captures remains intact. The set of morphs admitted by a grammar can therefore serve as a set of mathematical representations of the object types of the language that the grammar describes. The meaning of a grammar is the set of morphs admitted by the grammar.

Finally, I will show that Pollard feature structures and Pollard abstract feature structures have very close analogues in RSRL. The definition of extended Pollard abstract feature structures leads directly to a notion of the strong generative capacity of an RSRL grammar. According to one of the three definitions that Pollard offers, the strong generative capacity is the set of all extended Pollard abstract feature structures that can be abstracted from the models of the grammar. It is then possible to see the meaning of a grammar in its strong generative capacity.

3.1.1 The Description Language

Following SRL, descriptions of RSRL are true or false of entities in an interpretation. RSRL provides a class of formal languages consisting of a signature and a class of interpretations. The signature provides the sets of symbols which generate the formal language. Each interpretation of a signature provides a set of entities and assigns each symbol of the signature a meaning in terms of those entities. Formulae are built from variables, the sets of symbols of the signature, and a small set of reserved symbols. Each formula in which no variable occurs free is called a description. A theory is a set of descriptions. Each variable that occurs in a description in a theory will thus be bound by component quantification. Interpretations induce a formula denotation function, a description denotation function and a theory denotation function. The description denotation function assigns each description the set of entities of which it is true, and the theory denotation function assigns each theory the set of entities of which each description in the theory is true. A grammar is a signature together with a theory; and a model of a grammar is an interpretation of its signature that only contains entities of which every description in the theory of the grammar is true. Following King 1999, the meaning of a grammar can then be determined by a class of models which meet certain conditions. Informally, a grammar delimits a class of interpretations, EM, of its signature such that its theory is true of every entity in each interpretation in EM, and for every entity u in any other model of the grammar, each interpretation in EM contains an entity that is indiscernible from u.

The syntactic symbols of each formal language are given by the set of variables and the signature:

Definition 46 $\mathcal{VAR}$ *is a countably infinite set of symbols.*

I call each element of $\mathcal{VAR}$ a *variable*.

In contrast to SRL signatures, the signatures of RSRL contain an explicit sort hierarchy. The motivation for the inclusion of sort hierarchies comes from the intended main application of the formal languages, the specification of HPSG 94 grammars, which make heavy use of sort hierarchies and attribute inheritance in sort hierarchies. Including sort hierarchies explicitly in the formal languages merely avoids the extra step of the definition of a syntactic translation from grammars with a sort hierarchy to corresponding grammars without. From a purely mathematical point of view, nothing is gained by an explicit sort hierarchy. For reasons of mathematical elegance, it could as well be left implicit, as argued by King 1999, pp. 229–331. Similar to SRL, which allows infinite sets of species in signatures, I allow infinite sort hierarchies. In actual linguistic grammars, the sort hierarchies are finite.

Definition 47 Σ *is a* ***signature*** *iff*

Σ *is a septuple* $\langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$,

$\langle \mathcal{G}, \sqsubseteq \rangle$ *is a partial order,*

$$\mathcal{S} = \left\{ \sigma \in \mathcal{G} \;\middle|\; \begin{array}{l} \text{for each } \sigma' \in \mathcal{G}, \\ \text{if } \sigma' \sqsubseteq \sigma \text{ then } \sigma = \sigma' \end{array} \right\},$$

$\mathcal{A}$ *is a set,*

$\mathcal{F}$ *is a partial function from the Cartesian product of* $\mathcal{G}$ *and* $\mathcal{A}$ *to* $\mathcal{G}$,

for each $\sigma_1 \in \mathcal{G}$, *for each* $\sigma_2 \in \mathcal{G}$ *and for each* $\alpha \in \mathcal{A}$,

> *if* $\mathcal{F}\langle \sigma_1, \alpha \rangle$ *is defined and* $\sigma_2 \sqsubseteq \sigma_1$
> *then* $\mathcal{F}\langle \sigma_2, \alpha \rangle$ *is defined and* $\mathcal{F}\langle \sigma_2, \alpha \rangle \sqsubseteq \mathcal{F}\langle \sigma_1, \alpha \rangle$,

$\mathcal{R}$ *is a finite set, and*

$\mathcal{AR}$ *is a total function from* $\mathcal{R}$ *to* $\mathbb{N}^+$.

I call each element of $\mathcal{G}$ a *sort*, $\langle \mathcal{G}, \sqsubseteq \rangle$ the *sort hierarchy*, each element of $\mathcal{S}$ a *maximally specific sort* or a *species*, each element of $\mathcal{A}$ an *attribute*, $\mathcal{F}$ the *appropriateness function*, each element of $\mathcal{R}$ a *relation symbol*, and $\mathcal{AR}$ the *arity function*. I say that sort σ_2 is *at least as specific as* sort σ_1 iff $\sigma_2 \sqsubseteq \sigma_1$; conversely, I call σ_1 a *supersort* of σ_2. The symbols $\forall$, $\exists$, :, $\sim$, $\approx$, $\neg$, [,], $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\dagger$, $\triangleright$, *chain, echain, nechain* and *metatop* are reserved symbols, and I assume that none of them are a variable, a sort, an attribute or a relation symbol. For reasons of generality, the set of attributes in a signature is allowed to be infinite. In signatures of linguistic grammars, the set of attributes is finite. The appropriateness function, $\mathcal{F}$, respects the usual postulates about attribute inheritance in sort hierarchies. If an attribute, α, is appropriate to a sort σ_1 then it is appropriate to any sort σ_2 which is more specific than σ_1, and $\mathcal{F}\langle \sigma_2, \alpha \rangle$ is at least as specific as $\mathcal{F}\langle \sigma_1, \alpha \rangle$. The arity function, $\mathcal{AR}$, specifies the arity of each relation symbol as a positive integer. For example, the standard `member` relation has arity 2, and the standard `append` relation has arity 3.

Before I define the class of interpretations of each signature, I need to introduce a few notational conventions that will make the following definitions easier to read. Suppose S is a set. S^* is the set of finite strings over S, including the empty string. I will refer to it as the set of finite sequences of elements of S. Similarly, S^+ is the set of nonempty finite sequences of elements of S. For convenience, I will henceforth write $\overline{S}$ as an abbreviation for $S \cup S^*$.

RSRL signatures are interpreted as follows:

Definition 48 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, I *is a* Σ ***interpretation*** *iff*

> I *is a quadruple* $\langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,
>
> U *is a set,*
>
> S *is a total function from* U *to* $\mathcal{S}$,
>
> A *is a total function from* $\mathcal{A}$ *to the set of partial functions from* U *to* U,
>
> *for each* $\alpha \in \mathcal{A}$ *and each* $u \in \mathsf{U}$,
>
> > *if* $\mathsf{A}(\alpha)(u)$ *is defined*
> > *then* $\mathcal{F}\langle \mathsf{S}(u), \alpha \rangle$ *is defined, and* $\mathsf{S}(\mathsf{A}(\alpha)(u)) \sqsubseteq \mathcal{F}\langle \mathsf{S}(u), \alpha \rangle$, *and*

for each $\alpha \in \mathcal{A}$ *and each* $u \in \mathsf{U}$,

if $\mathcal{F}\langle \mathsf{S}(u), \alpha\rangle$ *is defined then* $\mathsf{A}(\alpha)(u)$ *is defined,*

R *is a total function from* $\mathcal{R}$ *to the power set of* $\bigcup_{n \in \mathbb{N}} \overline{\mathsf{U}}^{n}$, *and*

for each $\rho \in \mathcal{R}$, $\mathsf{R}(\rho) \subseteq \overline{\mathsf{U}}^{\mathcal{AR}(\rho)}$.

U is the *set of entities in the universe.* I call S the *species assignment function*, A the *attribute interpretation function*, and R the *relation interpretation function.* As in the previous chapter, I adopt the convention of saying that an entity, u, is an element of a Σ interpretation $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$ if $u \in \mathsf{U}$, and I sometimes write $u \in \mathsf{I}$ to indicate that u is in U.

S partitions the entities of the universe by assigning a species to every entity. Informally, one might say that each species denotes a set of entities, and each entity is in the denotation of exactly one species. Extending this terminology to sorts in general, I say that each sort, σ, denotes the union of the sets of entities denoted by the species of which σ is a supersort. The function A provides a denotation for attributes. Each attribute denotes a partial function from entities to entities. For convenience, I sometimes refer to the function denoted by an attribute, α, as (the function) α. The denotation of attributes in a Σ interpretation is governed by the appropriateness function, $\mathcal{F}$. An attribute, α, is defined on all or no entities in the denotation of a species, σ, depending on whether $\mathcal{F}\langle \sigma, \alpha\rangle$ is defined or not, and the value of the partial function denoted by α on an entity of species σ is in the denotation of $\mathcal{F}\langle \sigma, \alpha\rangle$. R interprets each relation symbol as a set of tuples of the form $\langle u_1, \ldots, u_n\rangle$, where: (a) each u_i is an entity in U, or (b) each u_i is a sequence of entities in U, or (c) some u_i are entities and others are sequences of entities in U. The number n is determined by the arity of the relation as given by the arity function.

As witnessed by the cases (b) and (c), arguments of relations may be sequences of entities in U. I call these sequences of entities *chains* (of entities). Technically, chains are strings over U. The syntax of the formal languages of RSRL provides several reserved symbols for the description of chains of entities. For example, we might want to specify the number of entities on a chain, their sorts, or the configurations of entities under the entities on a chain.

The main purpose of chains is to treat certain uses of apparent lists in the HPSG literature where the lists occur in the arguments of relations without being a component of the entities that are described. Only the elements of these "virtual lists" are components of the described entities, but not the apparent lists themselves. A typical relation of HPSG 94 grammars that uses chains in its arguments is the `shuffle` relation of linearization grammars. The relation `shuffle`, a relation between three lists, shuffles the lists in its first two arguments into the list in its third argument, preserving the relative order of their members.[3] In applications of `shuffle` in linguistic principles in the literature, the situation arises that one of the lists in the first two arguments of `shuffle` is the "output" of another relation that specifies that list based on other properties of the described configuration of entities. Then that list may occur nowhere in the configuration of entities under the entities that we wish to describe; it is merely an intermediate result that is shared between the arguments of two relations. This kind of apparent lists in the arguments of relations will be treated as chains, thus obviating the need for otherwise unmotivated book-keeping attributes and attribute values that would reify those lists as components of the described entities, or for a reformulation of the relations that does not reflect the underlying intuitions about the relevant principles of grammar, because it would miss generalizations that are important to the linguist. The linguistic generalizations that motivate chains in grammars will be discussed extensively in Section 4.3 below.

For reasoning with chains, I need to supplement the sort hierarchy, $\langle \mathcal{G}, \sqsubseteq \rangle$, the set of species, $\mathcal{S}$, and the set of attributes, $\mathcal{A}$. The symbols *chain, echain* and *nechain* serve as quasi-sorts for the description of chains, empty chains and nonempty chains, and are ordered in an extension to the sort hierarchy in the intuitive way: The symbols *chain, echain* and *nechain* are the elements of a finite partial order, the *chain hierarchy*, in which *echain* and *nechain* are below *chain*. I call *echain* and *nechain* maximally specific with respect to the chain hierarchy. With the quasi-attributes $\dagger$ and $\triangleright$, chains can be navigated. For the time being, it might be helpful to think of chains as lists, and to see the chain symbols in analogy to the standard symbols to describe lists in HPSG grammars. The quasi sorts *chain, echain* and *nechain* correspond to the sorts *list, elist* and *nelist*, and the quasi-attributes $\dagger$ and $\triangleright$ correspond to the FIRST and REST attributes, respectively. In DEFINITION 49, signatures are augmented by the partial order of quasi-sorts just described, the two

[3]The `shuffle` relation is defined in (64) in Section 4.3.

quasi-attributes, † and $\triangleright$, and another reserved quasi-sort symbol, *metatop*, which is the top of the expanded sort hierarchy:

Definition 49 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$,

$$\widehat{\mathcal{G}} = \mathcal{G} \cup \{chain, echain, nechain, metatop\},$$

$$\widehat{\sqsubseteq} = \sqsubseteq \cup \ \{\langle echain, chain\rangle, \langle nechain, chain\rangle\} \ \cup \left\{ \langle \sigma, \sigma\rangle \ \middle| \sigma \in \widehat{\mathcal{G}} \backslash \mathcal{G} \right\} \cup \left\{ \langle \sigma, metatop\rangle \ \middle| \sigma \in \widehat{\mathcal{G}} \right\},$$

$$\widehat{\mathcal{S}} = \mathcal{S} \cup \{echain, nechain\}, \text{ and}$$

$$\widehat{\mathcal{A}} = \mathcal{A} \cup \{\dagger, \triangleright\}.$$

Note that if $\langle \mathcal{G}, \sqsubseteq \rangle$ is a finite partial order then $\left\langle \widehat{\mathcal{G}}, \widehat{\sqsubseteq} \right\rangle$ is also a finite partial order. I call $\widehat{\mathcal{G}}$ the *expanded sort set*, $\left\langle \widehat{\mathcal{G}}, \widehat{\sqsubseteq} \right\rangle$ the *expanded sort hierarchy*, $\widehat{\sqsubseteq}$ the *expanded sort hierarchy relation*, $\widehat{\mathcal{S}}$ the *expanded species set*, and $\widehat{\mathcal{A}}$ the *expanded attribute set*. Besides the sorts of the signature, $\widehat{\mathcal{G}}$ contains the chain symbols and *metatop*, which, according to the definition of $\widehat{\sqsubseteq}$, is the top element of the expanded sort hierarchy. Moreover, the expanded sort hierarchy relation properly contains the chain hierarchy and the ordering relation of the sort hierarchy. The expanded species set, $\widehat{\mathcal{S}}$, includes the two maximally specific elements in the chain hierarchy, *echain* and *nechain*. For each σ_1 and for each σ_2 in the expanded sort hierarchy, I say that σ_1 *subsumes* σ_2 iff $\sigma_2 \mathrel{\widehat{\sqsubseteq}} \sigma_1$. Finally, the expanded attribute set, $\widehat{\mathcal{A}}$, contains the two reserved quasi-attribute symbols, † and $\triangleright$, besides the elements of the set of attributes, $\mathcal{A}$.

Parallel to the extension of the signature, the species assignment, S, and the attribute interpretation function, A, must be expanded to give the new symbols *echain*, *nechain*, †, and $\triangleright$ a denotation in an interpretation. The analogy of their denotation to the denotation of the normal list symbols *elist, nelist*, FIRST, and REST is obvious:

Definition 50 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,

$\widehat{\mathsf{S}}$ *is the total function from* $\overline{\mathsf{U}}$ *to* $\widehat{\mathcal{S}}$ *such that*

for each $u \in \mathsf{U}$, $\widehat{\mathsf{S}}(u) = \mathsf{S}(u)$,

for each $u_1 \in \mathsf{U}$, ..., *for each* $u_n \in \mathsf{U}$,

$$\widehat{\mathsf{S}}(\langle u_1, \ldots, u_n \rangle) = \begin{cases} \textit{echain} & \textit{if } n = 0, \\ \textit{nechain} & \textit{if } n > 0 \end{cases}, \textit{ and}$$

$\widehat{\mathsf{A}}$ *is the total function from* $\widehat{\mathcal{A}}$ *to the set of partial functions from* $\overline{\mathsf{U}}$ *to* $\overline{\mathsf{U}}$ *such that*

> *for each* $\alpha \in \mathcal{A}$, $\widehat{\mathsf{A}}(\alpha) = \mathsf{A}(\alpha)$,
> $\widehat{\mathsf{A}}(\dagger)$ *is the total function from* U^+ *to* U *such that for each* $\langle u_0, \ldots, u_n \rangle \in \mathsf{U}^+$,
>
> $$\widehat{\mathsf{A}}(\dagger)(\langle u_0, \ldots, u_n \rangle) = u_0, \textit{ and}$$
>
> $\widehat{\mathsf{A}}(\triangleright)$ *is the total function from* U^+ *to* U^* *such that for each* $\langle u_0, \ldots, u_n \rangle \in \mathsf{U}^+$,
>
> $$\widehat{\mathsf{A}}(\triangleright)(\langle u_0, \ldots, u_n \rangle) = \langle u_1, \ldots, u_n \rangle.$$

I call $\widehat{\mathsf{S}}$ the *expanded species assignment function*, and $\widehat{\mathsf{A}}$ the *expanded attribute interpretation function*. For all elements of U in its domain, $\widehat{\mathsf{S}}$ equals the species assignment function, S. In addition, $\widehat{\mathsf{S}}$ assigns the quasi-sort *echain* to the empty chain, and *nechain* to each nonempty chain of entities. Informally, I say that *echain* denotes the singleton set containing the empty chain, and *nechain* denotes the set of nonempty chains. Generalizing the terminology introduced for sorts to the expanded sort set, I say that each symbol, σ, in the expanded sort set denotes the union of the sets of entities or chains denoted by the maximally specific symbols which σ subsumes in the expanded sort hierarchy relation. Therefore, *metatop* denotes all entities and chains of entities of the universe in each interpretation. $\widehat{\mathsf{A}}$ equals the attribute interpretation function, A, for all attributes in its domain; but its domain also includes $\dagger$ and $\triangleright$, and its range includes partial functions whose domains are chains. $\widehat{\mathsf{A}}$ interprets the quasi-attribute $\dagger$ as mapping each nonempty chain to its first element, and the quasi-attribute $\triangleright$ as mapping each nonempty chain, c, to the chain that is just like c but with the first element of c missing.

Since the formal language of RSRL contains variables, variable assignments will be needed to define denotations for its expressions. Variables will occur in the arguments of relational expressions, and the arguments of relations can be entities or chains of entities. Variable assignments must thus be able to assign elements of U and chains of elements of U to variables:

Definition 51 *For each signature* Σ*, for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$*,*

$$Ass_{\mathsf{I}} = \overline{\mathsf{U}}^{\mathcal{VAR}} \text{ is the } \textbf{\textit{set of variable assignments in }} \mathsf{I}.$$

I call each element of Ass_{I} a *variable assignment in* I. Elements of Ass_{I} are functions which assign entities or chains of entities of the universe to variables. Of course, variables will potentially occur in syntactic environments other than the arguments of relations as well, thus giving rise to the possibility of describing chains of entities independent from the description of the arguments of relations.

We are now ready to define the syntax and semantics of the formal languages of RSRL. First, terms and their denotations are defined. The only difference to the terms of SRL is that terms may start with a variable instead of the colon. Second, the terms are used as basic syntactic expressions for constructing complex formulae and their denotation. The syntax of complex formulae will contain the interesting additions to the syntax of SRL, quantification and relations.

Definition 52 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$*,* $\mathcal{T}^{\Sigma}$ *is the smallest set such that*

$: \in \mathcal{T}^{\Sigma}$,

for each $v \in \mathcal{VAR}$, $v \in \mathcal{T}^{\Sigma}$, *and*

for each $\alpha \in \widehat{\mathcal{A}}$ *and each* $\tau \in \mathcal{T}^{\Sigma}$, $\tau\alpha \in \mathcal{T}^{\Sigma}$.

I call each element of $\mathcal{T}^{\Sigma}$ a Σ term. A Σ term consists of either the reserved symbol ':' or a variable followed by a (possibly empty) string of symbols of the expanded attribute set. Σ terms are interpreted as follows:

Definition 53 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$*, for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$*, for each* $ass \in Ass_{\mathsf{I}}$*,* T_{I}^{ass} *is the total function from* $\mathcal{T}^{\Sigma}$ *to the set of partial functions from* U *to* $\overline{\mathsf{U}}$ *such that for each* $u \in \mathsf{U}$*,*

$T_{\mathsf{I}}^{ass}(:)(u)$ *is defined and* $T_{\mathsf{I}}^{ass}(:)(u) = u$,

for each $v \in \mathcal{VAR}$, $T_{\mathsf{I}}^{ass}(v)(u)$ *is defined and* $T_{\mathsf{I}}^{ass}(v)(u) = ass(v)$,

for each $\tau \in \mathcal{T}^{\Sigma}$, *for each* $\alpha \in \widehat{\mathcal{A}}$,

$T_{\mathsf{I}}^{ass}(\tau\alpha)(u)$ *is defined* *iff* $T_{\mathsf{I}}^{ass}(\tau)(u)$ *is defined and* $\widehat{\mathsf{A}}(\alpha)(T_{\mathsf{I}}^{ass}(\tau)(u))$ *is defined, and*

if $T_{\mathsf{I}}^{ass}(\tau\alpha)(u)$ *is defined*

then $T_{\mathsf{I}}^{ass}(\tau\alpha)(u) = \widehat{\mathsf{A}}(\alpha)(T_{\mathsf{I}}^{ass}(\tau)(u))$.

I call T_{I}^{ass} the Σ *term interpretation function with respect to* I *under a variable assignment in* I*, ass*. Each Σ term is interpreted as a partial function from U to $\overline{\mathsf{U}}$. The colon denotes the identity function in U. A variable, v, denotes a constant total function from U to $\overline{\mathsf{U}}$, where each element of U is assigned the entity in U or the chain of entities in U that the variable assignment in I assigns to v. Finally, the meaning of complex Σ terms results from functional composition of the denotation of each of the symbols of the expanded attribute set succeeding the colon or variable in reverse order of their appearance in the Σ term, and the meaning of the colon or variable. Note that only Σ terms which begin with a variable which is followed by a (possibly empty) string of the reserved symbol $\triangleright$ may map elements of U to chains of entities of U. All other terms denote functions which, if they are defined on any entity of the universe at all, may only map them to entities of U. This is easy to see: A Σ term starting with the colon behaves like a term in SRL that only maps elements of U to elements of U. If a Σ term starts with a variable, it can only map an element of U to a chain of entities of U if the variable assignment in I assigns a chain of entities to the variable, because no attribute and no quasi-attribute maps an element of U to a chain of elements of U. Finally, $\triangleright$ is the only element of the expanded attribute set that maps chains of entities of U to chains of entities of U, while † maps chains of entities of U to entities of U, and attributes denote partial functions from U to U. I say that a Σ term, τ, is *defined on an entity*, u, iff τ begins with the colon and $T_{\mathsf{I}}^{ass}(\tau)(u)$ is defined.

Notice that, in case τ begins with a variable, v, the value of $T_{\mathsf{I}}^{ass}(\tau)(u)$, and the question of whether it is defined, is independent of u, because it depends entirely on $ass(v)$. A small example will make this clear. Assume $\mathcal{VAR} = \left\{ v_n \,\middle|\, n \in \mathbb{N} \right\}$, and consider the signature $\Sigma_{3.1}$ in Figure 3.1 (which is the analogue in RSRL to the 87 signature of Figure 2.3 and the SRL signature of Figure 2.4), the $\Sigma_{3.1}$ interpretation $\mathsf{I}_{3.2}$ in Figure 3.2 (which, in essence, repeats the interpretation of Figure 2.6), and the variable assignments in $\mathsf{I}_{3.2}$, ass_0 and ass_1 in (28).

(28) a. $ass_0 = \left\{ \langle v_n, \text{Anne} \rangle \,\middle|\, n \in \mathbb{N} \right\}$

Figure 3.1: The car scenario signature in RSRL

top

car OWNER *person*
DRIVER *person*

vw

bmw

person LIKES-BEST *top*

man

woman

and $\mathcal{R} = \emptyset$, $\mathcal{AR} = \emptyset$.

Figure 3.2: An interpretation of the car scenario signature

Let $\mathsf{I}_{3.2} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, with:

$\mathsf{U} = \{$Anne, Manfred, Maria, Mike, third car$\}$

S(Anne) = *woman*,

S(Maria) = *woman*,

S(Manfred) = *man*,

S(Mike) = *man*,

S(third car) = *vw*, and

A(LIKES-BEST)(Anne) = Manfred,

A(LIKES-BEST)(Manfred) = Anne,

A(LIKES-BEST)(Maria) = Mike,

A(LIKES-BEST)(Mike) = Maria,

A(OWNER)(third car) = Anne,

A(DRIVER)(third car) = Mike,

A(LIKES-BEST) is undefined on the third car,

A(OWNER) is undefined on Anne, Manfred, Maria and Mike,

A(DRIVER) is likewise undefined on Anne, Manfred, Maria and Mike, and

$\mathsf{R} = \emptyset$.

b. $ass_1 = \left\{ \langle v_n, \text{third car} \rangle \;\middle|\; n \in \mathbb{N} \right\}$

Then $T^{ass_0}_{\mathsf{I}_{3.2}}(v_0\text{OWNER})(u)$ is undefined for all $u \in \mathsf{I}_{3.2}$, i.e., for the third car, Anne, Manfred, Maria, and Mike, because $ass_0(v_0) = \text{Anne}$, and the attribute OWNER is not defined on Anne. $T^{ass_1}_{\mathsf{I}_{3.2}}(v_0\text{OWNER})(u)$, on the other hand, is defined for all $u \in \mathsf{I}_{3.2}$, and $T^{ass_1}_{\mathsf{I}_{3.2}}(v_0\text{OWNER})(u) = \text{Anne}$, because $ass_1(v_0) = \text{third car}$, the attribute OWNER is defined on the third car, and $\mathsf{A}(\text{OWNER})(\text{third car}) = \text{Anne}$.

When a Σ term of the form $v\pi$, where $v \in \mathcal{VAR}$ and $\pi \in \mathcal{A}^*$, is part of a formula without free occurrences of variables, component quantification will make the value of applying the denotation of $v\pi$ to an entity u dependent on u. It will guarantee that the assignment is always minimally changed such that v is mapped to a component (or chain of components) of u.

The syntax of formulae is based on terms. The syntax of RSRL's formulae contains relational formulae and quantification in addition to all syntactic constructs of the descriptions of SRL. Unlike definite clause extensions of constraint languages, which add relations in a separate layer to the core constraint language, relational formulae belong to the core description language itself, because I want to use them in the scope of negation and quantification.

Definition 54 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, $\mathcal{D}^\Sigma$ *is the smallest set such that*

for each $\sigma \in \widehat{\mathcal{G}}$, *for each* $\tau \in \mathcal{T}^\Sigma$, $\tau \sim \sigma \in \mathcal{D}^\Sigma$,

for each $\tau_1 \in \mathcal{T}^\Sigma$, *for each* $\tau_2 \in \mathcal{T}^\Sigma$, $\tau_1 \approx \tau_2 \in \mathcal{D}^\Sigma$,

for each $\rho \in \mathcal{R}$, *for each* $x_1 \in \mathcal{VAR}$, ..., *for each* $x_{\mathcal{AR}(\rho)} \in \mathcal{VAR}$,

$\quad \rho(x_1, \ldots, x_{\mathcal{AR}(\rho)}) \in \mathcal{D}^\Sigma$,

for each $x \in \mathcal{VAR}$, *for each* $\delta \in \mathcal{D}^\Sigma$, $\exists x\, \delta \in \mathcal{D}^\Sigma$,

for each $x \in \mathcal{VAR}$, *for each* $\delta \in \mathcal{D}^\Sigma$, $\forall x\, \delta \in \mathcal{D}^\Sigma$,

for each $\delta \in \mathcal{D}^\Sigma$, $\neg\delta \in \mathcal{D}^\Sigma$,

for each $\delta_1 \in \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathcal{D}^\Sigma$, $[\delta_1 \wedge \delta_2] \in \mathcal{D}^\Sigma$,

for each $\delta_1 \in \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathcal{D}^\Sigma$, $[\delta_1 \vee \delta_2] \in \mathcal{D}^\Sigma$,

for each $\delta_1 \in \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathcal{D}^\Sigma$, $[\delta_1 \rightarrow \delta_2] \in \mathcal{D}^\Sigma$, *and*

for each $\delta_1 \in \mathcal{D}^\Sigma$, *for each* $\delta_2 \in \mathcal{D}^\Sigma$, $[\delta_1 \leftrightarrow \delta_2] \in \mathcal{D}^\Sigma$.

I call each element of $\mathcal{D}^{\Sigma}$ a Σ *formula*. Σ formulae of the form $\tau \sim \sigma$ are called *sort assignment formulae*, Σ formulae of the form $\tau_1 \approx \tau_2$ are called *path equations*, and Σ formulae of the form $\rho(x_1, \ldots, x_{\mathcal{AR}(\rho)})$ are called *relational* Σ *formulae*. They constitute the atomic Σ formulae. Complex Σ formulae can be formed from atomic Σ formulae by means of existential and universal quantification and the standard logical connectives. Readers familiar with the conventional linguistic notation of relational expressions might be surprised that variables are the only Σ expressions that are allowed in the argument positions of relational Σ formulae. The only reason for the conservative definition here is the simplicity of the formal languages. The alternative notation of Section 3.2 that is customized for linguistic grammars will be more liberal.

The expressions defined in DEFINITION 54 are deliberately not called descriptions, in contrast to the expressions in the corresponding part of DEFINITION 17 of SRL. I reserve the term "description" for the subset of formulae that will be permitted in the theory of a grammar. The formulae in a theory of a grammar must not contain free occurrences of variables. In SRL grammars, that condition is trivially satisfied, since its expressions do not contain any variables at all. In the present formalism, the expressions given by DEFINITION 54 might contain free variables. In order to be able to say later what a formula without free occurrences of variables is, I define for each Σ term and formula, the set of free occurrences of variables in that Σ term or formula:

Definition 55 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$,

$FV(:) = \emptyset$,

for each $v \in \mathcal{VAR}$, $FV(v) = \{v\}$,

for each $\tau \in \mathcal{T}^{\Sigma}$, *for each* $\alpha \in \widehat{\mathcal{A}}$, $FV(\tau\alpha) = FV(\tau)$,

for each $\tau \in \mathcal{T}^{\Sigma}$, *for each* $\sigma \in \widehat{\mathcal{G}}$, $FV(\tau \sim \sigma) = FV(\tau)$,

for each $\tau_1, \tau_2 \in \mathcal{T}^{\Sigma}$, $FV(\tau_1 \approx \tau_2) = FV(\tau_1) \cup FV(\tau_2)$,

for each $\rho \in \mathcal{R}$, *for each* $x_1, \ldots, x_{\mathcal{AR}(\rho)} \in \mathcal{VAR}$,

$$FV(\rho(x_1, \ldots, x_{\mathcal{AR}(\rho)})) = \{x_1, \ldots, x_{\mathcal{AR}(\rho)}\},$$

for each $\delta \in \mathcal{D}^{\Sigma}$, *for each* $v \in \mathcal{VAR}$, $FV(\exists v\, \delta) = FV(\delta) \backslash \{v\}$, *for each* $\delta \in \mathcal{D}^{\Sigma}$, *for each* $v \in \mathcal{VAR}$, $FV(\forall v\, \delta) = FV(\delta)\backslash\{v\}$,

for each $\delta \in \mathcal{D}^{\Sigma}$, $FV(\neg\delta) = FV(\delta)$,

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$,

$$FV([\delta_1 \wedge \delta_2]) = FV(\delta_1) \cup FV(\delta_2),$$

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$,

$$FV([\delta_1 \vee \delta_2]) = FV(\delta_1) \cup FV(\delta_2),$$

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$,

$$FV([\delta_1 \rightarrow \delta_2]) = FV(\delta_1) \cup FV(\delta_2), \textit{ and}$$

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$,

$$FV([\delta_1 \leftrightarrow \delta_2]) = FV(\delta_1) \cup FV(\delta_2).$$

The definition of free occurrences of variables distinguishes between Σ terms that begin with the colon, which do not contain free variables, and Σ terms that begin with a variable, which is then free in that Σ term.

As motivated in Section 2.2.3, existential and universal quantification in RSRL is quantification over the components of entities. To be able to define quantification as component quantification, I first need to be able to refer to the set of the components of each entity in an interpretation. In other words, I need an analogue in RSRL to the set of components of an entity u in an SRL interpretation I, given as $\mathsf{Co}_{\mathsf{I}}(u)$ in DEFINITION 22. The SRL definition, however, will not do, since it defines $\mathsf{Co}_{\mathsf{I}}(u)$ as the set of entities in the Σ interpretation I that is obtained by collecting the results of applying the denotation of all Σ terms to u. This yields the intended result, because every Σ term of SRL starts with the colon. To achieve the same effect in RSRL, the right subset of all Σ terms must be chosen. The set of Σ terms with which the set of components of an entity is defined must be restricted to those Σ terms that start with the colon, because—as discussed above—the denotation on an entity, u, of Σ terms that start with a variable might yield any entity in the universe of which u is not an ancestor, depending on which entity or chain of entities in I is assigned to the variable by the variable assignment in I.

Definition 56 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, *and for each* $u \in \mathsf{U}$,

$$\mathsf{Co}^{u}_{\mathsf{I}} = \left\{ u' \in \mathsf{U} \;\middle|\; \begin{array}{l} \textit{for some } ass \in Ass_{\mathsf{I}}, \\ \textit{for some } \pi \in \mathcal{A}^*, \\ \quad T^{ass}_{\mathsf{I}}(:\pi)(u) \textit{ is defined, and} \\ \quad u' = T^{ass}_{\mathsf{I}}(:\pi)(u) \end{array} \right\}.$$

I call $\mathsf{Co}^{u}_{\mathsf{I}}$ the *set of components of* u *in* I. Informally, DEFINITION 56 limits the set of components of an entity u in I to those entities in U that are accessible from u by Σ terms defined on u. Note that the denotation of $:\pi$ is independent of the variable assignment in I, since there is no variable in Σ terms of that form.

As a final step towards the definition of the meaning of Σ formulae, I introduce a simple notational convention which I will need to fix the meaning of quantifiers:

Definition 57 *For each signature* Σ, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, *for each* $ass \in Ass_{\mathsf{I}}$, *for each* $v \in \mathcal{VAR}$, *for each* $w \in \mathcal{VAR}$, *for each* $u \in \overline{\mathsf{U}}$,

$$ass\tfrac{u}{v}(w) = \begin{cases} u & \textit{if } v = w \\ ass(w) & \textit{otherwise.} \end{cases}$$

For each variable assignment in I, ass, $ass\frac{u}{v}$ is just like the function ass, except that it assigns entity u to variable v.

Everything is now in place to define the denotation function for Σ formulae, D^{ass}_{I}:

Definition 58 *For each signature* $\Sigma = \langle \mathcal{G}, \sqsubseteq, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R}, \mathcal{AR} \rangle$, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, *for each* $ass \in Ass_{\mathsf{I}}$, D^{ass}_{I} *is the total function from* $\mathcal{D}^{\Sigma}$ *to the power set of* U *such that*

for each $\tau \in \mathcal{T}^{\Sigma}$, *for each* $\sigma \in \widehat{\mathcal{G}}$,

$$D^{ass}_{\mathsf{I}}(\tau \sim \sigma) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} T^{ass}_{\mathsf{I}}(\tau)(u) \textit{ is defined, and} \\ \widehat{\mathsf{S}}(T^{ass}_{\mathsf{I}}(\tau)(u)) \mathrel{\widehat{\sqsubseteq}} \sigma \end{array} \right\},$$

for each $\tau_1 \in \mathcal{T}^{\Sigma}$, *for each* $\tau_2 \in \mathcal{T}^{\Sigma}$,

$$D_{\mathsf{I}}^{ass}(\tau_1 \approx \tau_2) = \left\{ u \in \mathsf{U} \,\middle|\, \begin{array}{l} T_{\mathsf{I}}^{ass}(\tau_1)(u) \textit{ is defined,} \\ T_{\mathsf{I}}^{ass}(\tau_2)(u) \textit{ is defined, and} \\ T_{\mathsf{I}}^{ass}(\tau_1)(u) = T_{\mathsf{I}}^{ass}(\tau_2)(u) \end{array} \right\},$$

for each $\rho \in \mathcal{R}$, *for each* $x_1 \in \mathcal{VAR}$, ..., *for each* $x_{\mathcal{AR}(\rho)} \in \mathcal{VAR}$,

$$\begin{array}{l} D_{\mathsf{I}}^{ass}(\rho(x_1, \ldots, x_{\mathcal{AR}(\rho)})) \\ = \left\{ u \in \mathsf{U} \,\middle|\, \left\langle ass(x_1), \ldots, ass(x_{\mathcal{AR}(\rho)}) \right\rangle \in \mathsf{R}(\rho) \right\}, \end{array}$$

for each $v \in \mathcal{VAR}$, *for each* $\delta \in \mathcal{D}^{\Sigma}$,

$$D_{\mathsf{I}}^{ass}(\exists v\, \delta) = \left\{ u \in \mathsf{U} \,\middle|\, \begin{array}{l} \textit{for some } u' \in \overline{\mathsf{Co}_{\mathsf{I}}^{u}}, \\ u \in D_{\mathsf{I}}^{ass\frac{u'}{v}}(\delta) \end{array} \right\},$$

for each $v \in \mathcal{VAR}$, *for each* $\delta \in \mathcal{D}^{\Sigma}$,

$$D_{\mathsf{I}}^{ass}(\forall v\, \delta) = \left\{ u \in \mathsf{U} \,\middle|\, \begin{array}{l} \textit{for each } u' \in \overline{\mathsf{Co}_{\mathsf{I}}^{u}}, \\ u \in D_{\mathsf{I}}^{ass\frac{u'}{v}}(\delta) \end{array} \right\},$$

for each $\delta \in \mathcal{D}^{\Sigma}$,

$$D_{\mathsf{I}}^{ass}(\neg\delta) = \mathsf{U} \backslash D_{\mathsf{I}}^{ass}(\delta),$$

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$,

$$D_{\mathsf{I}}^{ass}([\delta_1 \wedge \delta_2]) = D_{\mathsf{I}}^{ass}(\delta_1) \cap D_{\mathsf{I}}^{ass}(\delta_2),$$

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$,

$$D_{\mathsf{I}}^{ass}([\delta_1 \vee \delta_2]) = D_{\mathsf{I}}^{ass}(\delta_1) \cup D_{\mathsf{I}}^{ass}(\delta_2),$$

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$,

$$D_{\mathsf{I}}^{ass}([\delta_1 \rightarrow \delta_2]) = (\mathsf{U} \backslash D_{\mathsf{I}}^{ass}(\delta)) \cup D_{\mathsf{I}}^{ass}(\delta_2), \textit{ and}$$

for each $\delta_1 \in \mathcal{D}^{\Sigma}$, *for each* $\delta_2 \in \mathcal{D}^{\Sigma}$,

$$\begin{array}{l} D_{\mathsf{I}}^{ass}([\delta_1 \leftrightarrow \delta_2]) \\ = ((\mathsf{U} \backslash D_{\mathsf{I}}^{ass}(\delta_1)) \cap (\mathsf{U} \backslash D_{\mathsf{I}}^{ass}(\delta_2))) \cup (D_{\mathsf{I}}^{ass}(\delta_1) \cap D_{\mathsf{I}}^{ass}(\delta_2)). \end{array}$$

I call D_{I}^{ass} the Σ *formula interpretation function with respect to* I *under a variable assignment in* I, *ass*.

Notice that conjunction, disjunction, implication and equivalence are interpreted classically; and negation, of course, corresponds to set complement. Notice also the crucial reference to the set comprising components and chains of components of u in I, $\overline{\mathsf{Co}_{\mathsf{I}}^{u}}$, in the interpretation of existential and universal quantification. This is where the restriction to quantification over components and chains of components is made. A Σ formula δ under existential quantification over the variable v, $\exists v\, \delta$, describes an entity u in a Σ interpretation I under the variable assignment *ass* in I, if there is at least one component (or chain of components), u', of u such that u is in the denotation of δ under the variable assignment ass' that we obtain from *ass* by letting it assign the component (or chain of components) u' to the variable v; in the case of universal quantification over the variable v, δ must describe u for all possible minimal changes to *ass* in which a component (or chain of components) of u is assigned to v.

To understand the denotation of sort assignment formulae and path equations, it is useful to distinguish cases when their Σ terms start with the colon from cases when they start with variables. If the Σ terms start with the colon, the denotation of the Σ formulae is just as in SRL. Sort assignment formulae, $\tau \sim \sigma$, in which τ starts with the colon denote the set of entities on which the Σ term τ is defined and leads to an entity which is in the denotation of σ. Path equations, $\tau_1 \approx \tau_2$, in which τ_1 and τ_2 start with the colon denote the set of entities on which τ_1 and τ_2 are defined and lead to the same entity.

The denotation of sort assignment formulae and path equations whose Σ terms start with a variable crucially depends on the chosen variable assignment in I. They either denote the entire universe or the empty set. Because the value of $T_{\mathsf{I}}^{ass}(\tau)(u)$, and the question of whether it is defined, is independent of u if τ starts with a variable, $\tau \sim \sigma$ denotes the entire universe if the function denoted by the string of symbols of the expanded attribute set following the variable is defined on the entity or chain assigned to the variable and leads to an entity or chain in the denotation of the symbol σ; and the empty set otherwise. A Σ formula, $\tau_1 \approx \tau_2$, denotes the entire universe if the functions denoted by the string of symbols of the expanded attribute set following the two variables are defined on the entities or chains that are assigned to the respective variables and lead to the same entity or chain; and the empty set otherwise.

This is best illustrated with an example. Recall the signature $\Sigma_{3.1}$ of Figure 3.1 (page 164), the $\Sigma_{3.1}$ interpretation $\mathsf{I}_{3.2}$ of Figure 3.2 (page 164), and the two variable assignments in $\mathsf{I}_{3.2}$, ass_0 and ass_1, given in (28) on page 163. If we choose the Σ formula interpretation function with respect to $\mathsf{I}_{3.2}$ under the variable assignment ass_0 to interpret the $\Sigma_{3.1}$ formula 'v_0OWNER $\sim$ *woman*', then we get $D^{ass_0}_{\mathsf{I}_{3.2}}(v_0\text{OWNER} \sim \textit{woman}) = \emptyset$, because $ass_0(v_0) =$ Anne, and OWNER is not defined on Anne. If we choose the very same formula interpretation function with respect to the same interpretation $\mathsf{I}_{3.2} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ to interpret the same formula, but this time under the variable assignment ass_1, then we get $D^{ass_1}_{\mathsf{I}_{3.2}}(v_0\text{OWNER} \sim \textit{woman}) = \mathsf{U}$, because ass_1 assigns the third car to the variable v_0, and the owner of the third car is a woman, namely Anne. Similarly for path equations: If we choose the Σ formula interpretation function with respect to $\mathsf{I}_{3.2}$ under the variable assignment ass_0 to interpret the $\Sigma_{3.1}$ formula 'v_0OWNER $\approx$ v_1OWNER LIKES-BEST LIKES-BEST', then we get $D^{ass_0}_{\mathsf{I}_{3.2}}(v_0\text{OWNER} \approx v_1\text{OWNER LIKES-BEST LIKES-BEST}) = \emptyset$, because $ass_0(v_0) =$ Anne and $ass_0(v_1) =$ Anne. For ass_1, we get $D^{ass_1}_{\mathsf{I}_{3.2}}(v_0\text{OWNER} \approx v_1\text{OWNER LIKES-BEST LIKES-BEST}) = \mathsf{U}$, because $ass_1(v_0) =$ third car and $ass_1(v_1) =$ third car, and the owner of the third car (Anne) is the same person as the person that the person liked best by the owner of the third car (Manfred) likes best.

An n-ary relational Σ formula, $\rho(x_1, \ldots, x_n)$, is interpreted as the set of entities such that an n-tuple of entities and chains which is in the relation interpretation of ρ is assigned to the n-tuple of variables, $(x_1, \ldots, x_n)$, describing its arguments. Similar to sort assignment formulae and path equations with Σ terms beginning with variables, a relational Σ formula either denotes the entire universe or the empty set, depending on the variable assignment in I, *ass*. It denotes U if an n-ary tuple of entities or chains (or an n-ary tuple of entities and chains) which is in the relation interpretation is assigned to the tuple of variables that describe its arguments; and the empty set otherwise. This depends, of course, entirely on the choice of the variable assignment function in I, and on nothing else. The reason for this definition of the denotation of relational Σ formulae becomes clear when we consider how the meaning of relations is determined in grammars, and how the definition interacts with component quantification in Σ formulae without free variables.

In grammars, we will restrict ourselves to Σ formulae without free variables (Σ descriptions). In that case, the denotation of relational Σ formulae, sort assignment formulae and path equations involving variables becomes more transparent. When looking at the denotation of Σ descriptions, it is important to remember that in RSRL, just as in SRL, expressions denote a subset of the entities of the universe rather than truth values. This formalizes the intuition underlying both formalisms that their expressions describe entities and are, thus, true *of entities in an interpretation.*

As we have seen in the examples above, the denotation of Σ terms and Σ formulae may depend on the choice of the variable assignment in I under which they are interpreted in I. The important property of Σ descriptions is that their denotation in an interpretation is independent of the choice of the variable assignment in I under which they are interpreted in I. For each $ass \in Ass_{\mathsf{I}}$, they denote the same set of entities. Before I explain the denotation of Σ descriptions, especially the denotation of relations in Σ descriptions, I want to establish this result and define formally what a Σ description is. This is done in PROPOSITION 9, DEFINITION 59, and COROLLARY 2. PROPOSITION 9, which can be verified by a simple inductive proof, states that a Σ formula, δ, denotes the same set of entities under two different variable assignments in I if the two variable assignments agree on the free variables in δ. This observation will ultimately allow us to abstract away from variable assignments when I move from arbitrary Σ formulae to Σ formulae without free variables.

Proposition 9 *For each signature Σ, for each Σ interpretation I, for each $ass_1 \in Ass_{\mathsf{I}}$, for each $ass_2 \in Ass_{\mathsf{I}}$,*

for each $\tau \in \mathcal{T}^{\Sigma}$,

if for each $v \in FV(\tau)$, $ass_1(v) = ass_2(v)$
then $T_{\mathsf{I}}^{ass_1}(\tau) = T_{\mathsf{I}}^{ass_2}(\tau)$, and

for each $\delta \in \mathcal{D}^{\Sigma}$,

if for each $v \in FV(\delta)$, $ass_1(v) = ass_2(v)$
then $D_{\mathsf{I}}^{ass_1}(\delta) = D_{\mathsf{I}}^{ass_2}(\delta)$.

If two variable assignments in I agree on the variables that occur free in a given Σ term, τ, then τ denotes the same partial function under the Σ term interpretation function with respect to I under both variable assignments in I. Similarly, if two variable assignments in I agree on the variables that occur free in a given Σ formula, δ, then δ denotes the same set of entities of the universe under the Σ formula interpretation function with respect to I under both variable assignments in I.

I distinguish the set of Σ formulae without free variables by a special symbol, $\mathcal{D}_0^\Sigma$:

Definition 59 *For each signature Σ,*

$$\mathcal{D}_0^\Sigma = \{\delta \in \mathcal{D}^\Sigma \mid FV(\delta) = \emptyset\}.$$

I call each element of $\mathcal{D}_0^\Sigma$ a Σ *description*, and each subset of $\mathcal{D}_0^\Sigma$ a Σ *theory*. When working with a fixed signature, I will henceforth refer to terms, formulae and descriptions. As a corollary of PROPOSITION 9, we get the following result for the meaning of Σ descriptions:

Corollary 2 *For each signature Σ, for each $\delta \in \mathcal{D}_0^\Sigma$, for each Σ interpretation I, for each $ass_1 \in Ass_\mathsf{I}$, for each $ass_2 \in Ass_\mathsf{I}$,*

$$D_\mathsf{I}^{ass_1}(\delta) = D_\mathsf{I}^{ass_2}(\delta).$$

According to COROLLARY 2 the meaning of Σ descriptions is independent from the initial choice of the variable assignment in I. If there are variables in a Σ description δ, they are bound by a quantifier. Therefore, when a Σ term in δ in which a variable v occurs is interpreted on an entity u in I, the value of the original variable assignment in I is changed on v, and a component of u is assigned to v; what the initial variable assignment in I assigns to v never plays any role. With COROLLARY 2 in hand, I return to the discussion of the denotation of Σ formulae, now focusing on Σ descriptions.

In a complex Σ description, an n-ary relational Σ formula, $\rho(x_1, \ldots, x_n)$, denotes the set of entities whose n-tuples of components or chains of components, as determined by the quantification, are in the denotation of the relation. If all variables in a relational Σ formula are existentially bound, it denotes the set of entities, u, which have an n-tuple of components or chains of components in I that are in the denotation of the relation; if they are all universally bound, it denotes the set of entities, u, all n-tuples of whose components and chains of components in I are in the denotation of the relation. These properties are exploited in defining the meaning of the relation symbols. Note that in a Σ interpretation, the denotation of a relation ρ, $\mathsf{R}(\rho)$, is not in any way predefined. Only the arity of each relation symbol is given by the signature. The meaning of the relation symbols is determined by members of the theory of the grammar, i.e., by appropriate Σ descriptions. The idea is that a relation holds of all appropriate components of all entities in a Σ interpretation according to descriptions in the theory of a grammar. Although there are other possibilities, for each relation symbol in the signature, the theory of a grammar usually contains one bi-implicational description that fixes the meaning of the relation symbol.

To illustrate this, I modify the car scenario signature: I eliminate the attribute LIKES-BEST, and let the new attribute PASSENGERS be appropriate to *car*. The sort *list* is appropriate for $\langle car, \text{PASSENGERS}\rangle$, and the usual sort hierarchy, attributes, and appropriateness conditions for lists are added. The relation that I am interested in is `member`, which is of arity 2. The new signature, $\Sigma_{3.3}$, is shown in Figure 3.3. With $\Sigma_{3.3}$, I can talk about the familiar two kinds of cars, their drivers, owners, and a list of passengers in those cars. With the relation `member`, I want to talk about the members of the passenger list. For that purpose, the symbol `member` must receive the appropriate interpretation, lest any pair of entities and chains can be in its denotation. Now consider the bi-implicational description of the `member` relation in (29):

(29) $$\forall v_1 \, \forall v_2 \left[\texttt{member}(v_1, v_2) \leftrightarrow \left[\begin{array}{l} v_1 \approx v_2\text{FIRST} \\ \vee \, \exists v_3 \, [v_3 \approx v_2\text{REST} \wedge \texttt{member}(v_1, v_3)] \end{array} \right] \right]$$

In any $\Sigma_{3.3}$ interpretation, the description (29) describes exactly each entity which can be characterized as follows: All pairs of components of the entity are in the `member` relation if and only if either the first component is the first element on the list that is the second component, or there is a third component of the entity that equals the tail of the list that is the second component, and the first component and the third component are in the `member` relation (or both defining conditions hold). But that means that if (29) describes all entities in an interpretation, then in that interpretation `member` means what we intuitively want to express with it. For example, consider the $\Sigma_{3.3}$ interpretation $\mathsf{I}_{3.4}$ in Figure 3.4. It is a small scenario in which Mike drives the third car, which is owned by Anne. The persons in the car (the passengers) are Anne, Manfred and Mike. Manfred and Mike are men, and Anne is a woman. (29) describes each entity in $\mathsf{I}_{3.4}$. It is trivially true of Anne, Mike, Manfred, and the empty list, L4, because they do not have proper components and they are not in the member relation

Figure 3.3: A car scenario signature with a relation

top

list

elist

nelist FIRST *top*
REST *list*

car OWNER *person*
DRIVER *person*
PASSENGERS *list*

vw

bmw

person

man

woman

and $\mathcal{R} = \{\texttt{member}\}$, $\mathcal{AR} = \{\langle \texttt{member}, 2\rangle\}$.

Figure 3.4: An interpretation of the car scenario signature with a relation

Let $\mathsf{I}_{3.4} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$, with:

$\mathsf{U} = \{\text{Anne}, \text{Manfred}, \text{Mike}, \text{third car}, \text{L1}, \text{L2}, \text{L3}, \text{L4}\}$,

$$\mathsf{S} = \left\{ \begin{array}{l} \langle \text{Anne}, \textit{woman}\rangle, \langle \text{Manfred}, \textit{man}\rangle, \langle \text{Mike}, \textit{man}\rangle, \langle \text{third car}, \textit{vw}\rangle, \\ \langle \text{L1}, \textit{nelist}\rangle, \langle \text{L2}, \textit{nelist}\rangle, \langle \text{L3}, \textit{nelist}\rangle, \langle \text{L4}, \textit{elist}\rangle \end{array} \right\},$$

$$\mathsf{A} = \left\{ \begin{array}{l} \langle \textsc{driver}, \{\langle \text{third car}, \text{Mike}\rangle\}\rangle, \\ \langle \textsc{owner}, \{\langle \text{third car}, \text{Anne}\rangle\}\rangle, \\ \langle \textsc{passengers}, \{\langle \text{third car}, \text{L1}\rangle\}\rangle, \\ \langle \textsc{first}, \{\langle \text{L1}, \text{Mike}\rangle, \langle \text{L2}, \text{Anne}\rangle, \langle \text{L3}, \text{Manfred}\rangle\}\rangle, \\ \langle \textsc{rest}, \{\langle \text{L1}, \text{L2}\rangle, \langle \text{L2}, \text{L3}\rangle, \langle \text{L3}, \text{L4}\rangle\}\rangle, \end{array} \right\}, \text{ and}$$

$$\mathsf{R}(\texttt{member}) = \left\{ \begin{array}{l} \langle \text{Mike}, \text{L1}\rangle, \langle \text{Anne}, \text{L1}\rangle, \langle \text{Anne}, \text{L2}\rangle, \\ \langle \text{Manfred}, \text{L1}\rangle, \langle \text{Manfred}, \text{L2}\rangle, \langle \text{Manfred}, \text{L3}\rangle \end{array} \right\}.$$

with themselves. It is true of the list L3, because the pair $\langle \text{Manfred}, \text{L3}\rangle$ is in $\mathsf{R}(\texttt{member})$. It is true of the list L2, because $\langle \text{Anne}, \text{L2}\rangle \in \mathsf{R}(\texttt{member})$,

$\langle \text{Manfred}, \text{L2} \rangle \in \mathsf{R}(\texttt{member})$, and $\langle \text{Manfred}, \text{L3} \rangle \in \mathsf{R}(\texttt{member})$. It is true of L1, because $\langle \text{Mike}, \text{L1} \rangle \in \mathsf{R}(\texttt{member})$, $\langle \text{Manfred}, \text{L1} \rangle \in \mathsf{R}(\texttt{member})$, $\langle \text{Anne}, \text{L1} \rangle \in \mathsf{R}(\texttt{member})$, $\langle \text{Anne}, \text{L2} \rangle \in \mathsf{R}(\texttt{member})$, $\langle \text{Manfred}, \text{L2} \rangle \in \mathsf{R}(\texttt{member})$, and $\langle \text{Manfred}, \text{L3} \rangle \in \mathsf{R}(\texttt{member})$. Finally, it is true of the third car for the same reason for which it is true of L1. In short, in $\mathsf{I}_{3.4}$ the `member` relation has the intuitively right meaning. Therefore, descriptions using `member` that are interpreted with respect to $\mathsf{I}_{3.4}$ under some variable assignment, *ass*, also have the intuitively right meaning relative to our little scenario. For example, the third car is in the set of entities $D^{ass}_{\mathsf{I}_{3.4}}([: \sim vw \wedge \exists v_1 \exists v_2\, [v_2 \approx :\text{PASSENGERS} \wedge \texttt{member}(v_1, v_2)]])$, because there are indeed passengers in the third car, and the third car is also in $D^{ass}_{\mathsf{I}_{3.4}}([: \sim vw \wedge \exists v_1 \exists v_2\, [: \approx v_1 \wedge v_2 \approx :\text{PASSENGERS} \wedge \neg\, \texttt{member}(v_1, v_2)]])$, because the third car is not a passenger of itself. '$[: \sim vw \wedge \exists v_1\, [v_1 \approx :\text{PASSENGERS} \wedge \forall v_2\, [\texttt{member}(v_2, v_1) \rightarrow v_2 \sim man]]]$', however, does not describe the third car in $\mathsf{I}_{3.4}$, because every passenger of the third car is not a man.

Now consider the $\Sigma_{3.3}$ interpretation that is just like $\mathsf{I}_{3.4}$ but with the relation interpretation R', which is exactly like R but with $\langle \text{Anne}, \text{L1} \rangle$ missing from the denotation of `member`. Call the resulting $\Sigma_{3.3}$ interpretation $\mathsf{I}'_{3.4}$. In $\mathsf{I}'_{3.4}$, (29) no longer describes every entity. It still describes Anne, Manfred, Mike, the empty list, L4, the nonempty list L3, and the nonempty list L2. Since $\langle \text{Anne}, \text{L1} \rangle$ is not in $\mathsf{R}'(\texttt{member})$, it no longer describes L1 and the third car. It follows that the denotation of some descriptions using the `member` relation receive an intuitively wrong denotation when interpreted in $\mathsf{I}'_{3.4}$. For example, for any variable assignment *ass* in $\mathsf{I}'_{3.4}$, $D^{ass}_{\mathsf{I}'_{3.4}}([: \sim vw \wedge \exists v_1\, [v_1 \approx :\text{PASSENGERS} \wedge \forall v_2\, [\texttt{member}(v_2, v_1) \rightarrow v_2 \sim man]]])$ describes the third car, although Anne is among the passengers, and Anne is a woman.

The denotation of relational formulae in an interpretation is the most complex aspect of the semantics of the formal languages of RSRL, because their denotation only makes intuitive sense relative to interpretations in which the descriptions that fix the meaning of the relations describe every entity. Moreover, the variables in the formula must be bound by quantification, because otherwise the choice of the variable assignment alone determines their denotation in the interpretation, just as in the examples of sort assignment formulae and path equations with free variables above.

As with the denotation of relational formulae, understanding the denotation in interpretations of sort assignment formulae and path equations becomes very easy when we limit our attention to formulae that are part of descriptions. If the variable in the Σ term, τ, of a sort assignment formula, $\tau \sim \sigma$ is existentially bound, the sort assignment formula denotes the set of entities which have at least one component or chain of components, c, such that the function denoted by the string of symbols from the expanded attribute set following the variable in τ is defined on c and yields an entity or chain in the denotation of σ. To come back to an earlier example from page 171, if the $\Sigma_{3.1}$ formula '$v_0\text{OWNER} \sim woman$' occurs as a sub-formula of the $\Sigma_{3.1}$ description '$\exists v_0\, v_0\text{OWNER} \sim woman$', it describes exactly the third car in the $\Sigma_{3.1}$ interpretation $\mathsf{I}_{3.2}$, irrespective of the variable assignment in $\mathsf{I}_{3.2}$ under which it is evaluated, i.e., $D^{ass_0}_{\mathsf{I}_{3.2}}(\exists v_0\, v_0\text{OWNER} \sim woman) = D^{ass_1}_{\mathsf{I}_{3.2}}(\exists v_0\, v_0\text{OWNER} \sim woman) = \{\text{third car}\}$, where ass_0 and ass_1 are as defined in (28). The third car is the only entity in $\mathsf{I}_{3.2}$ that has a component whose owner is a woman, and that component is the third car itself. If the variables in the Σ terms τ_1 and τ_2 of a path equation, $\tau_1 \approx \tau_2$, are existentially bound, that path equation denotes the set of entities such that there is at least one component or chain of components, c_1, on which the function denoted by the string of symbols from the expanded attribute set following the variable in τ_1 is defined, there is at least one component or chain of components, c_2, on which the function denoted by the string of symbols from the expanded attribute set following the variable in τ_2 is defined, and the two functions have the same entity or chain as value on c_1 and c_2, respectively. Corresponding observations hold for universal quantification.

I have shown that determining the denotation of Σ descriptions in an interpretation is independent of the variable assignments in the interpretation. Therefore, the Σ description denotation function in an interpretation does not need to refer to variable assignments in the interpretation:

Definition 60 *For each signature* Σ*, for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$*,* D_{I} *is the total function from* $\mathcal{D}^{\Sigma}_0$ *to the power set of* U *such that for each* $\delta \in \mathcal{D}^{\Sigma}_0$*,*

$$D_{\mathsf{I}}(\delta) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \textit{for each } ass \in Ass_{\mathsf{I}}, \\ u \in D^{ass}_{\mathsf{I}}(\delta) \end{array} \right\}.$$

I call D_{I} the Σ *description denotation function with respect to* I. D_{I} abstracts away from particular variable assignments in I. As a result, description denotation in RSRL is completely parallel to description denotation in SRL, and the following definitions of theory denotations, grammars, models of grammars and exhaustive models of grammars are also parallel to the corresponding definitions in SRL. With D_{I} in hand it is straightforward to define the denotation of Σ theories:

Definition 61 *For each signature* Σ, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, Θ_{I} *is the total function from the power set of* $\mathcal{D}_0^{\Sigma}$ *to the power set of* U *such that for each* $\theta \subseteq \mathcal{D}_0^{\Sigma}$,

$$\Theta_{\mathsf{I}}(\theta) = \left\{ u \in \mathsf{U} \,\middle|\, \begin{array}{l} \text{for each } \delta \in \theta, \\ u \in D_{\mathsf{I}}(\delta) \end{array} \right\}.$$

I call Θ_{I} the *theory denotation function with respect to* I. Under Θ_{I} a Σ theory, θ, denotes the set of entities of the universe which are in the denotation of all Σ descriptions in θ. Just as in SRL, grammars are pairs of a signature Σ and a set of Σ descriptions:

Definition 62 Γ *is a* ***grammar*** *iff*

> Γ *is a pair* $\langle \Sigma, \theta \rangle$,
>
> Σ *is a signature, and*
>
> $\theta \subseteq \mathcal{D}_0^{\Sigma}$.

A grammar is a signature, Σ, together with a set of Σ descriptions, θ. As we know from SRL, not every Σ interpretation is an interesting interpretation of a grammar, $\langle \Sigma, \theta \rangle$. For example, the ungrammatical sentence *Sandy walks Kim Fido* could very well be in an interpretation of the grammar of Pollard and Sag 1994, but that interpretation does certainly not reflect the predictions the authors make about English. In RSRL, we have seen an additional technical reason for why arbitrary interpretations cannot be what a grammarian has in mind. The relational symbols only obtain the denotation that is expressed in the descriptions in the theory of the grammar that are meant to fix their meaning if those descriptions are true of every entity in the interpretation. In interpretations in which that is not the case, descriptions that employ the relational symbols might not have the intended meaning. Consequently, the models of a grammar are a first approximation to the interpretations in which linguists are interested when they write grammars:

Definition 63 *For each grammar* $\Gamma = \langle \Sigma, \theta \rangle$, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,

I *is a* Γ ***model*** *iff* $\Theta_{\mathsf{I}}(\theta) = \mathsf{U}$.

In a Γ model, each description in the theory is true of each entity in the universe of the interpretation. Informally, the grammar denotes the whole universe of the interpretation. As pointed out before, the notion of a model formalizes the idea that with a grammar linguists want to characterize natural language as a set of entities which, in all their components, obey every principle of the grammar. However, as discussed in the previous chapter, arbitrary models cannot characterize natural languages adequately. In a sense, a model of a grammar might not be big enough to be the intended model, since it might not contain a configuration under an entity to which there exists an indiscernible configuration under some other entity in a different model of the grammar: One model of a grammar might lack a sentence that is contained in a different model of the grammar. A trivial example for an uninteresting model of a grammar is the model where U is the empty set, if the grammar in question has models with a nonempty universe. A less trivial example of a model that is too small is a model of Pollard and Sag's (1994) grammar which contains exactly the sentence *Kim likes bagels* and nothing else.

In SRL, the class of exhaustive models delimits the models that are big enough in the sense that they contain indiscernible counterparts to all possible configurations under an entity that can occur in some model of the grammar. The goal is to transfer that idea to RSRL. If this can be done then King's three conditions that say what it means for a grammar to be true of a natural language apply to RSRL grammars. This would explicate the meaning of a grammar, and I could define the tokens and types of a natural language as they are defined in SRL. From there it is only a small step to reconstruct the other accounts of the meaning of HPSG grammars of Pollard and Sag 1994 and of Pollard 1999.

Since all relevant notions of SRL have a counterpart in RSRL, it is easy now to rephrase the characterization of the class of exhaustive models that is based on description denotation alone in RSRL (cf. COROLLARY 1):

Definition 64 *For each grammar* $\Gamma = \langle \Sigma, \theta \rangle$, *for each* Σ *interpretation* I,

I *is an* ***exhaustive*** Γ ***model*** *iff*

I *is a* Γ *model, and*
for each $\theta' \subseteq \mathcal{D}_0^{\Sigma}$, *for each* Σ *interpretation* I',

if I' *is a* Γ *model and* $\Theta_{\mathsf{I}'}(\theta') \neq \emptyset$ *then* $\Theta_{\mathsf{I}}(\theta') \neq \emptyset$.

An exhaustive Γ model, I, is a Γ model such that any theory which has a nonempty denotation in any other Γ model also has a nonempty denotation in I. Recall that an exhaustive Γ model contains indiscernible counterparts to all configurations under an entity that occur in some model of Γ. Among those entities are, of course, all sentences that are licensed by Γ. In linguistic terms this means that a grammar is true of a natural language if the natural language is an exhaustive model of the grammar.

The relevant question to ask now is if all RSRL grammars have exhaustive models. King has shown that every SRL grammar has an exhaustive model; and in Section 2.2.2.4, I spelled out a proposal by Pollard 1999 of how to construct a canonical exhaustive model for each SRL grammar. Note that every SRL grammar can trivially be stated as an RSRL grammar; and that subset of RSRL grammars will thus have exhaustive models. However, the syntactic and semantic additions to SRL that are contained in RSRL might in principle cause RSRL grammars not to have exhaustive models in the general case. In Section 3.1.2, I will outline a proof of the following theorem that lays these apprehensions to rest:

Theorem 2 *For each signature* Σ*, for each* Σ *theory* θ*, there exists a* Σ *interpretation* I *such that* I *is an exhaustive* $\langle \Sigma, \theta \rangle$ *model.*

As a corollary of Theorem 2, we get immediately that if an RSRL grammar has a nonempty model, it also has a nonempty exhaustive model. In summation, a non-vacuous grammar has a linguistically sensible meaning in the class of its exhaustive models. King's explanation of the meaning of grammars, summarized in Section 2.2.2.2, *mutatis mutandis*, carries over to RSRL.

In Section 2.2.3, I have argued that it follows from the omission of true relations and of component quantification from the logical languages of SRL that SRL grammars are not capable of expressing many typical principles of HPSG 94 grammars directly and faithfully. The extension to SRL that I proposed in this section comprises a number of additional new devices in its formal apparatus besides relations and component quantification. Most notably, I introduced variables, variable assignments in interpretations, and sets of components of entities in interpretations, and I made a distinction between formulae with free occurrences of variables and descriptions, in which all variables are bound. Moreover, for independent reasons I generalized component quantification to quantification over components and chains of components.

It is the reference to components and chains of components of entities in the definition of the meaning of the two quantifiers of RSRL (DEFINITION 58) where our quantification differs from the quantification of first-order logic. As a consequence of that definition, the relations of RSRL grammars are relations between the components (and chains of components) of entities, because the meaning of the relation symbols of grammars is typically fixed by principles of grammar of a form that I have exemplified with the `member` relation in (29), and because the meaning of relation principles of that form crucially depends on universal component quantification. In an example, we have seen that this approach results in an intuitively correct meaning of relations and negated relational formulae in models of RSRL grammars. The relational extension of SRL that I propose is thus intimately linked to component quantification, and component quantification is the most important innovation that distinguishes RSRL from other extensions of feature logic.

3.1.2 Constructing a Canonical Exhaustive Model

One method to prove that every grammar has an exhaustive model is to construct a particular model of each grammar, and to show that this model is an exhaustive model. In this section, I will provide such a construction and outline the structure of a proof that shows that the models that I will construct have the desired properties.[4] I will strictly focus on those aspects of the construction and the proof that are relevant in order to understand how the different perspectives on the meaning of an HPSG 94 grammar can be captured in RSRL. I omit all mathematical details that are not essential for that purpose and might only distract from the general line of the argument. However, to satisfy the need for explicitness, the details of the proof are provided separately in Appendix A and can be checked there.

In DEFINITION 64, I defined the exhaustive models of a grammar using the properties of the denotation of the theory of the grammar with respect to an exhaustive model relative to the denotation of the theory with respect to other models of the grammar. My proof of THEOREM 2 about the existence of an exhaustive model of each grammar, however, exploits the algebraic

[4]The proof is due to Richter and King 1997.