

Some Remarks on the Decidability of the Generation Problem in LFG- and PATR-Style Unification Grammars

Jürgen Wedekind
Institute for Natural Language Processing
University of Stuttgart
Azenbergstr. 12
D-70174 Stuttgart, FRG
juergen@ims.uni-stuttgart.de

Abstract

In this paper, we prove the decidability of the generation problem for those unification grammars which are based on context-free phrase structure rule skeletons, like e.g. LFG and PATR-II. The result shows a perhaps unexpected asymmetry, since it is valid also for those unification grammars whose parsing problem is undecidable, e.g. grammars which do not satisfy the off-line parsability constraint. The general proof is achieved by showing that the space of the derivations which have to be considered in order to decide the problem for a given input is always restricted to derivations whose length is limited by some fixed upper bound which is determined relative to the “size” of the input.

1 Introduction

Unification Grammars with a context-free skeleton, like Lexical Functional Grammar (LFG) and PATR-II (cf. e.g. Kaplan and Bresnan 1982, Shieber et al. 1983) assign to a sentence not only a constituent structure (c-structure), but also an additional linguistic entity. In the rather restricted grammars of the early stage this entity is identified with a special graph structure, commonly called *feature structure*. Since a string is regarded as well-formed only if a (well-formed) feature structure is assigned to it by the grammar, two inverse decidability problems arise which had to be solved in order to know whether we can formulate terminating parsing and generation algorithms. If we retain the terminology of the early stages then an adequate parsing algorithm requires that we can decide for a given grammar and a given string whether there exists a feature structure assigned to it by the grammar (parsing problem) and an adequate generation algorithm requires that we

can decide for a given grammar and a given feature structure whether there exists a sentence to which this structure is assigned by the grammar (generation problem).

While we already know for a long time that the parsing problem is undecidable (cf. Kaplan and Bresnan 1982, Johnson 1988), we want to show in this paper that the generation problem is decidable even for unrestricted (not off-line parsable) unification grammars. For the proof we first introduce in section 2 the type of grammar we want to consider. In section 3 we then define the generation problem and show its decidability in two steps.

2 Preliminaries

The unification grammars we want to consider consist of rules with a context-free skeleton and a set of annotations associated with the constituents mentioned in the rules. Typical examples taken from LFG and PATR-II are given in figure 1. For the for-

$S \rightarrow$	NP	VP	$S \rightarrow NP VP$
	$(\uparrow \text{SUBJ}) = \downarrow$	$\uparrow = \downarrow$	$(VP \text{ AGR}) = (NP \text{ AGR})$
$NP \rightarrow$	$John$	$NP \rightarrow Uther$	
	$(\uparrow \text{PRED}) = JOHN$	$(NP \text{ AGR NUM}) = SG$	$(NP \text{ AGR PER}) = 3RD$

Figure 1
Examples of rules in LFG (left) and PATR-II format (right).

mal definition of those grammars we reconstruct the annotations as formulas of a quantifier-free sublanguage of a classical first-order language with equality whose (nonlogical) symbols are given by a finite set of unary partial function symbols and a finite set of constants. For the translation of LFG and PATR-II annotations we regard the *attributes* (in figure 1: SUBJ, PRED, AGR, NUM, PER) as *unary partial function symbols* and the *atomic values* (in figure 1: JOHN,

SG, 3RD) as *individual constants*. Furthermore, we assume for a context-free rule of the form $A \rightarrow w$ ($w \in (V_N \cup V_T)^*$) that the variable x_0 is associated with A and that for each occurrence w_i in w there is a variable x_i which is associated with w_i . For the formal reconstruction of LFG's we assume that each occurrence of \downarrow in the annotation of w_i corresponds to an occurrence of x_i and that each occurrence of \uparrow corresponds to an occurrence of x_0 . For grammars in PATR-II format we suppose that occurrences of categories in the annotations correspond to the associated variables.

Before we give the definition of the grammars we want to investigate, we introduce the following notation. In the following we use $S[x_1, \dots, x_n]$ to indicate that the variables occurring in the set of formulas S are included in $\{x_1, \dots, x_n\}$ and $S(x_1, \dots, x_n)$ if the set of variables occurring in S is exactly $\{x_1, \dots, x_n\}$.

1. DEFINITION. A *unification grammar* is a tuple $\langle V_N, V_T, S, F_1, V, \mathcal{V}, R \rangle$, consisting of a finite nonterminal vocabulary V_N , a finite terminal vocabulary V_T , a start symbol $S \in V_N$ and a *feature-description language* L determined by a finite set of unary partial function symbols F_1 , a finite set of atomic values V and a denumerable set of variables¹

$$\mathcal{V} = \{x_\sigma \mid \sigma \in \mathbb{N}^*\} \text{ with } x_\sigma \neq x_{\sigma'} \text{ for } \sigma \neq \sigma'.$$

All vocabularies are pairwise disjoint. R is a finite set of rules of the form $r = \langle \langle A, w \rangle, S_r[x_0, \dots, x_{|w|}] \rangle$ ($x_i \in \mathcal{V}$), with $\langle A, w \rangle \in V_N \times (V_N \cup V_T)^*$ (a context-free phrase structure rule) and $S_r[x_0, \dots, x_{|w|}]$ a finite set of (quantifier-free) literals of L .²

According to our definition the LFG rules in figure 1 are now expressed as depicted in (1a) and the PATR-II rules as given in (1b). Note that the structure of the terms is now "mirror imaged", since we assume the attributes to be unary partial function symbols.

- (1) (a) $\langle S \rightarrow NP VP, \{SUBJ x_0 \approx x_1, x_0 \approx x_2\} \rangle$
 $\langle NP \rightarrow John, \{PRED x_0 \approx JOHN\} \rangle$
 (b) $\langle S \rightarrow NP VP, \{AGR x_2 \approx AGR x_1\} \rangle$
 $\langle NP \rightarrow Uther, \{NUM AGR x_0 \approx SG, \{PER AGR x_0 \approx 3RD\} \} \rangle$

For the definition of the *sentences* derivable by a unification grammar we have to specify first what *derivations* are.

2. DEFINITION. A sequence of pairs $\pi_0 \dots \pi_n$ with $\pi_0 = \langle B_\emptyset, \emptyset \rangle$ ($B \in V_N$) is called *derivation* of length n iff for each $\pi_i = \langle B[\dots A_{\mu \dots}], S \rangle$ ($0 \leq i < n$) there is a rule $r = \langle A \rightarrow w_1^1 \dots w_m^k, S_r \rangle$ such that

$$\pi_{i+1} = \langle B[\dots A[w_{\mu.1}^1, \dots, w_{\mu.m}^k]_{\mu \dots}], S_r \rangle.$$

In the definition we assume that the order of the arcs of a tree is encoded by numbering the arcs and that each node is identified with the sequence of integers numbering the arcs along the path from the

¹The syntax and semantics of feature-description languages is given in the appendix.

²A literal is an atomic formula or the negation of an atomic formula.

root (\emptyset) to that node. In our bracket notation we add to a constituent its root node as the right and its root node label as the left index. In order to be able to refer to the c-structure derivation and to the sequence of feature descriptions and to have access to the nodes which are substituted in each step of a derivation, we define for a derivation π three other sequences.

3. DEFINITION. Let π be a derivation of length n . We then define two sequences ω and γ for each $i = 0, \dots, n$ with $\pi_i = \langle T_c, S \rangle$ by $\omega_i = T_c$ and $\gamma_i = S$ and a sequence ϖ for each $i = 1, \dots, n$ with $\omega_{i-1} = B[\dots A_{\mu \dots}]_\emptyset$ and $T_c = B[\dots A[w_{\mu.1}^1, \dots, w_{\mu.m}^k]_{\mu \dots}]_\emptyset$ by $\varpi_i = \mu$.

Let S be a set of literals and θ a unary partial mapping over the set of terms. Then the expression $S[\theta]$ denotes the set of expressions obtained from S by simultaneously replacing each occurrence of a term τ in each formula in S by $\theta(\tau)$. The feature description derived by π is then defined by means of the following operation.

4. DEFINITION. If π is a derivation of length n then the *feature description derived by π from h to k* ($0 \leq h < k \leq n$) is given by

$$S_{h \rightarrow k}^\pi = \bigcup_{i=h}^k \gamma_i[\{\langle x_j, x_{\varpi_i.j} \rangle \mid x_j \text{ occurs in } \gamma_i\}].$$

EXAMPLE 1. If we start a derivation π from $\langle S_\emptyset, \emptyset \rangle$ and apply the S-rule in (1a) and the following VP-rule

$$\langle VP \rightarrow V VP', \{x_0 \approx x_1, XCOMP x_0 \approx x_2\} \rangle$$

we end up with the following sequence.

$$\begin{aligned} \pi_0 &= \langle S_\emptyset, \emptyset \rangle \\ \pi_1 &= \langle S[NP_1, VP_2]_\emptyset, \{SUBJ x_0 \approx x_1, x_0 \approx x_2\} \rangle \\ \pi_2 &= \langle S[NP_1, VP[V_{2.1}, VP'_{2.2}]_2]_\emptyset, \{x_0 \approx x_1, XCOMP x_0 \approx x_2\} \rangle \\ &\vdots \end{aligned}$$

For the steps depicted above the sequence ϖ is given by $\varpi_1 = \emptyset$ and $\varpi_2 = 2$ and the feature description derived by π from 0 to 2 ($S_{0 \rightarrow 2}^\pi$) is

$$\{SUBJ x_0 \approx x_1, x_0 \approx x_2, x_2 \approx x_{2.1}, XCOMP x_2 \approx x_{2.2}\}.$$

Sentences are then defined as follows.

5. DEFINITION. A terminal string w ($w \in V_T^*$) is a *sentence* iff there is a derivation $\langle S_\emptyset, \emptyset \rangle = \pi_0 \dots \pi_n$ with $\omega_n = S[w]_\emptyset$ and $\exists x_{\mu_1} \dots x_{\mu_m} \wedge S_{0 \rightarrow n}^\pi(x_{\mu_1}, \dots, x_{\mu_m})$ satisfiable.³

In the following we write S^π for $S_{0 \rightarrow n}^\pi$ if the interval covers the whole derivation, i.e. if π is of length n .

Since a specific reduction algorithm and a few model-theoretic facts required in the proofs later on can be introduced by showing how satisfiability of such existential prenex formulas can be decided, we will continue with a short excursion on satisfiability.

³We use $S[w]_\emptyset$ to denote an S-rooted c-structure with yield w .

2.1 Satisfiability

In order to test whether for a given finite set of literals S of a feature-description language (2)

$$(2) \exists x_1 \dots x_l \wedge S(x_1, \dots, x_l)$$

is satisfiable, we can exploit by skolemization well-known test procedures available for quantifier- and variable-free sets of such literals. Let C be a set of Skolem-constants ($|\{x_1, \dots, x_l\}| = |C|$) and θ be a bijective function from $\{x_1, \dots, x_l\}$ to C , then (2) can be tested by testing the set of literals (3) over $L(C)$ ⁴

$$(3) S[\theta],$$

since (2) and (3) are equi-satisfiable. In the following we complete the procedure by introducing a reduction algorithm that reduces a set of literals (3) according to a measure in a sequence of measure decreasing rewrite steps to a deductively equivalent set (4) (in reduced form)

$$(4) (S[\theta])_\rho,$$

which is satisfiable iff the terms τ of all inequalities $\tau \not\approx \tau$ of (4) do not occur as subterms in equations of (4).⁵

For the proof we first introduce a few definitions and some notation. Let \mathcal{T} be the set of terms of a variable-free feature-description language $L(C)$. Then an injective function $m \in [\mathcal{T} \mapsto \mathbb{N}^*]$ is a *measure* iff it satisfies the following conditions for all $\tau, \tau' \in \mathcal{T}$ and $\sigma \in F_1^*$:

- (i) if $|\tau| < |\tau'|$, then $m(\tau) < m(\tau')$,
- (ii) if $m(\tau) \leq m(\tau')$, then $m(\sigma\tau) \leq m(\sigma\tau')$.

For literals and sets of literals S we extend a measure m as usual by $m((\sim)\tau \approx \tau') = m(\tau) + m(\tau')$ and $m(S) = \sum_{\phi \in S} m(\phi)$.

In the following we use $\tau \approx \tau'$ iff $m(\tau) > m(\tau')$ and $\tau \approx \tau'$ to denote ambiguously $\tau \approx \tau'$ or $\tau' \approx \tau$. Let S be a set of literals then E denotes the set of all equations in S , \mathcal{T}_S the set of terms occurring in the formulas of S ($\mathcal{T}_S = \{\tau, \tau' \mid (\sim)\tau \approx \tau' \in S\}$) and $\text{SUB}(\mathcal{T}_S)$ the set of all *subterms* of the terms in \mathcal{T}_S

$$\text{SUB}(\mathcal{T}_S) = \{\tau \mid \sigma\tau \in \mathcal{T}_S, \text{ with } \sigma \in F_1^*\}.$$

For the construction of a reduced form we need a specific partial choice function ρ which satisfies

$$\rho(S) \in \{\tau \approx \tau' \in S \mid \tau \in \text{SUB}(\mathcal{T}_S \setminus \{\tau \approx \tau'\})\}$$

if the specified set is nonempty and undefined otherwise.

6. DEFINITION. For a given finite set of literals S and a choice function ρ we define a sequence of sets S_{ρ_i} ($i \geq 0$) by induction:

$$S_{\rho_0} = S$$

$$S_{\rho_{i+1}} = \begin{cases} (S_{\rho_i} \setminus \{\tau \approx \tau'\})[\tau/\tau'] \cup \{\tau \approx \tau'\} & \text{if } \rho(S_{\rho_i}) = \tau \approx \tau' \\ S_{\rho_i} & \text{if } \rho(S_{\rho_i}) \text{ undef.} \end{cases}$$

⁴The feature-description language which in addition to L provides a distinct set of Skolem-constants C . Cf. the appendix for more details.

⁵The algorithm is adapted from Statman 1977 and Knuth and Bendix 1970 and first applied to feature-description languages by Beierle and Pletat (1988).

Since $m(S_{\rho_i}) > m(S_{\rho_{i+1}})$ if ρ is defined for S_{ρ_i} , the construction terminates with a finite set of literals. If we set

$$S_\rho = S_{\rho_t}; \text{ with } t = \min\{i \mid S_{\rho_i} = S_{\rho_{i+1}}\}$$

the following lemma can easily be proven by induction on the construction of S_ρ .⁶

7. LEMMA. For S_ρ it holds that:

- (i) $S \Vdash S_\rho$,
- (ii) if $\tau \approx \tau' \in S_\rho$ then $\tau \notin \text{SUB}(\mathcal{T}_{S_\rho \setminus \{\tau \approx \tau'\}})$.

Since S_ρ is obviously not satisfiable if it contains an inequality $\tau \not\approx \tau$ and τ occurs as a subterm in E_ρ , the whole proof is completed by showing that we can construct a canonical model satisfying S_ρ if S_ρ does not contain such an inequality.

For the model construction we need the set

$$\mathcal{T}_{E_\rho}^c = \{\tau \in \text{SUB}(\mathcal{T}_{E_\rho}) \mid \neg \exists \tau' (\tau \approx \tau' \in E_\rho)\}$$

and the function $h^c \in [\text{SUB}(\mathcal{T}_{E_\rho}) \mapsto \mathcal{T}_{E_\rho}^c]$ which is defined for each $\tau \in \text{SUB}(\mathcal{T}_{E_\rho})$ by

$$h^c(\tau) = \begin{cases} \tau' (\tau \approx \tau' \in E_\rho) & \text{if } \tau \notin \mathcal{T}_{E_\rho}^c \\ \tau & \text{otherwise.} \end{cases}$$

That h^c is well-defined results of course from 7(ii).

8. DEFINITION. For a set of literals S_ρ the *canonical term model* is given by the pair $M_\rho = \langle \mathcal{U}_\rho, \mathfrak{S}_\rho \rangle$, consisting of the universe

$$\mathcal{U}_\rho = \begin{cases} \mathcal{T}_{E_\rho}^c & \text{if } E_\rho \neq \emptyset \\ \{\emptyset\} & \text{otherwise} \end{cases}$$

and the interpretation function \mathfrak{S}_ρ , which is defined for $c \in V \cup C$, $f \in F_1$ and $\tau \in \mathcal{U}_\rho$ by:

$$\mathfrak{S}_\rho(c) = \begin{cases} h^c(c) & \text{if } c \in \text{SUB}(\mathcal{T}_{E_\rho}) \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\mathfrak{S}_\rho(f)(\tau) = \begin{cases} h^c(f\tau) & \text{if } f\tau \in \text{SUB}(\mathcal{T}_{E_\rho}) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

For M_ρ which is well-defined the following lemma holds:

9. LEMMA. If τ is a subterm of \mathcal{T}_{S_ρ} then

- (i) $\mathfrak{S}_\rho(\tau) = h^c(\tau)$, if $\tau \in \text{SUB}(\mathcal{T}_{E_\rho})$,
- (ii) $\tau \in \text{SUB}(\mathcal{T}_{E_\rho})$, if $\tau \in \text{Dom}(\mathfrak{S}_\rho)$.

PROOF. (By induction on the length of τ .) The lemma is trivial for constants. By showing (i) before (ii) we get the induction step for a subterm $f\tau$ of \mathcal{T}_{S_ρ} in both cases according to

$$\mathfrak{S}_\rho(f\tau) = \mathfrak{S}_\rho(f)(\mathfrak{S}_\rho(\tau)) = \mathfrak{S}_\rho(f)(h^c(\tau)) = \mathfrak{S}_\rho(f)(\tau).$$

We get $\mathfrak{S}_\rho(\tau) = h^c(\tau)$ by inductive hypothesis and $h^c(\tau) = \tau$, since $\tau \notin \mathcal{U}_\rho$ would imply the existence of

⁶In order to verify 7(i) cf. e.g. Wedekind 1991 and 1994.

$\tau \approx \tau' \in E_\rho$ and $f\tau$ could not be a subterm of \mathcal{T}_{S_ρ} according to lemma 7(ii). Now, if (i) $f\tau \in \text{SUB}(\mathcal{T}_{E_\rho})$ then $\mathfrak{F}_\rho(f)(\tau)$ is defined and equal to $h^c(f\tau)$ and (ii) if $f\tau \in \text{SUB}(\mathcal{T}_{S_\rho})$ and $\mathfrak{F}_\rho(f\tau)$ is defined then $f\tau \in \text{SUB}(\mathcal{T}_{E_\rho})$. \square

On the basis of lemma 9 it is now easy to prove:

10. LEMMA. $\forall \tau \not\approx \tau' \in S_\rho (\tau \notin \text{SUB}(\mathcal{T}_{E_\rho})) \rightarrow \models_{M_\rho} S_\rho$.

PROOF. (If the condition is satisfied $\models_{M_\rho} \phi$ holds for every $\phi \in S_\rho$.) If $\phi = \tau \approx \tau' \in S_\rho$ with $m(\tau') \leq m(\tau)$, then $\tau' \in \mathcal{T}_{E_\rho}^\varepsilon$ by 7(ii) and hence $h^c(\tau') = \tau'$. We get then $h^c(\tau) = \tau'$ for $m(\tau') = m(\tau)$ by $\tau' = \tau$ and for $m(\tau') < m(\tau)$ by the definition of h^c , since $\tau \notin \mathcal{T}_{E_\rho}^\varepsilon$. Thus $\mathfrak{F}_\rho(\tau) = \mathfrak{F}_\rho(\tau')$ by 9(i) and hence $\models_{M_\rho} \phi$.

Assume $\phi = \tau \not\approx \tau'$. If $\tau \approx \tau'$ were satisfied by M_ρ , we would get $\mathfrak{F}_\rho(\tau) = \mathfrak{F}_\rho(\tau')$ and by 9(ii) $\tau, \tau' \in \text{SUB}(\mathcal{T}_{E_\rho})$. Since 7(ii) ensures $h^c(\tau) = h^c(\tau') = \tau = \tau'$, we would have $\tau \approx \tau' \in S_\rho$ with $\tau \in \text{SUB}(\mathcal{T}_{E_\rho})$. \square

Finally it should be mentioned that M_ρ is a unique (up to isomorphism) minimal model for S_ρ , i.e. if M is a model for S_ρ , homomorphic to M_ρ , then every minimal submodel of M that satisfies S_ρ is isomorphic to M_ρ .

3 The Generation Problem and its Decidability

Although it was not necessary for the definition of the sentences derivable by a unification grammar, we now have to make explicit that also a feature description is assigned to a sentence.

11. DEFINITION. A terminal string w ($w \in V_T^*$) is *derivable with* feature description $\exists x_1..x_l \phi(x_1, \dots, x_l)$ iff the feature description is satisfiable and there is a derivation $\langle S_\emptyset, \emptyset \rangle = \pi_0.. \pi_n$ with $\omega_n = s[w]_\emptyset$ and $\phi = \bigwedge S^\pi$.

Since deductively equivalent consistent feature descriptions are assumed to describe the same set of feature structures (models), the assignment of entities to terminal strings determined by a unification grammar is then formally given by a binary relation Δ between terminal strings and sets of classes of deductively equivalent feature descriptions $[\exists x_1..x_l \phi(x_1, \dots, x_l)]_{\dashv}$.⁷

12. DEFINITION. For each terminal string $w \in V_T^*$ and each class $[\exists x_1..x_l \phi(x_1, \dots, x_l)]$:

$$\Delta(w, [\exists x_1..x_l \phi(x_1, \dots, x_l)]) \text{ iff } w \text{ is derivable with } \exists x_1..x_l \phi(x_1, \dots, x_l).$$

Definition 12 now brings us closer to the problem, since we can for any unification grammar in rather abstract terms specify what parsers and generators are: a parser is a procedure which recursively enumerates for any given string w the set

$$\{[\exists x_1..x_l \phi(x_1, \dots, x_l)] \mid \Delta(w, [\exists x_1..x_l \phi(x_1, \dots, x_l)])\}$$

⁷We omit the index of the equivalence classes in the following.

and a generator is a procedure which recursively enumerates for any given class $[\exists x_1..x_l \phi(x_1, \dots, x_l)]$:⁸

$$\{w \in V_T^* \mid \Delta(w, [\exists x_1..x_l \phi(x_1, \dots, x_l)])\}.$$

Whether adequate algorithms (effective procedures) can be formulated depends on the decidability of the corresponding *parsing* and *generation problem*. In our case (generation), it is the problem whether

$$\exists w \in V_T^* (\Delta(w, [\exists x_1..x_l \phi(x_1, \dots, x_l)]))$$

is decidable for any given class $[\exists x_1..x_l \phi(x_1, \dots, x_l)]$. The decidability of the generation problem alone ensures the existence of algorithms which terminate in any case with an output, although they might (of course) not be able to produce all possible solutions. Despite decidability, inputs can still be infinitely ambiguous ($\{w \in V_T^* \mid \Delta(w, [\exists x_1..x_l \phi(x_1, \dots, x_l)])\}$ infinite).

In order to prove the decidability of the generation problem (theorem 13), we proceed in two steps.

13. THEOREM. *It is decidable for each feature description $\exists y_1..y_k \psi(y_1, \dots, y_k)$ whether there is a terminal string $w \in V_T^*$ which is derivable with $\exists x_1..x_l \phi(x_1, \dots, x_l)$ and*

$$\exists y_1..y_k \psi(y_1, \dots, y_k) \dashv \vdash \exists x_1..x_l \phi(x_1, \dots, x_l).$$

In the first step we show that we can always shorten a derivation of a sentence w with (consistent) feature description ϕ to a derivation of a sentence w' with feature description ϕ' and $\phi \dashv \vdash \phi'$ whose length is bounded by the "size" of ϕ . By showing in the second step that two deductively equivalent consistent feature descriptions have the same "size" theorem 13 follows, since only a *finite* set of derivations (those whose length does not exceed this upper bound) have to be inspected in order to decide $\exists w \in V_T^* (\Delta(w, [\phi]))$ for an arbitrary consistent input ϕ .

3.1 Redundant Recursions and Pumping

For the proof that for a derivation of a sentence w with (consistent) feature description ϕ there always exists a short derivation of a sentence w' with feature description ϕ' and $\phi \dashv \vdash \phi'$ we exploit the fact that a c-structure may contain recursions of the form depicted in figure 2 whose corresponding subderivations in π are eliminable. Such recursions are called *redundant*.

14. DEFINITION. Let π be a derivation of a sentence $uvzxy$ of length $m + k + l$ whose c-structure derivation has the form $\omega_0.. \omega_m.. \omega_{m+k}.. \omega_{m+k+l} = \omega_n$ with $\omega_m = s[u, A_\mu, y]_\emptyset$ and $\omega_{m+k} = s[u, A[v, A_{\mu, \kappa}, x]_\mu, y]_\emptyset$ (reorder if necessary). If π' is a derivation of uzy of length $m + l$ which is defined for each j ($0 \leq j \leq m + l$) by

$$\pi'_j = \begin{cases} \pi_j & \text{if } j \leq m \\ \{s[ury]_\emptyset, S\} & \text{if } j > m \text{ and } \pi_{j+k} = \{s[uvrxy]_\emptyset, S\} \end{cases}$$

⁸We assume here strong reversibility, since a generator is for a given input y simply a parser which operates on Δ^{-1} : it recursively enumerates instead of $\{x \mid \Delta(x, y)\}$ the set $\{x \mid \Delta^{-1}(x, y)\}$.

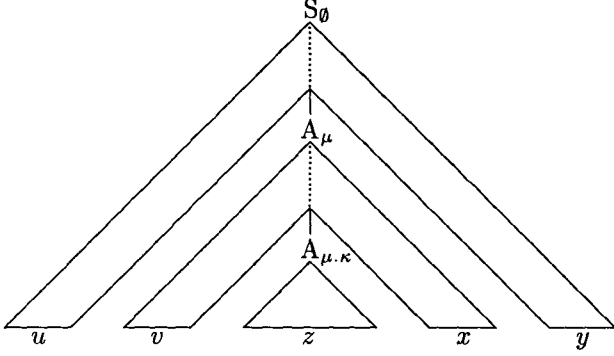


Figure 2

A c-structure with recursion $A[v, A_{\mu, \kappa}, x]_{\mu}$ ($|\kappa| > 0$).

and

$$\varpi'_j = \begin{cases} \varpi_j & \text{if } j \leq m \\ \mu.t & \text{if } j > m \text{ and } \varpi_{j+k} = \mu.\kappa.t \end{cases}$$

then $\pi_{m+1} \dots \pi_{m+k}$ is a *redundant recursion* iff $\exists x_1 \dots x_l \wedge S^\pi(x_1, \dots, x_l) \dashv\vdash \exists x'_1 \dots x'_k \wedge S^{\pi'}(x'_1, \dots, x'_k)$.

If we assume that a given derivation of a sentence is already shortened to a derivation without redundant recursions it remains to show that the length of such a derivation could not exceed the upper bound determined by the “size” of the derived feature description.

The “size” of a consistent feature description is on the one hand determined by the size of its minimal model, and on the other hand determined by a normal form into which every feature description can be converted. The conversion is performed in two steps. In the first step, we eliminate as many variables as possible by substitution.

15. DEFINITION. If $S(x_1, \dots, x_l)$ is a set of literals, then x_i is *eliminable* in $S(x_1, \dots, x_l)$ iff there is a term τ not containing x_i such that $\vdash \exists x_1 \dots x_l (\wedge S(x_1, \dots, x_l) \supset x_i \approx \tau)$.

16. NOTATION. In the following we write $S[x_1, \dots, x_l]$ iff each x_i is not eliminable in S .

17. DEFINITION. We assign to a set of literals $S'(x_1, \dots, x_l, x'_1, \dots, x'_k)$ a set $R(S')$ which contains a set $S[x_1, \dots, x_l]$ iff there is a substitution $\theta \in \{\{x'_1, \dots, x'_k\} \mapsto \mathcal{T}(x_1, \dots, x_l)\}$ such that

$$\vdash \exists x_1 \dots x_l x'_1 \dots x'_k (\wedge S' \supset x'_i \approx \theta(x'_i))$$

for all x'_i ($1 \leq i \leq k$) and $S[x_1, \dots, x_l] = S'[\theta]$.⁹

By the substitutivity theorem we get:

18. LEMMA. If $S[x_1, \dots, x_l] \in R(S'(x_1, \dots, x_l, x'_1, \dots, x'_k))$ then $\exists x_1 \dots x'_k (\wedge S \equiv \wedge S')$.

In the second step, we make the set of literals independent, i.e. we remove those literals which are implied by the remaining subset.

⁹ $\mathcal{T}(x_1, \dots, x_l)$ denotes the set of terms over V , $\{x_1, \dots, x_l\}$ and F_1 .

19. DEFINITION. A set of literals $S(x_1, \dots, x_l)$ is *independent* iff there is no formula $\phi \in S$ for which $\vdash \exists x_1 \dots x_l (\wedge (S \setminus \{\phi\}) \supset \phi)$ holds.

Normal forms are then defined as follows.

20. DEFINITION. A consistent feature description $\exists x_1 \dots x_l \wedge S[x_1, \dots, x_l]$ is in *normal form* (in the following indicated by a ν index) iff $S[x_1, \dots, x_l]$ is independent.

Furthermore, we call $\exists x_1 \dots x_l \wedge S_\nu[x_1, \dots, x_l]$ a *normal form* of $\exists x_1 \dots x_l x'_1 \dots x'_k \wedge S'(x_1, \dots, x'_k)$ iff $S_\nu[x_1, \dots, x_l]$ is an independent subset of $S'[\theta] \in R(S')$ and $\exists x_1 \dots x_l (\wedge S_\nu[x_1, \dots, x_l] \equiv \wedge S'[\theta])$.

Lemma 18 and the condition in definition 20 ensure that a consistent feature description and its normal forms are deductively equivalent.

In order to be able to show the existence of a redundant recursion, we exploit the simple fact that the information which contributes a literal in a normal form with a minimal model $\langle \mathcal{U}_\rho, \mathfrak{F}_\rho \rangle$ can be specified by an equation $\tau \approx \tau'$ where $|\tau| + |\tau'| \leq |\mathcal{U}_\rho| + 2$. A literal $\tau \approx \sigma\sigma'\tau''$ ($|\sigma| > 0$) whose terms are longer must always be reducible by a loop $\sigma'\tau'' \approx \tau''$ to a shorter equation. Since the construction of such an information piece can be done with a subderivation of some fixed length, there must be a redundant recursion if the length of whole derivation exceeds a fixed value which is dependent on $|\mathcal{U}_\rho|$ and $|\mathcal{S}_\nu|$ and exactly specified in lemma 21.¹⁰

21. LEMMA. Suppose that $w \in V_T^*$ is derivable with $\phi = \exists x_1 \dots x_l x'_1 \dots x'_k \wedge S^\pi(x_1, \dots, x'_k)$ over π of length n , that $\exists x_1 \dots x_l \wedge S_\nu[x_1, \dots, x_l]$ is a normal form of ϕ and that $M_\rho = \langle \mathcal{U}_\rho, \mathfrak{F}_\rho \rangle$ is a minimal model of ϕ . If π has no redundant recursions then each path of ω_n is shorter or equal to $|V_N| \cdot (3|\mathcal{U}_\rho| + 1) \cdot (|\mathcal{S}_\nu| + 1)$.

PROOF. Suppose, one path of ω_n were longer than $|V_N| \cdot (3|\mathcal{U}_\rho| + 1) \cdot (|\mathcal{S}_\nu| + 1)$, then more than $(3|\mathcal{U}_\rho| + 1) \cdot (|\mathcal{S}_\nu| + 1)$ different nodes on that path had to be labelled by the same $A \in V_N$. Without loss of generality we can assume that π is a derivation whose c-structure derivation ω has the form $\omega_0 \dots \omega_m \dots \omega_n$ with $\omega_m = s[u, A_\mu, y]_\emptyset$ for each node μ on that path which is labelled by A (reorder if necessary). In order to exclude that complex inferences are used to build up θ , we assume furthermore that θ is non-deterministically constructed from S^π by recursive variable substitution, i.e. we require for each $\langle x, \sigma\tau \rangle \in \theta$ either $x \approx \sigma\tau \in S^\pi$ or $\exists x \approx \sigma y \in S^\pi (\langle y, \tau \rangle \in \theta)$. Finally, let $S \subseteq S^\pi$ with $S_\nu = S[\theta]$. In order to identify the redundant recursion we have to consider the following cases.

1. Suppose there are more than $|\mathcal{S}_\nu| + 1$ A -labelled nodes μ such that x_μ does not occur in S^π , then there must be more than $|\mathcal{S}_\nu|$ non-overlapping recursions. For at least one of those recursions $\pi_{m+1} \dots \pi_{m+k}$ it must hold that

$$(S \cap S_{m+1 \rightarrow m+k}^\pi) \subseteq (S_{0 \rightarrow m}^\pi \cup S_{m+k+1 \rightarrow n}^\pi).$$

¹⁰The given factor $|V_N| \cdot (3|\mathcal{U}_\rho| + 1) \cdot (|\mathcal{S}_\nu| + 1)$ decreases for more restricted grammars, like e.g. grammars which allow only feature descriptions with single-rooted and/or acyclic minimal models.

But then

$$S \subseteq (S_{0 \rightarrow m}^\pi \cup S_{m+k+1 \rightarrow n}^\pi)$$

and $\pi_{m+1} \dots \pi_{m+k}$ must be redundant.

2. If case 1 does not apply there must be more than $3|\mathcal{U}_\rho| \cdot (|S_\nu| + 1)$ distinct A-labelled nodes μ on that path such that x_μ occurs in S^π and for more than $3|S_\nu| + 1$ of these nodes must pairwise hold

$$\vdash \exists x_1 \dots x'_k (\wedge S^\pi \supset x_\mu \approx x_{\mu.\kappa}).$$

But then there must be at least three recursions such that

$$\vdash \exists x_1 \dots x'_k (\wedge S^\pi \supset x_\iota \approx x_{\iota.\lambda} \approx x_{\iota.\lambda.v} \approx x_{\iota.\lambda.v.\epsilon})$$

and

$$(S \cap S_{i+1 \rightarrow i+l+v+z}^\pi) \subseteq (S_{0 \rightarrow i}^\pi \cup S_{i+l+v+z+1 \rightarrow n}^\pi).$$

We can then assign to each recursion $\pi_{m+1} \dots \pi_{m+k}$ ($m=i, k=l; m=i+l, k=v$ or $m=i+l+v, k=z$) a type which corresponds to the strongest of the following conditions the recursion satisfies.

(a) $\pi_{m+1} \dots \pi_{m+k}$ satisfies

$$\vdash \exists x_1 \dots x'_k (\wedge S_{m+1 \rightarrow m+k}^\pi \supset x_\mu \approx x_{\mu.\kappa}).$$

(b) It holds only

$$\vdash \exists x_1 \dots x'_k (\wedge S_{m+1 \rightarrow m+k}^\pi \supset x_{\mu.\kappa} \approx \sigma x_\mu)$$

with $|\sigma| > 0$ and $\sigma\theta(x_\mu) \approx \theta(x_{\mu.\kappa})$ is implied by $\vdash \exists x_1 \dots x'_k (\wedge S_{0 \rightarrow m+k}^\pi \wedge S_g)$ where S_g is the set of ground literals of S .

(c) Or it holds

$$\vdash \exists x_1 \dots x'_k (\wedge S_{m+1 \rightarrow m+k}^\pi \supset \sigma' x_{\mu.\kappa} \approx x_\mu)$$

with $|\sigma'| > 0$ and $\sigma'\theta(x_{\mu.\kappa}) \approx \theta(x_{\mu.\kappa})$ is implied by $\vdash \exists x_1 \dots x'_k (\wedge S_{m+1 \rightarrow n}^\pi \wedge S_g)$.

(d) If a recursion which satisfies

$$\vdash \exists x_1 \dots x'_k (\wedge S_{m+1 \rightarrow m+k}^\pi \supset x_{\mu.\kappa} \approx \sigma x_\mu)$$

($|\sigma| > 0$) is not of type (b) then there must be a ground term τ which is not reducible in terms of x_μ , i.e. θ can not satisfy $\tau = \sigma'\theta(x_\mu)$ for some non-empty prefix σ' , and

$$\vdash \exists x_1 \dots x'_k (\wedge S^\pi \supset x_{\mu.\kappa} \approx \tau \wedge \tau \approx \theta(x_\mu)).$$

(e) For a recursion with

$$\vdash \exists x_1 \dots x'_k (\wedge S_{m+1 \rightarrow m+k}^\pi \supset \sigma' x_{\mu.\kappa} \approx x_\mu)$$

which is not of type (c) we get for x_μ the same property as for $x_{\mu.\kappa}$ in (d).

(f) If the previous cases do not apply, the recursion might satisfy

$$\vdash \exists x_1 \dots x'_k (\wedge S_{m+1 \rightarrow m+k}^\pi \supset \sigma' x_{\mu.\kappa} \approx \sigma x_\mu)$$

with $|\sigma| > 0$ and $|\sigma'| > 0$. Since x_μ is not eliminable in terms of $x_{\mu.\kappa}$ and vice versa, there must be ground terms τ, τ' such that

$$\vdash \exists x_1 \dots x'_k (\wedge S^\pi \supset x_\mu \approx \tau \wedge x_{\mu.\kappa} \approx \tau' \wedge \tau \approx \tau').$$

(g) If a recursion is not of type (a-f) then

$$\vdash \exists x_1 \dots x'_k (\wedge S_{m+1 \rightarrow m+k}^\pi \supset \sigma x \approx x_\mu \wedge \sigma' y \approx x_{\mu.\kappa}).$$

But then x_μ and $x_{\mu.\kappa}$ must be ground eliminable as in (f).

Since a recursion of type (a-c) is not redundant if it contains terms τ or τ' such that τ is not reducible in terms of x_μ and τ' is not reducible in terms of $x_{\mu.\kappa}$ and τ or τ' are used to eliminate x_μ and $x_{\mu.\kappa}$, there must be at least one recursion $\pi_{m+1} \dots \pi_{m+k}$ such that

$$\vdash \exists x_1 \dots x'_k (\wedge S^\pi \equiv (S_{0 \rightarrow m}^\pi \cup S_{m+k+1 \rightarrow n}^\pi \cup \{x_\mu \approx x_{\mu.\kappa}\}))$$

and θ still follows either by ground inferences or due to the properties of (b) and (c). \square

If $l_{max} = \max\{|w| \mid \langle \langle A, w \rangle, S_r \rangle \in R\}$ then the following *pumping lemma* follows immediately as a corollary.

22. COROLLARY. *Suppose that $w \in V_T^*$ is derivable with $\phi = \exists x_1 \dots x_l x'_1 \dots x'_k \wedge S^\pi(x_1, \dots, x'_k)$ over π of length n , that $\exists x_1 \dots x_l \wedge S_\nu[x_1, \dots, x_l]$ is a normal form of ϕ and that $M_\rho = (\mathcal{U}_\rho, \mathfrak{S}_\rho)$ is a minimal model of ϕ . If $|w| > l_{max}^{|V_N| \cdot (2|\mathcal{U}_\rho| + 1)}$ then w has the form $uvzxy$ with $|vx| > 0$ and for all $i \geq 1$: $\langle uv^i zx^i y, [\exists x_1 \dots x_l x'_1 \dots x'_k \wedge S^\pi(x_1, \dots, x'_k)] \rangle \in \Delta$.*

PROOF. If $|w| > l_{max}^{|V_N| \cdot (2|\mathcal{U}_\rho| + 1)}$ then at least one path of ω_n is longer than $|V_N| \cdot (2|\mathcal{U}_\rho| + 1)$ and more than $2|\mathcal{U}_\rho| + 1$ different nodes on that path are labelled by the same $A \in V_N$. Without loss of generality we assume again that π is a derivation whose c-structure derivation ω has the form $\omega_0 \dots \omega_m \dots \omega_n$ with $\omega_m = s[u, A_\mu, y]_\emptyset$ for each node μ on that path which is labelled by A, and that θ is non-deterministically constructed from S^π by recursive variable substitution, i.e. we require for each $\langle x, \sigma\tau \rangle \in \theta$ either $x \approx \sigma\tau \in S^\pi$ or $\exists x \approx \sigma y \in S^\pi (\langle y, \tau \rangle \in \theta)$. Suppose furthermore that $S \subseteq S^\pi$ with $S_\nu = S[\theta]$. In order to isolate the recursion which allows pumping we have to distinguish the following cases.

1. If π contains a recursion $\pi_{m+1} \dots \pi_{m+k}$ with $\omega_m = s[u, A_\mu, y]_\emptyset$, $\omega_{m+k} = s[u, A[v, A_{\mu.\kappa}, x]_\mu, y]_\emptyset$ and $|vx| > 0$ and x_μ and $x_{\mu.\kappa}$ do not occur in S^π , we take $\pi_{m+1} \dots \pi_{m+k}$.

2. If π does not contain such a recursion there must be at least three distinct A-labelled nodes $\iota, \iota.\lambda, \iota.\lambda.v$ on that path such that

$$\begin{aligned} \omega_\iota &= s[u', A_\iota, y']_\emptyset, \\ \omega_{\iota+\lambda} &= s[u', A[v', A_{\iota.\lambda}, x']_{\iota}, y']_\emptyset, \\ \omega_{\iota+\lambda+v} &= s[u', A[v', A[r, A_{\iota.\lambda.v}, v]_{\iota.\lambda}, x']_{\iota}, y']_\emptyset \end{aligned}$$

with $|v'x'| > 0$, $|rs| > 0$ and

$$\vdash \exists x_1 \dots x'_k (\wedge S^\pi \supset x_\iota \approx x_{\iota.\lambda} \approx x_{\iota.\lambda.v}).$$

2.1 Suppose there is a recursion of type (a-c) (cf. proof of lemma 21) we choose this one.

2.2 If π does not contain such a recursion each of the recursions must be of type (d), (e), (f) or (g). But then there must be one recursion $\pi_{m+1} \dots \pi_{m+k}$ ($m=i, k=l$ or $m=i+l, k=v$) with

$$\vdash \exists x_1 \dots x'_k (\wedge S_{m+1 \rightarrow m+k}^\pi \supset x_\mu \approx \tau \vee x_{\mu.\kappa} \approx \tau')$$

for some ground terms τ, τ' . This recursion is chosen for the proof.

On the basis of the recursion $\pi_{m+1} \dots \pi_{m+k}$ we can now define derivations π^i as follows. We set $\pi^1 = \pi$ and define π^{i+1} on the basis of π^i by

$$\pi_j^{i+1} = \begin{cases} \langle S[u, A[v^i r x^i]_{\mu, \kappa^i}, y]_{\emptyset}, S \rangle & \text{if } j > m + ik \text{ and} \\ \pi_{j-k}^i = \langle S[u, A[v^{i-1} r x^{i-1}]_{\mu, \kappa^{i-1}}, y]_{\emptyset}, S \rangle & \\ \pi_j^i & \text{if } j \leq m + ik \end{cases}$$

and

$$\varpi_j^{i+1} = \begin{cases} \varpi_j^i & \text{if } j \leq m + ik \\ \mu, \kappa^i, l & \text{if } j > m + ik \text{ and } \varpi_{j-k}^i = \mu, \kappa^{i-1}, l. \end{cases}$$

By induction on i it can then be shown for all possible cases that $\vdash \exists x_1 \dots x_l. (\bigwedge S^{\pi^i} \equiv \bigwedge S^{\pi})$. \square

3.2 Invariance of the Parameters under Deductive Equivalence

Since the universes of the minimal models of two deductively equivalent consistent feature descriptions must have the same cardinality, for the completion of the proof of theorem 13 it remains to be shown that two deductively equivalent consistent feature descriptions have the same ‘‘information content’’, i.e. that the sets of literals of their normal forms have the same cardinality:

23. LEMMA. *Suppose that $\exists x_1 \dots x_l \bigwedge S_\nu[x_1, \dots, x_l]$ and $\exists y_1 \dots y_k \bigwedge S'_\nu[y_1, \dots, y_k]$ are deductively equivalent consistent feature descriptions in normal form then $|S_\nu[x_1, \dots, x_l]| = |S'_\nu[y_1, \dots, y_k]|$.*

We proof lemma 23 in two steps. First, we show that we can convert $S'_\nu[y_1, \dots, y_k]$ into a set $S''_\nu[x_1, \dots, x_l]$ with the same cardinality such that $\exists x_1 \dots x_l (\bigwedge S_\nu[x_1, \dots, x_l] \equiv \bigwedge S''_\nu[x_1, \dots, x_l])$ holds.

24. LEMMA. *Assume that $\exists x_1 \dots x_l \bigwedge S_\nu[x_1, \dots, x_l]$ and $\exists y_1 \dots y_k \bigwedge S'_\nu[y_1, \dots, y_k]$ are deductively equivalent consistent feature descriptions in normal form. Then $l = k$ and there is a set $S''_\nu[x_1, \dots, x_l]$ with*

- (i) $|S''_\nu[x_1, \dots, x_l]| = |S'_\nu[y_1, \dots, y_k]|$ and
- (ii) $\vdash \exists x_1 \dots x_l (\bigwedge S_\nu[x_1, \dots, x_l] \equiv \bigwedge S''_\nu[x_1, \dots, x_l])$.

PROOF. Suppose that $\{x_1, \dots, x_l\} \cap \{y_1, \dots, y_k\} = \emptyset$ (rename if necessary), that $S'_\nu[y_1, \dots, y_k]$ is in reduced form (the reduction of an independent set does not change the cardinality) and that $M = \langle \mathcal{U}, \mathfrak{F} \rangle$ is an arbitrary model of $\exists x_1 \dots x_l \bigwedge S_\nu[x_1, \dots, x_l]$ and $\exists y_1 \dots y_k \bigwedge S'_\nu[y_1, \dots, y_k]$. Let α and α' be assignments such that $\alpha \models_M S_\nu[x_1, \dots, x_l]$ and $\alpha' \models_M S'_\nu[y_1, \dots, y_k]$.

We show first that there is a bijective function $\vartheta \in \{\{x_1, \dots, x_l\} \mapsto \{y_1, \dots, y_k\}\}$ such that for all x_i there is a $\sigma_i \in F_1^*$ ($i = 1, \dots, l$) and a variable $\vartheta(x_i)$ occurring in S' with $\alpha(x_i) = \mathfrak{F}(\sigma_i \vartheta(x_i))(\alpha')$. First of all ϑ is left-total, since $\alpha(x_i) = \mathfrak{F}(\tau)$ with τ variable-free would imply that x_i is eliminable in S . In order to show that ϑ is a function, assume $\alpha(x_i) = \mathfrak{F}(\sigma_j y_j)(\alpha') = \mathfrak{F}(\sigma_h y_h)(\alpha')$ for y_j, y_h occurring in S' with $y_j \neq y_h$. Since y_j and y_h are not eliminable in S' there must

be terms τ_j, τ_h such that $\alpha'(y_j) = \mathfrak{F}(\tau_j)(\alpha)$, $\alpha'(y_h) = \mathfrak{F}(\tau_h)(\alpha)$, $\mathfrak{F}(\sigma_j \tau_j)(\alpha) = \mathfrak{F}(\sigma_h \tau_h)(\alpha)$ and there is no $\sigma \in F_1^*$ such that $\mathfrak{F}(\sigma \tau_j)(\alpha) = \mathfrak{F}(\tau_h)(\alpha)$ or $\mathfrak{F}(\sigma \tau_h)(\alpha) = \mathfrak{F}(\tau_j)(\alpha)$. Thus, τ_j and τ_h must be terms in $\mathcal{T}(\{x_1, \dots, x_l\} \setminus \{x_i\})$ and x_i would be eliminable in S . Suppose now that y_j is not in the range of ϑ . Then there must be a term τ with $\mathfrak{F}(\tau)(\alpha) = \alpha'(y_j)$. Since y_j is not eliminable in S' , τ must be of the form σx_i and there must be a term τ' with $\alpha(x_i) = \mathfrak{F}(\tau')(\alpha')$. If τ' is a term in $\mathcal{T}(\{y_1, \dots, y_k\} \setminus \{y_j\})$, y_j would be eliminable. Otherwise τ' is of the form $\sigma' y_j$ and we would get $\alpha(x_i) = \mathfrak{F}(\sigma' y_j)(\alpha')$. Hence ϑ is onto. Assume finally $\alpha(x_i) = \mathfrak{F}(\sigma_i y_h)(\alpha')$ and $\alpha(x_j) = \mathfrak{F}(\sigma_j y_h)(\alpha')$ with $x_i \neq x_j$. Then there must be a term τ with $\alpha'(y_h) = \mathfrak{F}(\tau)(\alpha)$. Since x_i and x_j would be eliminable in S if τ is a term in $\mathcal{T}(\{x_1, \dots, x_l\} \setminus \{x_i, x_j\})$, τ is of the form σx_i or σx_j . But then either $\alpha(x_j) = \mathfrak{F}(\sigma_j \sigma x_i)(\alpha)$ or $\alpha(x_i) = \mathfrak{F}(\sigma_i \sigma x_j)(\alpha)$. Thus, ϑ is bijective and $l = k$.

On the basis of ϑ we then define a sequence of new sets S'_i ($0 \leq i \leq l$) by induction as follows (within the induction we assume $\vartheta(x_i) = y$):

$$S'_0 = S'_\nu[y_1, \dots, y_l]_\rho$$

$$S'_i = \begin{cases} S'_{i-1}[y/x_i] & \text{if } \alpha(x_i) = \alpha'(y) \\ (S'_{i-1} \setminus \{y \approx \sigma \sigma' y\})[y/\sigma x_i] \cup \{x_i \approx \sigma' \sigma x_i\} & \text{if (A),} \end{cases}$$

where (A) means $\alpha(x_i) \neq \alpha'(y)$, $y \approx \sigma \sigma' y \in S'_{i-1}$ and $\alpha(x_i) = \mathfrak{F}(\sigma' y)(\alpha')$. In the case where the variables refer to different nodes on a loop ($\alpha(x_i) \neq \alpha'(y)$) the definition is well-formed, since $S'_\nu[y_1, \dots, y_l]_\rho$ is reduced and normalized and thus there must be exactly one equation $y \approx \sigma \sigma' y$ in S'_{i-1} describing the loop with the node to which x_i refers. For $S'' = S'_l$, $|S''_\nu[x_1, \dots, x_l]| = |S'_\nu[y_1, \dots, y_k]|$ follows immediately by induction on the construction of S'' .

Finally we get (ii), since

$$\vdash \exists x_1 \dots x_l (\exists x_{i+1} \dots x_l \bigwedge S_\nu[x_1, \dots, x_l] \equiv \exists \vartheta(x_{i+1}) \dots \vartheta(x_l) \bigwedge S'_i[x_1, \dots, x_i, \vartheta(x_{i+1}), \dots, \vartheta(x_l)])$$

can easily be verified by induction on the construction of S'' . \square

Since two deductively equivalent independent and consistent sets of (variable-free) literals reduce to the same set of literals in reduced form, lemma 25 follows by skolemization and completes the proof of lemma 23.

25. LEMMA. *If $\exists x_1 \dots x_l \bigwedge S_\nu[x_1, \dots, x_l]$ is a consistent feature description in normal form and $\vdash \exists x_1 \dots x_l (\bigwedge S_\nu[x_1, \dots, x_l] \equiv \bigwedge S''_\nu[x_1, \dots, x_l])$ then $|S_\nu[x_1, \dots, x_l]| = |S''_\nu[x_1, \dots, x_l]|$.*

Appendix: Syntax and Semantics of Feature-Description Languages

A feature-description language $L(C)$ consists of the logical connectives \sim (negation), \supset (implication), the equality symbol \approx , the existential quantifier \exists and the parentheses $(,)$. The *nonlogical* vocabulary is given by a finite set of constants V (*atomic values*),

a possibly empty finite set of constants C (*Skolem-constants*) and a finite set of *unary partial function symbols* F_1 (V, C, F_1 pairwise disjoint). The class of terms and formulas of $L(C)$ are recursively defined as usual. Feature descriptions of $L(C)$ are expressions of the form $\exists x_1 \dots x_l \wedge S[x_1, \dots, x_l]$, where S is a finite set of (quantifier-free) literals. (We assume that the connectives \vee (disjunction), \wedge (conjunction) and \equiv (equivalence) are introduced by their usual definitions.)

A *model* for $L(C)$ consists of a nonempty *universe* \mathcal{U} and an *interpretation* function \mathfrak{S} . Since not every term denotes an element in \mathcal{U} if the function symbols are interpreted as unary partial functions, we generalize the partiality of the denotation by assuming that \mathfrak{S} itself is a partial function. It is only required that all Skolem-constants denote. Suppose $[X \mapsto Y]$ designates the set of all partial functions from X to Y and $[X \mapsto Y]$ the set of all total functions from X to Y , then a model is defined as follows:¹¹

DEFINITION. A *model* for $L(C)$ is a pair $M = \langle \mathcal{U}, \mathfrak{S} \rangle$, consisting of a nonempty set \mathcal{U} and an interpretation function $\mathfrak{S} = \mathfrak{S}_V \cup \mathfrak{S}_C \cup \mathfrak{S}_{F_1}$, such that

- (i) $\mathfrak{S}_V \in [V \mapsto \mathcal{U}]$,
- (ii) $\mathfrak{S}_C \in [C \mapsto \mathcal{U}]$,
- (iii) $\mathfrak{S}_{F_1} \in [F_1 \mapsto [\mathcal{U} \mapsto \mathcal{U}]]$,
- (iv) $\forall f \in F_1 (f \in \text{Dom}(\mathfrak{S}) \rightarrow \mathfrak{S}(f) \neq \emptyset)$.

If we extend the denotation function to terms and variable assignments α , the definition of the satisfaction relation differs only in the clause for the equations from the usual one:

$\alpha \models_M \tau \approx \tau'$ iff $\mathfrak{S}(\tau)(\alpha)$ and $\mathfrak{S}(\tau')(\alpha)$ are defined and $\mathfrak{S}(\tau)(\alpha) = \mathfrak{S}(\tau')(\alpha)$.

References

- Beierle, C., and U. Pletat. 1988. Feature Graphs and Abstract Data Types: A Unifying Approach. In *Proceedings of the 12th International Conference on Computational Linguistics*. Budapest.
- Kaplan, R., and J. Bresnan. 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In J. Bresnan, ed., *The Mental Representation of Grammatical Relations*. Cambridge, Mass.: The MIT Press.
- Johnson, M. 1988. *Attribute-Value Logic and the Theory of Grammar*. Chicago: CSLI Lecture Notes Series, Chicago University Press.
- Knuth, D., and P. Bendix. 1970. Simple Word Problems in Universal Algebra. In J. Leech, ed., *Computational Problems in Universal Algebra*. Elmsford: Pergamon Press.
- Shieber, S., H. Uszkoreit, F. Pereira, J. Robinson, and M. Tyson. 1983. The Formalism and Implementation of PATR-II. In B. Grosz and M. Stickel, eds., *Research on Interactive Acquisition and Use of Knowledge*. SRI Final Report 1984. SRI International, Menlo Park.

¹¹If constant-consistency and constant/complex-consistency are to be guaranteed for the atomic values V or acyclicity has to be ensured for the models, special conditions can be added to the definition.

Statman, R. 1977. Herbrand's Theorem and Gentzen's Notion of a Direct Proof. In J. Barwise, ed., *Handbook of Mathematical Logic*. Amsterdam: Elsevier North-Holland.

Wedekind, J. 1991. Classical Logics for Attribute-Value Languages. In *Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics*. Berlin.

Wedekind, J. 1994. Some Remarks on the Logic of Unification Grammars. In C. J. Rupp, M. Rosner, and R. Johnson, eds., *Constraints, Language and Computation*. London: Academic Press.