1 \sqsubset F_2 \sqsubset F_3.$$

However, the interpretations of F_1, F_2 and F_3 are equally informative. We can use $\langle \mathsf{Type}, \sqsubseteq \rangle$ and *Approp* to infer all the information in the interpretation of F_2 from the information in the interpretation of F_1, and infer all the information in the interpretation of F_3 from the information in the interpretation of F_2.

12: Definition (semi-well-typed feature structure)

A semi-well-typed feature feature structure is a feature structure $\langle Q, \overline{q}, \theta, \delta \rangle$ *such that for each* $f \in \mathsf{Feat}$ *and* $q \in Q$,

if $\delta(f, q)$ *is defined then* $Approp(f, \theta(q))$ *is defined.*

For example, F_2 and F_3 are semi-well-typed feature structures, whereas F_1 is not. We write $\mathcal{STF}$ for the class of semi-well-typed feature structures.

13: Definition (well-typed feature structure)[13]

A well-typed feature feature structure is a feature structure $\langle Q, \overline{q}, \theta, \delta \rangle$ *such that for each* $f \in \mathsf{Feat}$ *and* $q \in Q$,

if $\delta(f, q)$ *is defined*

then $Approp(f, \theta(q))$ *is defined and* $Approp(f, \theta(q)) \sqsubseteq \theta(\delta(f, q))$.

For example, F_3 is a well-typed feature structure, whereas F_1 and F_2 are not. We write $\mathcal{TF}$ for the class of well-typed feature structures.

The following proposition follows immediately.

14: Proposition

$\mathcal{TF} \subseteq \mathcal{STF}$.

[13] Cf. [Carpenter 1992, definition 6.2].

The proof of [CARPENTER 1992, Theorem 6.3] defines a partial function *TypInf* as the composition of partial functions *TypDom* and *TypRan*, where for each feature structure F,

> *TypDom*(F) is the least explicit semi-well-typed feature structure (up to alphabetic variance) subsumed by F,

and for each semi-well-typed feature structure F',

> *TypRan*(F') is the least explicit well-typed feature structure (up to alphabetic variance) subsumed by F'.

Thus, by proposition 14, for each feature structure F,

> *TypInf*(F) is the least explicit well-typed feature structure (up to alphabetic variance) subsumed by F.

Thus, using our illustrative feature structures,

> $\mathit{TypDom}(F_1) \sim F_2$,
>
> $\mathit{TypRan}(F_2) \sim F_3$, and
>
> $\mathit{TypInf}(F_1) \sim F_3$.

[CARPENTER 1992] calls *TypInf* type inference.

[CARPENTER 1992] defines *TypDom* using a feature introduction function. Therefore, [CARPENTER 1992] imposes the feature introduction condition to well-define *TypDom*. However, the purpose of our paper is to show that the feature introduction condition is unnecessary, and we therefore redefine *TypDom* in section 3. [CARPENTER 1992] defines *TypRan* without using a feature introduction function, and we use this definition. However, to show unequivocally that [CARPENTER 1992] defines *TypRan* without using a feature introduction function, we subdivide the remainder of this subsection into three parts. These subdivisions present [CARPENTER 1992]'s definitions of *TypDom*, *TypRan* and *TypInf* respectively, but we assume the feature introduction condition in the first and last subdivisions only.

1.6.1 A Definition of *TypDom*

Warning! Throughout 1.6.1 we assume the existence of a feature introduction function $\mathit{Intro} : \mathsf{Feat} \to \mathsf{Type}$.

15: Definition (Φ_F)

If $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{F}}$ *then* $\Phi_F : Q \rightharpoonup \mathsf{Type}$ *such that*

> $\Phi_F(q)$ *is defined iff* $\theta(q) \sqcup (\bigsqcup\{\mathit{Intro}(f) \mid \delta(f,q) \text{ is defined}\})$ *is defined, and*
>
> *if* $\Phi_F(q)$ *is defined then* $\Phi_F(q) = \theta(q) \sqcup (\bigsqcup\{\mathit{Intro}(f) \mid \delta(f,q) \text{ is defined}\})$.

16: Definition (*TypDom*)

$\mathit{TypDom} : \boldsymbol{\mathcal{F}} \rightharpoonup \boldsymbol{\mathcal{STF}}$ *such that for each* $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{F}}$,

> *TypDom*(F) *is defined iff for each* $q \in Q$, $\Phi_F(q)$ *is defined, and*
>
> *if TypDom*(F) *is defined then* $\mathit{TypDom}(F) = \langle Q, \overline{q}, \Phi_F, \delta \rangle$.

17: Proposition

TypDom is well-defined.

Proof

TypDom is clearly a partial function with domain $\boldsymbol{\mathcal{F}}$, but it remains to be shown that the range of *TypDom* is $\boldsymbol{\mathcal{STF}}$.

For each $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{F}}$,

$TypDom(F)$ is defined

$\implies TypDom(F)$ is defined and for each $q \in Q$, $\Phi_F(q)$ is defined

$\implies TypDom(F)$ is defined and for each $f \in \mathsf{Feat}$ and each $q \in Q$,

$\delta(f, q)$ is defined

$\implies Intro(f) \sqsubseteq \Phi_F(q)$

$\implies Approp(f, \Phi_F(q))$ is defined

$\implies TypDom(F) \in \boldsymbol{\mathcal{STF}}$.

□

18: Proposition

If $F \in \boldsymbol{\mathcal{F}}$ and $TypDom(F)$ is defined then

$F \sqsubseteq TypDom(F)$.

Proof

Suppose

$F = \langle Q, \overline{q}, \theta, \delta \rangle$, and

h is the identity function on Q.

Thus, for each $q \in Q$,

$\Phi_F(q)$ is defined and $\theta(q) \sqsubseteq \Phi_F(q)$.

Thus,

h is a feature structure morphism from F to $TypDom(F)$.

Thus,

$F \sqsubseteq TypDom(F)$.

□

19: Proposition

If $F \in \boldsymbol{\mathcal{F}}$ and $F' \in \boldsymbol{\mathcal{STF}}$ then

$F \sqsubseteq F'$ *iff* $TypDom(F)$ *is defined and* $TypDom(F) \sqsubseteq F'$.

Proof

Part 1. From left to right.

Suppose

$F = \langle Q, \overline{q}, \theta, \delta \rangle$,

$F' = \langle Q', \overline{q}', \theta', \delta' \rangle$, and

h is a feature structure morphism from F to F'.

Thus, for each $q \in Q$,

$\theta(q) \sqsubseteq \theta'(h(q))$,

and for each $f \in \mathsf{Feat}$ and each $q \in Q$,

$\delta(f, q)$ is defined

$\implies \delta'(f, h(q))$ is defined

$\implies Approp(f, \theta'(h(q)))$ is defined — since $F' \in \boldsymbol{\mathcal{STF}}$

$\implies Intro(f) \sqsubseteq \theta'(h(q))$.

Thus, for each $q \in Q$,

$\Phi_F(q)$ is defined and $\Phi_F(q) \sqsubseteq \theta'(h(q))$.

Thus,

$\quad$ *TypDom*(F) is defined, and

$\quad$ h is a feature structure morphism from *TypDom*(F) to F'.

Thus,

$\quad$ *TypDom*(F) is defined and *TypDom*$(F) \sqsubseteq F'$.

Part 2. From right to left.

$\quad$ *TypDom*(F) is defined and *TypDom*$(F) \sqsubseteq F'$

$\quad$ $\Longrightarrow F \sqsubseteq$ *TypDom*(F) and *TypDom*$(F) \sqsubseteq F'$ $\qquad$ by proposition 18

$\quad$ $\Longrightarrow F \sqsubseteq F'$.

□

1.6.2 A Definition of *TypRan*

20: Definition (π_F)

If $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{STF}}$ *then* $\pi_F : Q \to Pow(\mathsf{Type})$ *such that*

$\quad$ $\pi_F(q) = \{Approp(f, \theta(q')) \mid \delta(f, q')$ *is defined and* $\delta(f, q') = q\}$.

21: Definition (Π_F)

If $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{STF}}$ *then* $\Pi_F : Q \rightharpoonup \mathsf{Type}$ *such that*

$\quad$ $\Pi_F(q)$ *is defined iff* $\theta(q) \sqcup (\bigsqcup \pi_F(q))$ *is defined, and*

$\quad$ *if* $\Pi_F(q)$ *is defined then* $\Pi_F(q) = \theta(q) \sqcup (\bigsqcup \pi_F(q))$.

22: Definition (*TypProg*)

TypProg : $\boldsymbol{\mathcal{STF}} \rightharpoonup \boldsymbol{\mathcal{STF}}$ *such that for each* $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{STF}}$,

$\quad$ *TypProg*(F) *is defined iff for each* $q \in Q$, $\Pi_F(q)$ *is defined, and*

$\quad$ *if TypProg*(F) *is defined then TypProg*$(F) = \langle Q, \overline{q}, \Pi_F, \delta \rangle$.

23: Proposition

TypProg is well-defined.

Proof

TypProg is clearly a partial function with domain $\boldsymbol{\mathcal{STF}}$, but it remains to be shown that the range of *TypProg* is $\boldsymbol{\mathcal{STF}}$.

For each $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{STF}}$,

$\quad$ *TypProg*(F) is defined

$\quad$ $\Longrightarrow$ *TypProg*(F) is defined and for each $q \in Q$, $\Pi_F(q)$ is defined

$\quad$ $\Longrightarrow$ *TypProg*(F) is defined and for each $f \in \mathsf{Feat}$ and each $q \in Q$,

$\qquad\quad$ $\delta(f, q)$ is defined

$\qquad\quad$ $\Longrightarrow Approp(f, \theta(q))$ is defined $\qquad$ since $F \in \boldsymbol{\mathcal{STF}}$

$\qquad\quad$ $\Longrightarrow Approp(f, \theta(q))$ is defined and $\theta(q) \sqsubseteq \Pi_F(q)$

$\qquad\quad$ $\Longrightarrow Approp(f, \Pi_F(q))$ is defined

$\quad$ $\Longrightarrow$ *TypProg*$(F) \in \boldsymbol{\mathcal{STF}}$.

□

24: Proposition

If $F \in \boldsymbol{\mathcal{STF}}$ *and TypProg*(F) *is defined then*

$\quad$ $F \sqsubseteq$ *TypProg*(F).

Proof

Suppose

$F = \langle Q, \overline{q}, \theta, \delta \rangle$, and

h is the identity function on Q.

Thus, for each $q \in Q$,

$\Pi_F(q)$ is defined and $\theta(q) \sqsubseteq \Pi_F(q)$.

Thus,

h is a feature structure morphism form F to $\mathit{TypProg}(F)$.

Thus,

$F \sqsubseteq \mathit{TypProg}(F)$.

□

25: Proposition

If $F \in \mathcal{STF}$ *and* $F' \in \mathcal{TF}$ *then*

$F \sqsubseteq F'$ *iff* $\mathit{TypProg}(F)$ *is defined and* $\mathit{TypProg}(F) \sqsubseteq F'$.

Proof

Part 1. From left to right.

Suppose

$F = \langle Q, \overline{q}, \theta, \delta \rangle$,

$F' = \langle Q', \overline{q}', \theta', \delta' \rangle$, and

h is a feature structure morphism from F to F'.

Thus, for each $q \in Q$,

$\theta(q) \sqsubseteq \theta'(h(q))$,

and for each $f \in \mathsf{Feat}$ and all q and $q' \in Q$,

$\delta(f, q')$ is defined and $\delta(f, q') = q$

$\Longrightarrow$ $\mathit{Approp}(f, \theta(q'))$ is defined, since $F' \in \mathcal{STF}$

$\delta'(f, h(q'))$ is defined, and

$\delta'(f, h(q')) = h(q)$

$\Longrightarrow$ $\mathit{Approp}(f, \theta(q'))$ is defined,

$\mathit{Approp}(f, \theta'(h(q')))$ is defined, since $F' \in \mathcal{TF}$

$\mathit{Approp}(f, \theta(q')) \sqsubseteq \mathit{Approp}(f, \theta'(h(q')))$, and

$\mathit{Approp}(f, \theta'(h(q'))) \sqsubseteq \theta'(h(q))$ since $F' \in \mathcal{TF}$

$\Longrightarrow$ $\mathit{Approp}(f, \theta(q'))$ is defined and $\mathit{Approp}(f, \theta(q')) \sqsubseteq \theta'(h(q))$

Thus, for each $q \in Q$,

$\Pi_F(q)$ is defined and $\Pi_F(q) \sqsubseteq \theta'(h(q'))$.

Thus,

$\mathit{TypProg}(F)$ is defined, and

h is a feature structure morphism from $\mathit{TypProg}(F)$ to F'.

Thus,

$\mathit{TypProg}(F)$ is defined and $\mathit{TypProg}(F) \sqsubseteq F'$.

Part 2. From right to left.

$\mathit{TypProg}(F)$ is defined and $\mathit{TypProg}(F) \sqsubseteq F'$

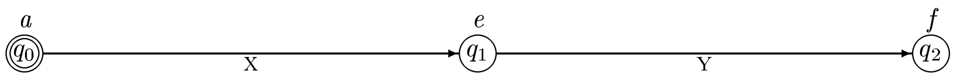
$\Longrightarrow F \sqsubseteq \mathit{TypProg}(F)$ and $\mathit{TypProg}(F) \sqsubseteq F'$ by proposition 24

$\Longrightarrow F \sqsubseteq F'$.

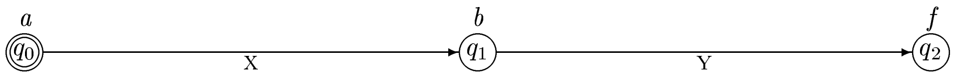
□

The range of *TypProg* is not $\boldsymbol{\mathcal{TF}}$. Suppose F is a semi-well-typed feature structure. Suppose feature f labels an arc from node q_1 to node q_2, type t_1 labels node q_1 and type t_2 labels node q_2 in F. Suppose $\mathit{TypProg}(F)$ is defined. Feature f still labels an arc from node q_1 to node q_2 in $\mathit{TypProg}(F)$. Suppose type t'_1 labels node q_1 and type t'_2 labels node q_2 in $\mathit{TypProg}(F)$. The definition of *TypProg* ensures that type $\mathit{Approp}(f, t_1)$ subsumes type t'_2, but it does not ensure that type $\mathit{Approp}(f, t'_1)$ subsumes type t'_2. Loosely speaking, *TypProg* 'improves' a feature structure by labelling each target node with a type at least as specific as the most specific type suitable for the feature labelling each arc entering that node on the type labelling the source node of that arc *in the original feature structure*. However, a source node can itself be a target node, and by relabelling the target node of a first arc, *TypProg* may relabel the source node of a second arc. Thus, *TypProg* may not label each node with a type at least as specific as the most specific type suitable for the feature labelling each arc entering that node on the type labelling the source node of that arc *in the improved feature structure*.

For example, recall the semi-well-typed feature structure F_2 of page 10. F_2 is



and $\mathit{TypProg}(F_2)$ is



$\mathit{TypProg}(F_2)$ is not well-typed. Type e labels node q_1 in F_2, and *TypProg* certainly ensures that type $\mathit{Approp}(\text{Y}, e)$ subsumes type f. However, type b labels node q_1 in $\mathit{TypProg}(F_2)$, and *TypProg* does not ensure that type $\mathit{Approp}(\text{Y}, b)$ subsumes type f.

If F is a semi-well-typed feature structure then we apply *TypProg* iteratively to F until a fixed point is reached in order to obtain $\mathit{TypRan}(F)$.

26: Lemma

If $F \in \boldsymbol{\mathcal{STF}}$, $\mathit{TypProg}(F)$ is defined and $\mathit{TypProg}(F) = F$ then

$F \in \boldsymbol{\mathcal{TF}}$.

Proof

For each $F = \langle Q, \overline{q}, \theta, \delta\rangle \in \boldsymbol{\mathcal{STF}}$,

$\mathit{TypProg}(F)$ is defined and $\mathit{TypProg}(F) = F$

$\Longrightarrow$ for each $q \in Q$, $\Pi_F(q)$ is defined and $\Pi_F(q) = \theta(q)$

$\Longrightarrow$ for each $f \in \mathsf{Feat}$ and each $q \in Q$,

$\delta(f, q)$ is defined

$\Longrightarrow \mathit{Approp}(f, \theta(q))$ is defined and $\delta(f, q)$ is defined since $F \in \boldsymbol{\mathcal{STF}}$

$\Longrightarrow \mathit{Approp}(f, \theta(q))$ is defined and $\mathit{Approp}(f, \theta(q)) \in \pi_F(\delta(f, q))$

$\Longrightarrow \mathit{Approp}(f, \theta(q))$ is defined, and

$\mathit{Approp}(f, \theta(q)) \sqsubseteq \Pi_F(\delta(f, q))$

$\Longrightarrow \mathit{Approp}(f, \theta(q))$ is defined and $\mathit{Approp}(f, \theta(q)) \sqsubseteq \theta(\delta(f, q))$

$\Longrightarrow F \in \boldsymbol{\mathcal{TF}}$.

□

27: Lemma

If $F \in \boldsymbol{\mathcal{STF}}$ and for each $n \in \mathbb{N}$, $\mathit{TypProg}^n(F)$ is defined[14] *then*

$\langle\{\mathit{TypProg}^n(F) \mid n \in \mathbb{N}\}, \sqsubseteq\rangle$ *has a unique maximal member.*

Proof

By lemmata 11 and 69,

$\langle\{\mathit{TypProg}^n(F) \mid n \in \mathbb{N}\}, \sqsubseteq\rangle$ is an upward Noetherian partial order,

and by proposition 24,

for each $n \in \mathbb{N}$, $\mathit{TypProg}^n(F) \sqsubseteq \mathit{TypProg}^{n+1}(F)$.

Thus, by lemma 70,

$\langle\{\mathit{TypProg}^n(F) \mid n \in \mathbb{N}\}, \sqsubseteq\rangle$ has a unique maximal member.

□

28: Definition (*TypRan*)

$\mathit{TypRan} : \boldsymbol{\mathcal{STF}} \rightharpoonup \boldsymbol{\mathcal{TF}}$ such that

$\mathit{TypRan}(F)$ is defined iff for each $n \in \mathbb{N}$, $\mathit{TypProg}^n(F)$ is defined, and

if $\mathit{TypRan}(F)$ is defined

then $\mathit{TypRan}(F)$ is the maximal member of $\langle\{\mathit{TypProg}^n(F) \mid n \in \mathbb{N}\}, \sqsubseteq\rangle$.

29: Proposition

TypRan is well-defined.

Proof

By lemma 27, *TypRan* is a partial function with domain $\boldsymbol{\mathcal{STF}}$, but it remains to be shown that the range of *TypRan* is $\boldsymbol{\mathcal{TF}}$.

For each $F = \langle Q, \overline{q}, \theta, \delta\rangle \in \boldsymbol{\mathcal{STF}}$,

$\mathit{TypRan}(F)$ is defined

$\Longrightarrow \mathit{TypRan}(F)$ is defined,

$\mathit{TypRan}(F) \in \boldsymbol{\mathcal{STF}}$,

$\mathit{TypProg}(\mathit{TypRan}(F))$ is defined

$\mathit{TypRan}(F) \sqsubseteq \mathit{TypProg}(\mathit{TypRan}(F))$, and — by proposition 24

$\mathit{TypProg}(\mathit{TypRan}(F)) \sqsubseteq \mathit{TypRan}(F)$

since $\mathit{TypRan}(F)$ is the maximal member of $\langle\{\mathit{TypProg}^n(F) \mid n \in \mathbb{N}\}, \sqsubseteq\rangle$

$\Longrightarrow \mathit{TypRan}(F)$ is defined,

$\mathit{TypRan}(F) \in \boldsymbol{\mathcal{STF}}$,

$\mathit{TypProg}(\mathit{TypRan}(F))$ is defined, and

$\mathit{TypProg}(\mathit{TypRan}(F)) = \mathit{TypRan}(F)$ — by lemma 11

$\Longrightarrow \mathit{TypRan}(F) \in \boldsymbol{\mathcal{TF}}$. — by lemma 26

□

30: Proposition

If $F \in \boldsymbol{\mathcal{STF}}$ and $\mathit{TypRan}(F)$ is defined then

$F \sqsubseteq \mathit{TypRan}(F)$.

[14] If $\theta : X \rightharpoonup X$ then θ^0 is the identity function on X, and θ^{n+1} is $\theta \circ \theta^n$.

Proof

Proposition 30 follows immediately from proposition 24 $\square$

31: Proposition

If $F \in \mathcal{STF}$ and $F' \in \mathcal{TF}$ then

$$F \sqsubseteq F' \text{ iff } \mathit{TypRan}(F) \text{ is defined and } \mathit{TypRan}(F) \sqsubseteq F'.$$

Proof

$F \sqsubseteq F'$

$\Longrightarrow$ for each $n \in \mathbb{N}$, $\mathit{TypProg}^n(F)$ is defined and $\mathit{TypProg}^n(F) \sqsubseteq F'$ by proposition 25

$\Longrightarrow$ $\mathit{TypRan}(F)$ is defined and $\mathit{TypRan}(F) \sqsubseteq F'$

and

$\mathit{TypRan}(F)$ is defined and $\mathit{TypRan}(F) \sqsubseteq F'$

$\Longrightarrow F \sqsubseteq \mathit{TypRan}(F)$ and $\mathit{TypRan}(F) \sqsubseteq F'$ by proposition 30

$\Longrightarrow F \sqsubseteq F'$.

$\square$

1.6.3 A Definition of *TypInf*

Warning! Throughout 1.6.3 we assume the existence of a feature introduction function $\mathit{Intro} : \mathsf{Feat} \to \mathsf{Type}$.

32: Definition (*TypInf*)

$\mathit{TypInf} : \mathcal{F} \rightharpoonup \mathcal{TF}$ such that

$$\mathit{TypInf} = \mathit{TypRan} \circ \mathit{TypDom}.$$

33: Proposition

If $F \in \mathcal{F}$ and $\mathit{TypInf}(F)$ is defined then

$$F \sqsubseteq \mathit{TypInf}(F).$$

Proof

For each $F \in \mathcal{F}$,

$\mathit{TypInf}(F)$ is defined

$\Longrightarrow$ $\mathit{TypRan} \circ \mathit{TypDom}(F)$ is defined, and
$\mathit{TypInf}(F) = \mathit{TypRan} \circ \mathit{TypDom}(F)$

$\Longrightarrow$ $\mathit{TypDom}(F)$ is defined,
$\mathit{TypRan}(\mathit{TypDom}(F))$ is defined, and
$\mathit{TypInf}(F) = \mathit{TypRan}(\mathit{TypDom}(F))$

$\Longrightarrow F \sqsubseteq \mathit{TypDom}(F)$, by proposition 18
$\mathit{TypDom}(F) \sqsubseteq \mathit{TypRan}(\mathit{TypDom}(F))$, and by proposition 30
$\mathit{TypInf}(F) = \mathit{TypRan}(\mathit{TypDom}(F))$

$\Longrightarrow F \sqsubseteq \mathit{TypInf}(F)$.

$\square$

34: Proposition

If $F \in \mathcal{F}$ and $F' \in \mathcal{TF}$ then

$$F \sqsubseteq F' \text{ iff } \mathit{TypInf}(F) \text{ is defined and } \mathit{TypInf}(F) \sqsubseteq F'.$$

Proof

$$
\begin{aligned}
& F \sqsubseteq F' \\
\iff & \mathit{TypDom}(F) \text{ is defined and } \mathit{TypDom}(F) \sqsubseteq F' \qquad \text{by propositions 14 and 19} \\
\iff & \mathit{TypRan}(\mathit{TypDom}(F)) \text{ is defined and } \mathit{TypRan}(\mathit{TypDom}(F)) \sqsubseteq F' \\
& \qquad \text{by proposition 31} \\
\iff & \mathit{TypRan} \circ \mathit{TypDom}(F) \text{ is defined and } \mathit{TypRan} \circ \mathit{TypDom}(F) \sqsubseteq F' \\
\iff & \mathit{TypInf}(F) \text{ is defined and } \mathit{TypInf}(F) \sqsubseteq F'.
\end{aligned}
$$

□

2 Criticism

Subsection 1.4 said [CARPENTER 1992] imposes a feature introduction condition that obligates the existence of a feature introduction function $\mathit{Intro} : \mathsf{Feat} \to \mathsf{Type}$ such that for each feature f, $\mathit{Intro}(f)$ is the least specific type that deserves f. [CARPENTER 1992] claims that the feature introduction condition is imposed in order to

> "ensure that type inference ... is well defined"
>
> [CARPENTER 1992, page 86]

and subsection 1.6 showed that [CARPENTER 1992]'s definition of type inference requires the existence of a feature introduction function.

However, to the best of our knowledge, only [CARPENTER 1992] imposes the feature introduction condition, and several popular typed unification formalisms do not meet it. For example, HPSG[15] is a grammar with an underlying typed unification formalism. HPSG uses arbitrary length lists of objects to account for certain aspects of natural language, and the underlying formalism uses the "first/rest" method of the programming languages LISP and Prolog to represent such lists. The "first/rest" method introduces three types, *list*, *empty-list* and *proper-list*, and two features, FIRST and REST. The denotation of *list* is the collection of all lists, the denotation of *empty-list* is the collection of all lists with no members, and the denotation of *proper-list* is the collection of all lists with some members. *list* and *empty-list* deserve no features, but *proper-list* deserves features FIRST and REST. The interpretation of FIRST maps proper list l to the first member of l, and the interpretation of REST maps proper list l to l with its first member removed. Figure 7 depicts this inheritance hierarchy and appropriateness specification.

However, rather than lists of objects of arbitrary type, HPSG uses different kinds of list, where each kind of list comprises objects of a specific type. Thus, for each type τ, HPSG incorporates the fragment in figure 8 into the inheritance hierarchy and appropriateness specification, where the denotation of *list-of-*τ is the collection of all lists whose members are of type τ, and the denotation of *proper-list-of-*τ is the collection of all non empty lists whose members are of type τ. HPSG does not generate list types from list types: if τ is *empty-list*, *list-of-*τ' or *proper-list-of-*τ' then HPSG does not generate *list-of-*τ or *proper-list-of-*τ. Thus, the inheritance hierarchy remains finite.

If figure 9 depicts HPSG's unexpanded inheritance hierarchy and appropriateness specification then figure 10 depicts HPSG's expanded inheritance hierarchy and appropriateness specification.[16] Notice that there is no Intro(FIRST) or Intro(REST). HPSG's underlying

[15] Head-driven phrase structure grammar. See [POLLARD AND SAG 1987] and [POLLARD AND SAG 1993].

[16] This is an extreme simplification of HPSG's inheritance hierarchy and appropriateness specification, but it is sufficient to illustrate our point. See [POLLARD AND SAG 1993] for HPSG's true inheritance

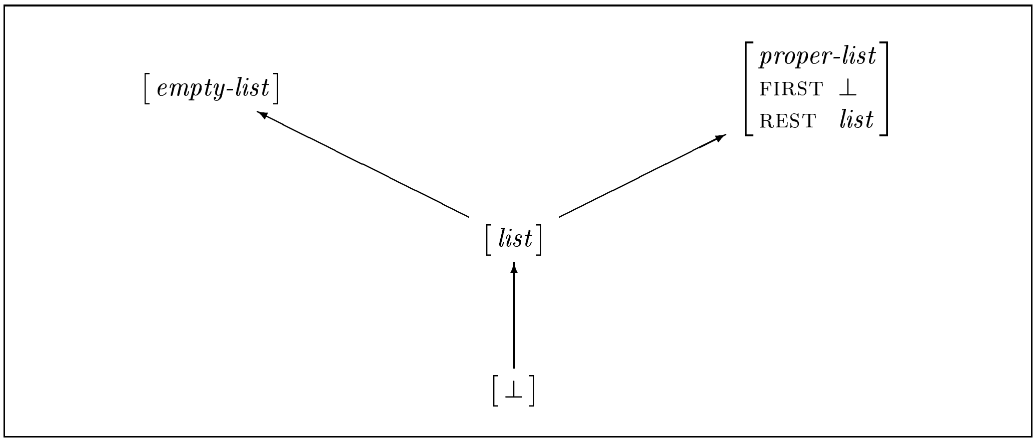


Figure 7

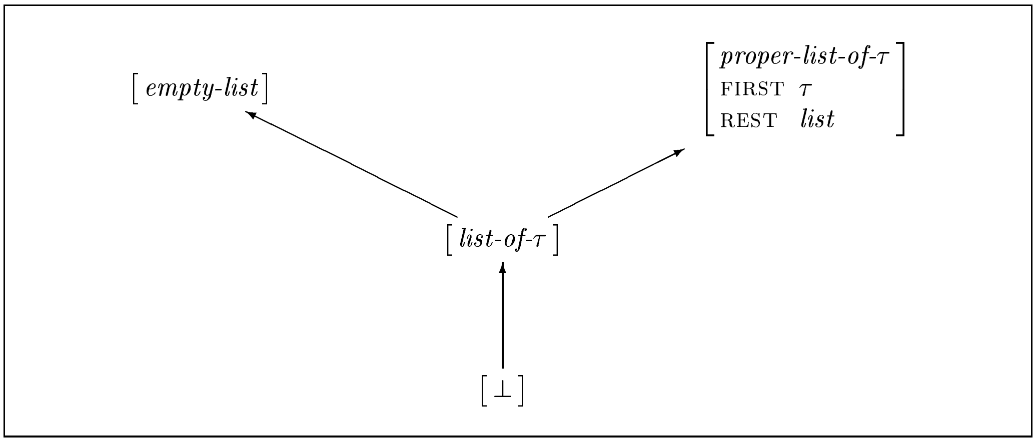


Figure 8

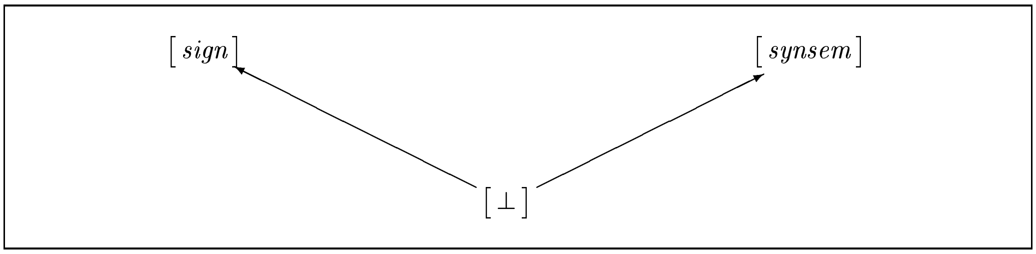


Figure 9

formalism does not meet the feature introduction condition.

There are two obvious ways to amend HPSG's underlying formalism to meet the feature introduction condition. The first amends HPSG's inheritance hierarchy. The second amends HPSG's appropriateness specification.

---

hierarchy and appropriateness specification.

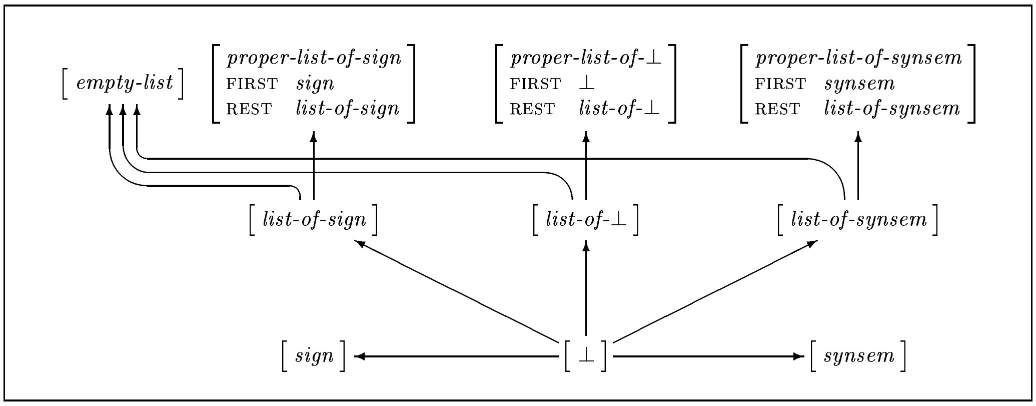


Figure 10

Amend HPSG's inheritance hierarchy to let

$$\textit{proper-list-of-}\bot \sqsubseteq \textit{proper-list-of-sign}$$

and

$$\textit{proper-list-of-}\bot \sqsubseteq \textit{proper-list-of-synsem}$$

To maintain an appropriateness specification, one must also amend HPSG's inheritance hierarchy to let

$$\textit{list-of-}\bot \sqsubseteq \textit{list-of-sign}$$

and

$$\textit{list-of-}\bot \sqsubseteq \textit{list-of-synsem}.$$

Figure 11 depicts HPSG's ammended inheritance hierarchy and appropriateness specifica-

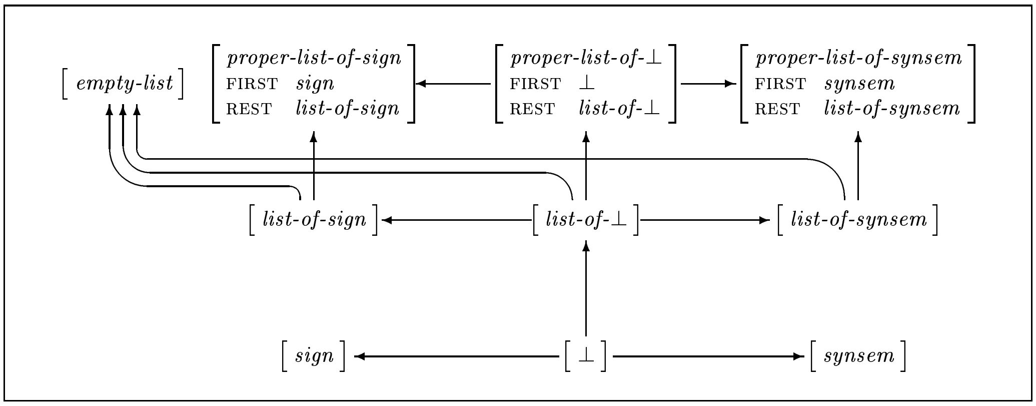


Figure 11

tion. Clearly, HPSG's underlying formalism now meets the feature introduction condition. However, this amendment has unforseen consequences. For example, if figure 12 depicts feature structure F then F is now well-typed, whereas formerly it was not.

Alternatively, ammend HPSG's set of features and appropriateness specification so that for each $\tau \in \{\bot, \textit{sign}, \textit{synsem}\}$, *proper-list-of-*$\tau$ deserve features $\textsc{first}_\tau$ and $\textsc{rest}_\tau$ instead of FIRST and REST. Figure 13 depicts HPSG's reammended inheritance hierarchy and appropriateness specification. Again, HPSG's underlying formalism now meets the feature

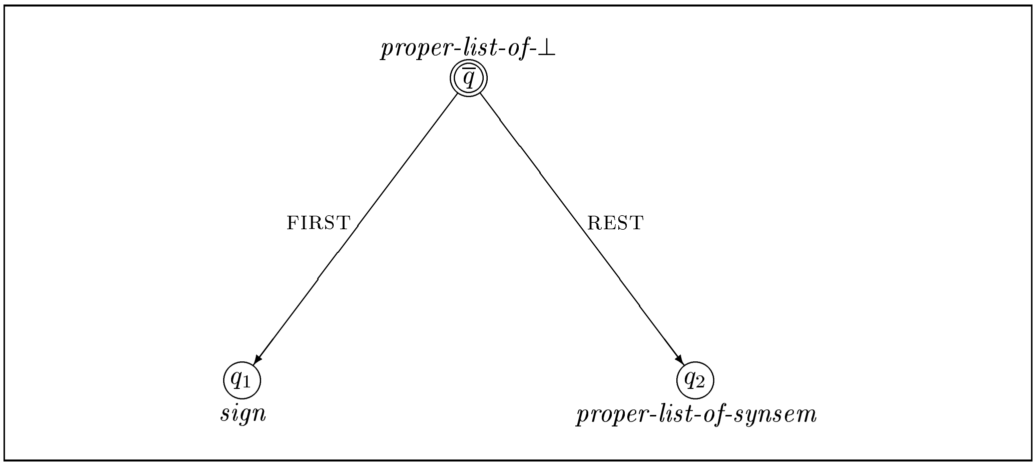


Figure 12

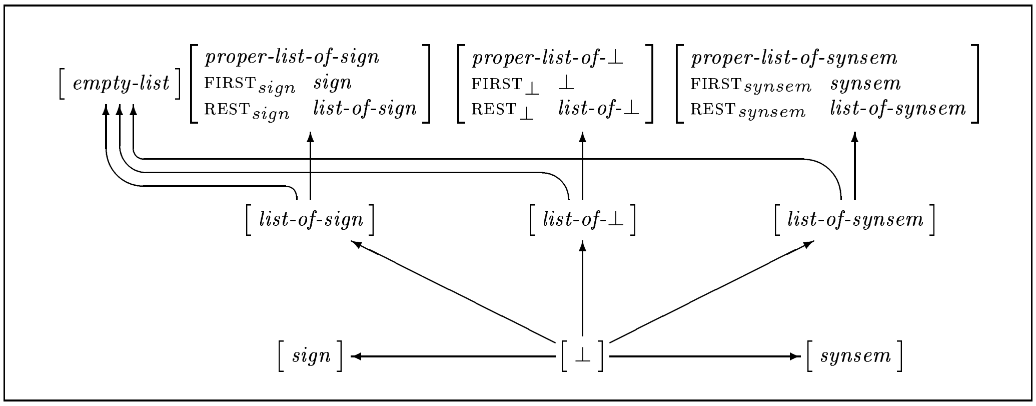


Figure 13

introduction specification. Further, the second amendment avoids the pitfalls of the first amendment concerning well-typed feature structures. However, the second amendment obliges the authors of an HPSG inheritance hierarchy and appropriateness specification to differentiate features they might otherwise not want to differentiate.

To summarize, [Carpenter 1992] imposes the feature introduction condition to enable type inference. However, the condition at worst interferes with well-typedness, and at best unnaturally complicates an inheritance hierarchy and appropriateness specification. Clearly, it would be better if one *could* eliminate the feature introduction condition, and section 3 shows that one *can* eliminate the feature introduction condition.

3 Modification

Suppose figure 14 depicts $\langle \mathsf{Type}, \sqsubseteq \rangle$ and *Approp*. There is no least specific type that deserves feature X. Thus, the feature introduction condition does not hold, and *TypDom* is undefined. Suppose feature structure F_a is

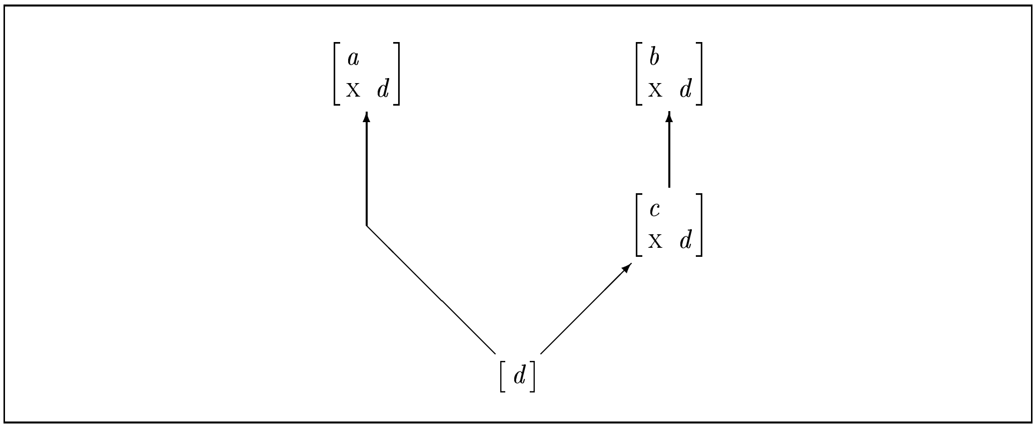
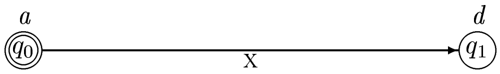
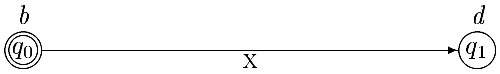


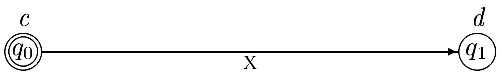
Figure 14



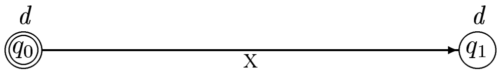
feature structure F_b is



feature structure F_c



and feature structure F_d is



F_d is not a semi-well-typed feature structure, but it subsumes semi-well-typed feature structures F_a, F_b and F_c. However, there is no least explicit semi-well-typed feature structure subsumed by F_d. That is, there is no semi-well-typed feature structure F such that for each semi-well-typed feature structure F',

$F_d \sqsubseteq F'$ iff $F \sqsubseteq F'$.

[CARPENTER 1992] invokes the feature introduction condition expressly to avoid this situation.

However, there is a finite *set* of least explicit semi-well-typed feature structures subsumed by F. That is, there is a set P of semi-well-typed feature structures such that

$\langle P, \sqsubseteq \rangle$ is a finite and pairwise incomparable preorder, and

for each semi-well-typed feature structure F',

$F_d \sqsubseteq F'$ iff for some $F \in P$, $F \sqsubseteq F'$.

Simply,

$P = \{F_a, F_c\}$.

This section modifies *TypDom* and *TypInf* to give functions $TypDom^*$ and $TypInf^*$ such that $TypDom^*(F)$ is the finite set of least explicit semi-well-typed feature structures and $TypInf^*(F)$ is the finite set of least explicit well-typed feature structures subsumed by feature structure F. Both $TypDom^*$ and $TypInf^*$ are well defined regardless of the feature introduction condition, but if the feature introduction condition does hold then $TypInf^*$ reduces to *TypInf* in that for each feature structure F,

$$TypInf^*(F) = \begin{cases} \{TypInf(F)\} & \text{if } TypInf(F) \text{ is defined} \\ \emptyset & \text{if } TypInf(F) \text{ is undefined.} \end{cases}$$

3.1 Preliminary Definitions

35: Definition (Pow^*)

If $\langle P, \preceq\rangle$ is a preorder then

$$Pow^*(\langle P, \preceq\rangle) = \{S \subseteq P \mid \langle S, \preceq\rangle \text{ is finite and pairwise incomparable}\}.$$

For convenience, we write $Pow^*(\boldsymbol{\mathcal{STF}})$ for $Pow^*(\langle \boldsymbol{\mathcal{STF}}, \sqsubseteq\rangle)$, and we write $Pow^*(\boldsymbol{\mathcal{TF}})$ for $Pow^*(\langle \boldsymbol{\mathcal{TF}}, \sqsubseteq\rangle)$.[17]

36: Definition (Min)

If $\langle P, \preceq\rangle$ is a preorder then

$$Min(\langle P, \preceq\rangle) = \{p \in P \mid p \text{ is a minimal member of } \langle P, \preceq\rangle\}.$$

For convenience, if $P \subseteq \mathsf{Type}$ or $P \subseteq \boldsymbol{\mathcal{F}}$ then we write $Min(P)$ for $Min(\langle P, \sqsubseteq\rangle)$.

37: Lemma

If $\langle P, \preceq\rangle$ is a partial order then

$Min(\langle P, \preceq\rangle)$ *is pairwise incomparable.*

Proof

For each p_1 and $p_2 \in Min(\langle P, \preceq\rangle)$,

$$\begin{aligned} & p_1 \preceq p_2 \\ \Longrightarrow\ & p_1 \preceq p_2,\ p_1 \in P \text{ and } p_2 \text{ is a minimal member of } \langle P, \preceq\rangle \\ \Longrightarrow\ & p_1 \preceq p_2 \text{ and } p_2 \preceq p_1 \\ \Longrightarrow\ & p_1 = p_2. && \text{since } \langle P, \preceq\rangle \text{ is a partial order} \end{aligned}$$

Thus,

$Min(\langle P, \preceq\rangle)$ is pairwise incomparable.

□

38: Lemma

If $\langle P, \preceq\rangle$ is a downward Noetherian preorder then for each $p \in P$,

for some $p' \in Min(\langle P, \preceq\rangle)$, $p' \preceq p$.

Proof

For each $p \in P$, suppose

$$S_p = \{s \in P \mid s \preceq p\}.$$

[17] Strictly, we may not be able to apply Pow^* to $\langle \mathcal{STF}, \sqsubseteq\rangle$ or $\langle \mathcal{TF}, \sqsubseteq\rangle$. If $\langle Q, \overline{q}, \theta, \delta\rangle$ is a feature structure then Q is a finite set. The members of Q are taken from an unspecified class C. C may not be a set. Thus, $\mathcal{F}$, $\mathcal{STF}$ and $\mathcal{TF}$ may not be sets. Thus, $\langle \mathcal{F}, \sqsubseteq\rangle$, $\langle \mathcal{STF}, \sqsubseteq\rangle$ and $\langle \mathcal{TF}, \sqsubseteq\rangle$ may not be preorders. However, we have two solutions to this problem. Either redefine a preorder as a pair $\langle P, \preceq\rangle$ such that P is a class, or insist that C be a set.

Thus,

$p \in P$

$\Longrightarrow S_p \subseteq P$ and $S_p \neq \emptyset$

$\Longrightarrow \langle S_p, \preceq \rangle$ has a minimal member. since $\langle P, \preceq \rangle$ is downward Noetherian

For each $p \in P$, suppose

$\widetilde{p}$ is a minimal member of $\langle S_p, \preceq \rangle$.

Firstly, for each $p \in P$,

$\widetilde{p} \preceq p$. ...(1)

Secondly, for each p and $p' \in P$,

$p' \preceq \widetilde{p}$

$\Longrightarrow p' \preceq \widetilde{p}$ and $\widetilde{p} \preceq p$

$\Longrightarrow p' \preceq p$ and $p' \preceq \widetilde{p}$

$\Longrightarrow p' \in S_p$ and $p' \preceq \widetilde{p}$

$\Longrightarrow \widetilde{p} \preceq p'$. since $\widetilde{p}$ is a minimal member of $\langle S_p, \preceq \rangle$

Thus, for each $p \in P$,

$\widetilde{p} \in Min(\langle P, \preceq \rangle)$. ...(2)

By (1) and (2), for each $p \in P$,

for some $p' \in Min(\langle P, \preceq \rangle)$, $p' \preceq p$.

□

39: Definition (*Choice*)

If X and Y are sets and $\theta : X \to Pow(Y)$ is a total function then

$Choice(\theta) = \{\varphi : X \to Y \mid \text{for each } x \in X, \varphi(x) \in \theta(x)\}$.

3.2 A Modification of *TypDom*

40: Definition (ψ_F)

If $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{F}}$ then $\psi_F : Q \to Pow(\mathsf{Type})$ such that for each $t \in \mathsf{Type}$ and each $q \in Q$,

$t \in \psi_F(q)$

iff $\theta(q) \sqsubseteq t$ and for each $f \in \mathsf{Feat}$,

if $\delta(f, q)$ is defined then $Approp(f, t)$ is defined.

41: Definition (Ψ_F)

If $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{F}}$ then $\Psi_F : Q \to Pow(\mathsf{Type})$ such that

$\Psi_F(q) = Min(\psi_F(q))$.

42: Lemma

If $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{F}}$ then for each $q \in Q$,

$\langle \Psi_F(q), \sqsubseteq \rangle$ is pairwise incomparable.

Proof

For each $q \in Q$,

$\langle \psi_F(q), \sqsubseteq \rangle$ is a partial order.

Thus, by lemma 37, for each $q \in Q$,

$\langle \Psi_F(q), \sqsubseteq \rangle$ is pairwise incomparable.

□

43: Lemma

If $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{F}}$ *and* $q \in Q$ *then for each* $t \in \psi_F(q)$,

for some $t' \in \Psi_F(q)$, $t' \sqsubseteq t$.

Proof

By lemma 69, if $q \in Q$ then

$\langle \psi_F(q), \sqsubseteq \rangle$ is a downward Noetherian preorder.

Thus, by lemma 38, if $q \in Q$ then for each $t \in \psi_F(q)$,

for some $t' \in \Psi_F(q)$, $t' \sqsubseteq t$.

□

44: Definition (*TypDom)**

$\mathit{TypDom}^* : \boldsymbol{\mathcal{F}} \to \mathit{Pow}^*(\boldsymbol{\mathcal{STF}})$ *such that for each* $F = \langle Q, \overline{q}, \theta, \delta \rangle \in \boldsymbol{\mathcal{F}}$,

$\mathit{TypDom}^*(F) = \{\langle Q, \overline{q}, \theta', \delta \rangle \mid \theta' \in \mathit{Choice}(\Psi_F)\}$.

45: Proposition

TypDom is well defined.*

Proof

TypDom^* is clearly a total function with domain $\boldsymbol{\mathcal{F}}$, but it remains to be shown that the range of TypDom^* is $\mathit{Pow}^*(\boldsymbol{\mathcal{STF}})$.

Firstly,

$F \in \boldsymbol{\mathcal{F}}$ and $F' \in \mathit{TypDom}^*(F)$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta \rangle$ and for some $\theta' \in \mathit{Choice}(\Psi_F)$, $F' = \langle Q, \overline{q}, \theta', \delta \rangle$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta \rangle$ and for some $\theta' : Q \to \mathsf{Type}$,

$F' = \langle Q, \overline{q}, \theta', \delta \rangle$ and for each $f \in \mathsf{Feat}$ and each $q \in Q$,

$\delta(f, q)$ is defined

$\Longrightarrow$ for each $t \in \psi_F(q)$, $\mathit{Approp}(f, t)$ is defined

$\Longrightarrow$ for each $t \in \Psi_F(q)$, $\mathit{Approp}(f, t)$ is defined

$\Longrightarrow \mathit{Approp}(f, \theta'(q))$ is defined.

$\Longrightarrow F' \in \boldsymbol{\mathcal{STF}}$.

Thus, for each $F \in \boldsymbol{\mathcal{F}}$,

$\mathit{TypDom}^*(F) \subseteq \boldsymbol{\mathcal{STF}}$. ...(1)

Secondly,

$F \in \boldsymbol{\mathcal{F}}$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta \rangle$ and for each $q \in Q$, $\Psi_F(q)$ is finite — since Type is finite

$\Longrightarrow \mathit{Choice}(\Psi_F)$ is finite — since Q is finite

$\Longrightarrow \mathit{TypDom}^*(F)$ is finite.

Thus, for each $F \in \boldsymbol{\mathcal{F}}$,

$\langle \mathit{TypDom}^*(F), \sqsubseteq \rangle$ is finite. ...(2)

Thirdly,

$F \in \boldsymbol{\mathcal{F}}$, $F_1 \in \mathit{TypDom}^*(F)$, $F_2 \in \mathit{TypDom}^*$ and $F_1 \sqsubseteq F_2$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta\rangle$ and for some θ_1 and $\theta_2 \in \mathit{Choice}(\Psi_F)$,
$F_1 = \langle Q, \overline{q}, \theta_1, \delta\rangle$,
$F_2 = \langle Q, \overline{q}, \theta_2, \delta\rangle$, and
for some $h : Q \to Q$,
h is a feature structure morphism from F_1 to F_2

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta\rangle$ and for some θ_1 and $\theta_2 \in \mathit{Choice}(\Psi_F)$,
$F_1 = \langle Q, \overline{q}, \theta_1, \delta\rangle$,
$F_2 = \langle Q, \overline{q}, \theta_2, \delta\rangle$, and
for each $q \in Q$, $\theta_1(q) \sqsubseteq \theta_2(q)$ by lemma 10

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta\rangle$ and for some θ_1 and $\theta_2 : Q \to \mathsf{Type}$,
$F_1 = \langle Q, \overline{q}, \theta_1, \delta\rangle$,
$F_2 = \langle Q, \overline{q}, \theta_2, \delta\rangle$, and
for each $q \in Q$,
$\theta_1(q) \sqsubseteq \theta_2(q)$, $\theta_1(q) \in \Psi_F(q)$ and $\theta_2(q) \in \Psi_F(q)$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta\rangle$ and for some θ_1 and $\theta_2 : Q \to \mathsf{Type}$,
$F_1 = \langle Q, \overline{q}, \theta_1, \delta\rangle$,
$F_2 = \langle Q, \overline{q}, \theta_2, \delta\rangle$, and
for each $q \in Q$, $\theta_1(q) = \theta_2(q)$ by lemma 42

$\Longrightarrow F_1 = F_2$.

Thus, for each $F \in \boldsymbol{\mathcal{F}}$,
$\langle \mathit{TypDom}^*(F), \sqsubseteq\rangle$ is pairwise incomparable. ...(3)

By (1), (2) and (3), for each $F \in \boldsymbol{\mathcal{F}}$,
$\mathit{TypDom}^*(F) \in \mathit{Pow}^*(\boldsymbol{\mathcal{STF}})$.

□

46: Proposition

If $F \in \boldsymbol{\mathcal{F}}$ *and* $F' \in \mathit{TypDom}^*(F)$ *then*
$F \sqsubseteq F'$.

Proof

$F \in \boldsymbol{\mathcal{F}}$ and $F' \in \mathit{TypDom}^*(F)$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta\rangle$ and for some $\theta' \in \mathit{Choice}(\Psi_F)$, $F' = \langle Q, \overline{q}, \theta', \delta\rangle$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta\rangle$ and for some $\theta' : Q \to \mathsf{Type}$,
$F' = \langle Q, \overline{q}, \theta', \delta\rangle$ and for each $q \in Q$, $\theta'(q) \in \Psi_F(q)$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta\rangle$ and for some $\theta' : Q \to \mathsf{Type}$,
$F' = \langle Q, \overline{q}, \theta', \delta\rangle$ and for each $q \in Q$, $\theta(q) \sqsubseteq \theta'(q)$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta\rangle$, and
the identity function on Q is a feature structure morphism from F to F'

$\Longrightarrow F \sqsubseteq F'$.

□

47: Proposition

If $F \in \boldsymbol{\mathcal{F}}$ *and* $F'' \in \boldsymbol{\mathcal{STF}}$ *then*
$F \sqsubseteq F''$ *iff for some* $F' \in \mathit{TypDom}^*(F)$, $F' \sqsubseteq F''$.

Proof

Part 1. From left to right.

Suppose

$F = \langle Q, \overline{q}, \theta, \delta \rangle$,

$F'' = \langle Q'', \overline{q}'', \theta'', \delta'' \rangle$, and

h is a feature structure morphism from F to F'.

Thus, for each $q \in Q$

$\theta(q) \sqsubseteq \theta''(h(q))$,

and for each $f \in \mathsf{Feat}$ and each $q \in Q$,

$\delta(f, q)$ is defined

$\Longrightarrow \delta''(f, h(q))$ is defined

$\Longrightarrow Approp(f, \theta''(h(q)))$ is defined since $F'' \in \boldsymbol{\mathcal{STF}}$

Thus, for each $q \in Q$,

$\theta''(h(q)) \in \psi_F(q)$.

Thus, for each $q \in Q$,

for some $t \in \Psi_F(q)$, $t \sqsubseteq \theta''(h(q))$. by lemma 43

Thus, for some $\theta' \in Choice(\Psi_F)$,

for each $q \in Q$, $\theta'(q) \sqsubseteq \theta''(h(q))$.

Thus, for some $\theta' \in Choice(\Psi_F)$,

h is a feature structure morphism from $\langle Q, \overline{q}, \theta', \delta \rangle$ to F''.

Thus, for some $F' \in TypDom^*(F)$,

h is a feature structure morphism from F' to F''.

Thus, for some $F' \in TypDom^*(F)$,

$F' \sqsubseteq F''$.

Part 2. From right to left.

for some $F' \in TypDom^*(F)$, $F' \sqsubseteq F''$

$\Longrightarrow$ for some $F' \in TypDom^*(F)$, $F \sqsubseteq F'$ and $F' \sqsubseteq F''$ by proposition 46

$\Longrightarrow F \sqsubseteq F''$.

□

3.3 A Modification of *TypInf*

48: Definition (Θ)

$\Theta : \boldsymbol{\mathcal{F}} \to Pow(\boldsymbol{\mathcal{F}})$ *such that*

$\Theta(F) = \{TypRan(F') \mid F' \in TypDom^*(F)$ *and* $TypRan(F')$ *is defined*$\}$.

49: Definition ($TypInf^*$)

Let $TypInf^* : \boldsymbol{\mathcal{F}} \to Pow^*(\boldsymbol{\mathcal{TF}})$ *by*

$TypInf^*(F) = Min(\Theta(F))$.

50: Proposition

$TypInf^*$ is well defined.

Proof

TypInf* is clearly a total function with domain $\boldsymbol{\mathcal{F}}$, but it remains to be shown that the range of *TypInf** is *Pow*$^*(\boldsymbol{\mathcal{TF}})$.

Firstly,

$F \in \boldsymbol{\mathcal{F}}$ and $F'' \in \mathit{TypInf}^*(F)$

$\Longrightarrow F \in \boldsymbol{\mathcal{F}}$ and $F'' \in \Theta(F)$

$\Longrightarrow F \in \boldsymbol{\mathcal{F}}$ and for some $F' \in \mathit{TypDom}^*(F)$,

$\quad\quad \mathit{TypRan}(F')$ is defined and $F'' = \mathit{TypRan}(F')$

$\Longrightarrow$ for some $F' \in \boldsymbol{\mathcal{STF}}$, $\mathit{TypRan}(F')$ is defined and $F'' = \mathit{TypRan}(F')$

$\Longrightarrow F'' \in \boldsymbol{\mathcal{TF}}$.

Thus, for each $F \in \boldsymbol{\mathcal{F}}$,

$\mathit{TypInf}^*(F) \subseteq \boldsymbol{\mathcal{TF}}$. ...(1)

Secondly,

$F \in \boldsymbol{\mathcal{F}}$

$\Longrightarrow \mathit{TypDom}^*(F)$ is finite

$\Longrightarrow \Theta(F)$ is finite

$\Longrightarrow \mathit{TypInf}^*(F)$ is finite.

Thus, for each $F \in \boldsymbol{\mathcal{F}}$,

$\langle \mathit{TypInf}^*(F), \sqsubseteq\rangle$ is finite. ...(2)

Thirdly,

$F \in \boldsymbol{\mathcal{F}}$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta\rangle$ and $\mathit{TypDom}^*(F) \subseteq \{\langle Q, \overline{q}, \theta', \delta\rangle \mid \theta' : Q \to \mathsf{Type}\}$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta\rangle$ and $\Theta(F) \subseteq \{\langle Q, \overline{q}, \theta', \delta\rangle \mid \theta' : Q \to \mathsf{Type}\}$

$\Longrightarrow \langle \Theta(F), \sqsubseteq\rangle$ is a partial order by lemma 11

$\Longrightarrow \langle \mathit{TypInf}^*(F), \sqsubseteq\rangle$ is pairwise incomparable. by lemma 37

Thus, for each $F \in \boldsymbol{\mathcal{F}}$,

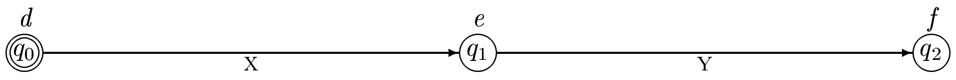
$\langle \mathit{TypInf}^*(F), \sqsubseteq\rangle$ is pairwise incomparable. ...(3)

By (1), (2) and (3), for each $F \in \boldsymbol{\mathcal{F}}$,

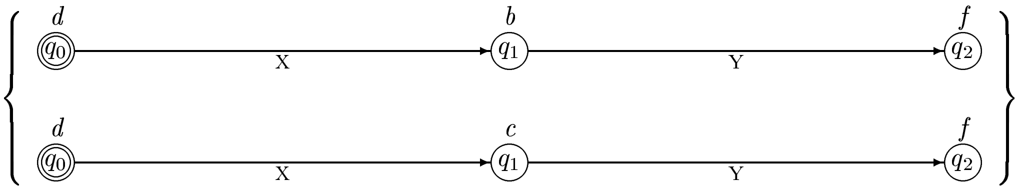
$\mathit{TypInf}^*(F) \in \mathit{Pow}^*(\boldsymbol{\mathcal{TF}})$.

□

If F is a feature structure then $\langle \Theta(F), \sqsubseteq\rangle$ need not be pairwise incomparable. For example, suppose figure 15 depicts $\langle \mathsf{Type}, \sqsubseteq\rangle$ and *Approp*, and feature structure F is



Then $\mathit{TypDom}^*(F)$ is



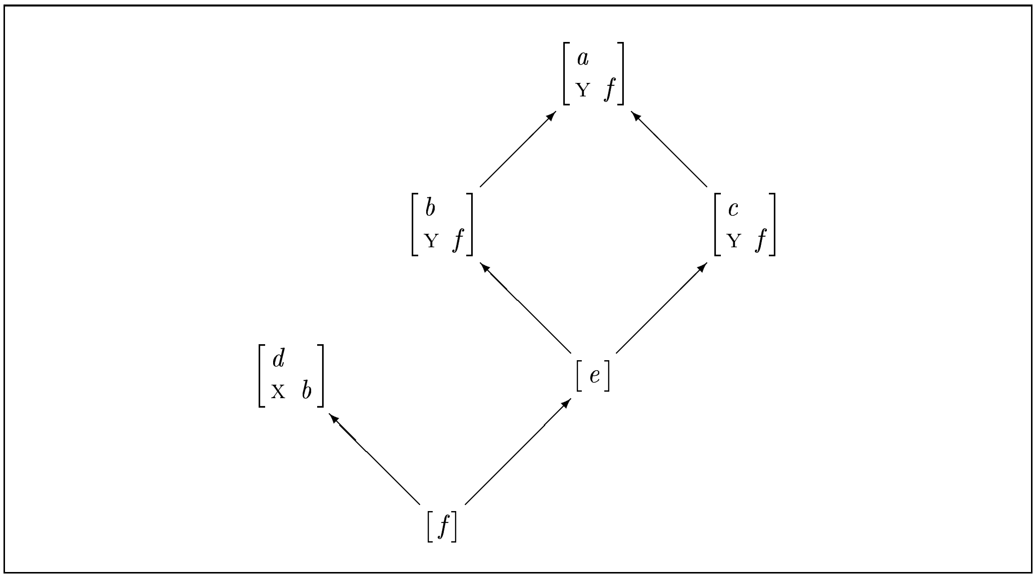
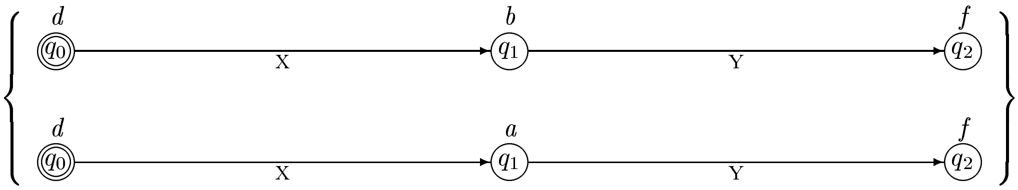


Figure 15

and $\Theta(F)$ is



$\langle \Theta(F), \sqsubseteq \rangle$ is not pairwise incomparable. By defining $\mathit{TypInf}^*(F)$ as $\mathit{Min}(\langle \Theta(F), \sqsubseteq \rangle)$, we ensure that if F is a feature structure then $\langle \mathit{TypInf}^*(F), \sqsubseteq \rangle$ is pairwise incomparable.

51: Lemma

If $F \in \boldsymbol{\mathcal{F}}$ then for each $F'' \in \Theta(F)$,

for some $F' \in \mathit{TypInf}^*(F)$, $F' \sqsubseteq F''$.

Proof

$F \in \boldsymbol{\mathcal{F}}$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta \rangle$ and $\mathit{TypDom}^*(F) \subseteq \{\langle Q, \overline{q}, \theta', \delta \rangle \mid \theta' : Q \to \mathsf{Type}\}$

$\Longrightarrow F = \langle Q, \overline{q}, \theta, \delta \rangle$ and $\Theta(F) \subseteq \{\langle Q, \overline{q}, \theta', \delta \rangle \mid \theta' : Q \to \mathsf{Type}\}$

$\Longrightarrow \langle \Theta(F), \sqsubseteq \rangle$ is a downward Noetherian preorder — by lemmata 11 and 69

$\Longrightarrow$ for each $F'' \in \Theta(F)$, for some $F' \in \mathit{TypInf}^*(F)$, $F' \sqsubseteq F''$. — by lemma 38

$\square$

52: Proposition

If $F \in \boldsymbol{\mathcal{F}}$ and $F'' \in \mathit{TypInf}^*(F)$ then

$F \sqsubseteq F''$.

Proof

$F \in \boldsymbol{\mathcal{F}}$ and $F'' \in \mathit{TypInf}^*(F)$

$\Longrightarrow F \in \boldsymbol{\mathcal{F}}$ and $F'' \in \Theta(F)$

$\Longrightarrow F \in \boldsymbol{\mathcal{F}}$ and for some $F' \in \mathit{TypDom}^*(F)$,

$\mathit{TypRan}(F')$ is defined and $F'' = \mathit{TypRan}(F')$

$\Longrightarrow$ for some $F' \in \boldsymbol{\mathcal{STF}}$, $F \sqsubseteq F'$ and $F' \sqsubseteq F''$ by propositions 46 and 30

$\Longrightarrow F \sqsubseteq F''$.

□

53: Proposition

If $F \in \boldsymbol{\mathcal{F}}$ and $F'' \in \boldsymbol{\mathcal{TF}}$ *then*

$F \sqsubseteq F''$ *iff for some* $F' \in \mathit{TypInf}^*(F)$, $F' \sqsubseteq F''$.

Proof

$F \sqsubseteq F''$

$\Longleftrightarrow$ for some $F^* \in \mathit{TypDom}^*(F)$, $F^* \sqsubseteq F''$ by propositions 14 and 47

$\Longleftrightarrow$ for some $F^* \in \mathit{TypDom}^*(F)$,

$\mathit{TypRan}(F^*)$ is defined and $\mathit{TypRan}(F^*) \sqsubseteq F''$ by proposition 31

$\Longleftrightarrow$ for some $F^{**} \in \Theta(F)$, $F^{**} \sqsubseteq F''$

$\Longleftrightarrow$ for some $F' \in \mathit{TypInf}^*(F)$, $F' \sqsubseteq F''$. by lemma 51

□

3.4 Relating *TypInf* and *TypInf**

This subsection shows that if the feature introduction condition holds then TypDom^* and TypInf^* reduce to *TypDom* and *TypInf*.

Warning! Throughout 3.4 we assume the existence of a feature introduction function $\mathit{Intro} : \mathsf{Feat} \to \mathsf{Type}$.

54: Lemma

If $F = \langle Q, \overline{q}, \theta, \delta\rangle \in \boldsymbol{\mathcal{F}}$ *then for each* $q \in Q$ *and each* $t \in \mathsf{Type}$,

$t \in \psi_F(q)$ *iff* $\Phi_F(q)$ *is defined and* $\Phi_F(q) \sqsubseteq t$.

Proof

For each $q \in Q$ and each $t \in \mathsf{Type}$,

$t \in \psi_F(q)$

$\Longleftrightarrow \theta(q) \sqsubseteq t$ and for each $f \in \mathsf{Feat}$,

if $\delta(f, q)$ is defined then $\mathit{Approp}(f, t)$ is defined

$\Longleftrightarrow \theta(q) \sqsubseteq t$ and for each $f \in \mathsf{Feat}$, if $\delta(f, q)$ is defined then $\mathit{Intro}(f) \sqsubseteq t$

$\Longleftrightarrow \Phi_F(q)$ is defined and $\Phi_F(q) \sqsubseteq t$.

□

55: Lemma

If $F = \langle Q, \overline{q}, \theta, \delta\rangle \in \boldsymbol{\mathcal{F}}$ *then for each* $q \in Q$,

$$\Psi_F(q) = \begin{cases} \{\Phi_F(q)\} & \text{if } \Phi_F(q) \text{ is defined} \\ \emptyset & \text{if } \Phi_F(q) \text{ is undefined.} \end{cases}$$

Proof

Firstly, for each $q \in Q$,

$\Phi_F(q)$ is defined

$\Longrightarrow \Phi_F(q) \in \psi_F(q)$ and for each $t \in \psi_F(q)$, $\Phi_F(q) \sqsubseteq t$ by lemma 54

$\Longrightarrow \Phi_F(q) \in \Psi_F(q)$ and for each $t \in \Psi_F(q)$, $\Phi_F(q) \sqsubseteq t$

$\Longrightarrow \Phi_F(q) \in \Psi_F(q)$ and for each $t \in \Psi_F(q)$, $\Phi_F(q) = t$ by lemma 42

$\Longrightarrow \Psi_F(q) = \{\Phi_F(q)\}$. ...(1)

Secondly, for each $q \in Q$,

$\Phi_F(q)$ is undefined

$\Longrightarrow \psi_F(q) = \emptyset$ by lemma 54

$\Longrightarrow \Psi_F(q) = \emptyset$. ...(2)

By (1) and (2), for each $q \in Q$,

$$\Psi_F(q) = \begin{cases} \{\Phi_F(q)\} & \text{if } \Phi_F(q) \text{ is defined} \\ \emptyset & \text{if } \Phi_F(q) \text{ is undefined.} \end{cases}$$

□

56: Proposition

If $F \in \mathcal{F}$ then

$$\mathit{TypDom}^*(F) = \begin{cases} \{TypDom(F)\} & \textit{if } \mathit{TypDom}(F) \textit{ is defined} \\ \emptyset & \textit{if } \mathit{TypDom}(F) \textit{ is undefined.} \end{cases}$$

Proof

Suppose

$F = \langle Q, \overline{q}, \theta, \delta \rangle$.

Firstly,

$\mathit{TypDom}(F)$ is defined

$\Longrightarrow$ for each $q \in Q$, $\Phi_F(q)$ is defined

$\Longrightarrow$ for each $q \in Q$, $\Psi_F(q) = \{\Phi_F(q)\}$ by lemma 55

$\Longrightarrow \mathit{Choice}(\Psi_F) = \{\Phi_F\}$

$\Longrightarrow \mathit{TypDom}^*(F) = \{\mathit{TypDom}(F)\}$. ...(1)

Secondly,

$\mathit{TypDom}(F)$ is undefined

$\Longrightarrow$ for some $q \in Q$, $\Phi_F(q)$ is undefined

$\Longrightarrow$ for each $q \in Q$, $\Psi_F(q) = \emptyset$ by lemma 55

$\Longrightarrow \mathit{Choice}(\Psi_F) = \emptyset$

$\Longrightarrow \mathit{TypDom}^*(F) = \emptyset$. ...(2)

By (1) and (2),

$$\mathit{TypDom}^*(F) = \begin{cases} \{TypDom(F)\} & \text{if } \mathit{TypDom}(F) \text{ is defined} \\ \emptyset & \text{if } \mathit{TypDom}(F) \text{ is undefined.} \end{cases}$$

□

57: Proposition

If $F \in \mathcal{F}$ then

$$\mathit{TypInf}^*(F) = \begin{cases} \{TypInf(F)\} & \textit{if } \mathit{TypInf}(F) \textit{ is defined} \\ \emptyset & \textit{if } \mathit{TypInf}(F) \textit{ is undefined.} \end{cases}$$

Proof

Firstly,

$\mathit{TypInf}(F)$ is defined

$\Longrightarrow \mathit{TypDom}(F)$ is defined and $\mathit{TypRan}(\mathit{TypDom}(F))$ is defined

$\Longrightarrow$ *TypDom*$^*(F) = \{$*TypDom*$(F)\}$ and *TypRan*(*TypDom*(F)) is defined
by proposition 56

$\Longrightarrow \Theta(F) = \{$*TypInf*$(F)\}$

$\Longrightarrow$ *TypInf*$^*(F) = \{$*TypInf*$(F)\}$. ...(1)

Secondly,

TypInf(F) is undefined

$\Longrightarrow$ *TypDom*(F) is undefined, or
TypDom(F) is defined and *TypRan*(*TypDom*(F)) is undefined

$\Longrightarrow$ *TypDom*$^*(F) = \emptyset$, or
TypDom$^*(F) = \{$*TypDom*$(F)\}$ and *TypRan*(*TypDom*(F)) is undefined,
by proposition 56

$\Longrightarrow$ *TypInf*$^*(F) = \emptyset$. ...(2)

By (1) and (2),

$$\mathit{TypInf}^*(F) = \begin{cases} \{TypInf(F)\} & \text{if } \mathit{TypInf}(F) \text{ is defined} \\ \emptyset & \text{if } \mathit{TypInf}(F) \text{ is undefined.} \end{cases}$$

□

A Appendix

This appendix contains the standard mathematical definitions and results concerning preorders that the reader needs to understand our paper.

58: Definition (preorder)

A preorder is a pair $\langle P, \preceq \rangle$ such that

P is a set,

$\preceq$ is a binary relation,

for each $p \in P$, $p \preceq p$, and

for each $p_1, p_2, p_3 \in P$, if $p_1 \preceq p_2$ and $p_2 \preceq p_3$ then $p_1 \preceq p_3$.

59: Definition (finite preorder)

A finite preorder is a pair $\langle P, \preceq \rangle$ such that

$\langle P, \preceq \rangle$ is a preorder, and

P is finite.

60: Definition (pairwise incomparable preorder)

A pairwise incomparable preorder is a pair $\langle P, \preceq \rangle$ such that

$\langle P, \preceq \rangle$ is a preorder, and

for each $p_1, p_2 \in P$, if $p_1 \preceq p_2$ then $p_1 = p_2$.

61: Definition (partial order)

A partial order is a pair $\langle P, \preceq \rangle$ such that

$\langle P, \preceq \rangle$ is a preorder, and

for each $p_1, p_2 \in P$, if $p_1 \preceq p_2$ and $p_2 \preceq p_1$ then $p_1 = p_2$.

62: Definition (upper bound)

If $\langle P, \preceq \rangle$ is a preorder and $S \subseteq P$ then an upper bound of S in $\langle P, \preceq \rangle$ is an entity p such that

$p \in P$, and

for each $s \in S$, $s \preceq p$.

63: Definition (least upper bound)

If $\langle P, \preceq \rangle$ is a preorder and $S \subseteq P$ then a least upper bound of S in $\langle P, \preceq \rangle$ is an entity p such that

p is an upper bound of S in $\langle P, \preceq \rangle$, and

if p' is an upper bound of S in $\langle P, \preceq \rangle$ then $p \preceq p'$.

64: Definition (bounded complete preorder)

A bounded complete preorder is a pair $\langle P, \preceq \rangle$ such that

$\langle P, \preceq \rangle$ is a preorder, and

for each $S \subseteq P$,

if S has an upper bound in $\langle P, \preceq \rangle$

then S has a least upper bound in $\langle P, \preceq \rangle$.

65: Definition (minimal member)

If $\langle P, \preceq \rangle$ is a preorder then a minimal member of $\langle P, \preceq \rangle$ is an entity p such that

$p \in P$, and

for each $p' \in P$, if $p' \preceq p$ then $p \preceq p'$.

66: Definition (maximal member)

If $\langle P, \preceq \rangle$ is a preorder then a maximal member of $\langle P, \preceq \rangle$ is an entity p such that

$p \in P$, and

for each $p' \in P$, if $p \preceq p'$ then $p' \preceq p$.

67: Definition (upward Noetherian preorder)

An upward Noetherian preorder is a pair $\langle P, \preceq \rangle$ such that

$\langle P, \preceq \rangle$ is a preorder, and

for each $S \subseteq P$, if $S \neq \emptyset$ then $\langle S, \preceq \rangle$ has a maximal member.

68: Definition (downward Noetherian preorder)

A downward Noetherian preorder is a pair $\langle P, \preceq \rangle$ such that

$\langle P, \preceq \rangle$ is a preorder, and

for each $S \subseteq P$, if $S \neq \emptyset$ then $\langle S, \preceq \rangle$ has a minimal member.

69: Lemma

If $\langle P, \preceq \rangle$ is a finite preorder then

$\langle P, \preceq \rangle$ is an upward Noetherian preorder, and

$\langle P, \preceq \rangle$ is a downward Noetherian preorder.

Proof

For each preorder $\langle P, \preceq \rangle$,

$\langle P, \preceq \rangle$ is not upward Noetherian

$\Longrightarrow$ for some $S \subseteq P$, $S \neq \emptyset$ and $\langle S, \preceq \rangle$ has no maximal member

$\Longrightarrow$ for some $S \subseteq P$,

$S \neq \emptyset$ and for each $s \in S$, for some $s' \in S$, $s \preceq s'$ and $s' \not\preceq s$.

$\Longrightarrow$ for some $\{s_i \mid i \in \mathbb{N}\} \subseteq P$, for each $i \in \mathbb{N}$, $s_i \preceq s_{i+1}$ and $s_{i+1} \not\preceq s_i$.

$\Longrightarrow$ for some $\{s_i \mid i \in \mathbb{N}\} \subseteq P$, for all i and $j \in \mathbb{N}$,

$i < j$ and $s_i = s_j$

$\Longrightarrow i + 1 \leq j$ and $s_i = s_j$

$\Longrightarrow s_{i+1} \preceq s_j$ and $s_i = s_j$

$\Longrightarrow s_{i+1} \preceq s_i$

$\Longrightarrow$ contradiction

$\Longrightarrow$ for some $\{s_i \mid i \in \mathbb{N}\} \subseteq P$, for all i and $j \in \mathbb{N}$, if $i \neq j$ then $s_i \neq s_j$

$\Longrightarrow$ for some $\{s_i \mid i \in \mathbb{N}\} \subseteq P$, $\{s_i \mid i \in \mathbb{N}\}$ is infinite

$\Longrightarrow P$ is infinite

$\Longrightarrow \langle P, \preceq\rangle$ is not finite.

Thus,

if $\langle P, \preceq\rangle$ is a finite preorder then $\langle P, \preceq\rangle$ is an upward Noetherian preorder.

Similarly, for each preorder $\langle P, \preceq\rangle$,

if $\langle P, \preceq\rangle$ is not downward Noetherian then $\langle P, \preceq\rangle$ is not finite.

Thus,

if $\langle P, \preceq\rangle$ is a finite preorder then $\langle P, \preceq\rangle$ is a downward Noetherian preorder.

□

70: Lemma

If $\langle\{p_n \mid n \in \mathbb{N}\}, \preceq\rangle$ *is an upward Noetherian partial order and for each* $n \in \mathbb{N}$, $p_n \preceq p_{n+1}$ *then*

$\langle\{p_n \mid n \in \mathbb{N}\}, \preceq\rangle$ *has a unique maximal member.*

Proof

Firstly, since $\langle\{p_n \mid n \in \mathbb{N}\}, \preceq\rangle$ is an upward Noetherian preorder,

$\langle\{p_n \mid n \in \mathbb{N}\}, \preceq\rangle$ has a maximal member.

Secondly, for each i and $j \in \mathbb{N}$,

p_i and p_j are maximal members of $\langle\{p_n \mid n \in \mathbb{N}\}, \preceq\rangle$

$\overset{\text{wlog}}{\Longrightarrow} p_i$ and p_j are maximal members of $\langle\{p_n \mid n \in \mathbb{N}\}, \preceq\rangle$ and $i \leq j$

$\Longrightarrow p_i$ and p_j are maximal members of $\langle\{p_n \mid n \in \mathbb{N}\}, \preceq\rangle$ and $p_i \preceq p_j$

$\Longrightarrow p_i \preceq p_j$ and $p_j \preceq p_i$

$\Longrightarrow p_i = p_j$. since $\langle\{p_n \mid n \in \mathbb{N}\}, \preceq\rangle$ is a partial order

Thus,

$\langle\{p_n \mid n \in \mathbb{N}\}, \preceq\rangle$ has a unique maximal member.

□

B References

[Carpenter 1992]. Bob Carpenter. *The Logic of Typed Feature Structures.* Volume 32 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press. Cambridge, Massachusetts, USA. 1992.

[Kay 1979]. Martin Kay. "Functional Grammar". In (pages 142-158): C. Chiarello et. al. (editors). *The Proceedings of the 5th Annual Meeting of the Berkeley*

Linguistics Society. Berkeley Linguistics Society. Berkeley, California, USA. 1979.

[KAY 1984]. Martin Kay. "Functional Unification Grammar: a Formalism for Machine Translation". In (pages 75-78): *The Proceedings of the 10th International Conference on Computational Linguistics*. North-Holland. Amsterdam, The Netherlands. 1984.

[KING 1989]. Paul J. King. *A Logical Formalism for Head-driven Phrase Structure Grammar*. Ph. D. Thesis. Manchester University. Manchester, England. 1989.

[POLLARD AND MOSHIER 1990]. Carl J. Pollard and M. Drew Moshier. "Unifying Partial Descriptions of Sets". In: P. Hanson (editor). *Information, Language and Cognition*. Volume 1 of *Vancouver Studies in Cognitive Science*. University of British Columbia Press. Vancouver, British Columbia, Canada. 1990.

[POLLARD AND SAG 1987]. Carl J. Pollard and Ivan A. Sag. *Information-based Syntax and Semantics (Volume 1: Fundamentals)*. Number 13 of *CSLI Lecture Notes*. CSLI Publications. Stanford, California, USA. 1987.

[POLLARD AND SAG 1993]. Carl J. Pollard and Ivan A. Sag. *Head-driven Phrase Structure Grammar*. University of Chicago Press and CSLI Publications. Chicago, Illinois, USA. 1993.

[SMOLKA 1988]. Gert Smolka. *A Feature Logic with Subsorts*. LILOG-Report 33. IBM Deutschland GmbH. Stuttgart, Germany. 1988.