

Semantic-driven Generation with LFG- and PATR-style Grammars

Jürgen Wedekind*
University of Stuttgart

To find an appropriate utterance for a semantic representation is a problem normally treated in the domain of (tactical) natural language generation. For unification-based approaches, like LFG, PATR, or HPSG (Kaplan and Bresnan 1982; Shieber et al. 1983; Pollard and Sag 1994), this problem turns out to be a formal problem of the underlying grammar formalism, when the mapping between strings and semantic representations is defined by the grammar. Semantic representations are then encoded in a separate part of the feature structures (henceforth f-structures) that are assigned to the sentences by the grammar. This is normally achieved by a distinct attribute SEM (or an additional σ -projection that is formally reconstructable by such an attribute) whose value is intended to represent the semantics of the sentence the f-structure is assigned to. The f-structure given in (1), which might be assigned to the sentence *John arrives* by a unification grammar for English, is a simple example.

$$(1) \left[\begin{array}{l} \text{PRED} \quad \text{'ARRIVE}\langle(\text{SUBJ})\rangle' \\ \text{TENSE} \quad \text{PRES} \\ \text{SUBJ} \quad \left[\text{PRED} \quad \text{'JOHN'} \right] \\ \text{SEM} \quad \left[\begin{array}{l} \text{REL} \quad \text{arrive} \\ \text{ARG1} \quad \text{john} \end{array} \right] \end{array} \right]$$

Since the f-structures assigned to the sentences are always subsumed by the semantic representations they contain, a semantic-driven generator has to compute for a given semantic representation Φ' a sentence with an f-structure Φ that is subsumed by the input (in the following, notated by $\Phi' \sqsubseteq \Phi$). To state the underlying decidability problem more formally, we need the fact that a unification grammar G defines a binary relation Δ_G between terminal strings w and f-structures Φ , as given in (2)

$$(2) \quad \Delta_G(w, \Phi) \text{ iff } G \text{ assigns } \Phi \text{ to } w.$$

The problem of determining for a given semantic representation Φ' whether there is a sentence with an f-structure Φ that is subsumed by the input turns out then to be an instance of the problem of whether we can decide (3)

$$(3) \quad \exists w \exists \Phi (\Phi' \sqsubseteq \Phi \wedge \Delta_G(w, \Phi))$$

for any given input Φ' .

The undecidability of the *generation problem* in (3) was shown for definite clause grammars by Dymetman (1991), who reduced the problem to Hilbert's Tenth Prob-

* Institute for Natural Language Processing, University of Stuttgart, Azenbergstr. 12, D-70174 Stuttgart, Germany. E-mail: juergen@ims.uni-stuttgart.de

lem. Van Noord (1993) provided a proof for PATR-style grammars using a reduction to Post's Correspondence Problem. Moreover, a reduction to Hilbert's Tenth Problem was also used by Roach (1983) to show the undecidability of the *emptiness problem* of lexical-functional languages, a result that was later shown by Nishino (1991) using a reduction to Post's Correspondence Problem. In this brief note, we want to investigate the close relationship between the emptiness problem of lexical-functional and PATR languages and the generation problem in (3). We give a much simpler undecidability proof of the emptiness problem using a reduction to the emptiness problem of the intersection of arbitrary context-free languages, a reduction that Wedekind and Kaplan (1996) used to show the undecidability of ambiguity-preserving generation. The close connection of the problems—already indicated by the fact that their undecidability proofs were achieved by the same reductions—results, then, from the fact that the undecidability of the emptiness problem trivially implies the undecidability of semantic-driven generation. This result also applies to other unification-based formalisms such as HPSG, since they are powerful enough to simulate context-free derivations.

We begin our construction by defining for each context-free language L a unification grammar that generates L and that associates with each derivable terminal string an f -structure consisting of the string's difference list encoding (plus concatenation information).¹ For the association of the annotated information with the constituents described by a context-free rule of the form $A \rightarrow w$, we use—similar to PATR—a set of distinct metavariables $\{x_0, \dots, x_{|w|}\}$; x_0 refers to the mother and x_i ($i = 1, \dots, |w|$) to the i th daughter.

Definition

Let G be a context-free grammar in Chomsky normal form whose nonterminal vocabulary, terminal vocabulary, start-symbol, and rules are given by $\langle V_N, V_T, S, R \rangle$. I.e., each rule has the form $A \rightarrow \epsilon$, $A \rightarrow a$ or $A \rightarrow BC$ with $A, B, C \in V_N$, $a \in V_T$ and ϵ denoting the empty string. A **string grammar** $String(G)$ for G is a unification grammar $\langle V_N, V_T, S, R_s \rangle$ whose rule set is determined as follows. In the first step we construct for each context-free rule $r = A \rightarrow w$ a set of annotations S_r :

$$S_r = \begin{cases} \{(x_0 \text{ IN}) \approx (x_0 \text{ OUT})\} & \text{if } w = \epsilon \\ \{(x_0 \text{ IN FIRST}) \approx a, (x_0 \text{ IN REST}) \approx (x_0 \text{ OUT})\} & \text{if } w = a \\ \{(x_0 \text{ IN}) \approx (x_1 \text{ IN}), (x_1 \text{ OUT}) \approx (x_2 \text{ IN}), (x_0 \text{ OUT}) \approx (x_2 \text{ OUT})\} & \text{if } w = BC. \end{cases}$$

The set of rules is then given by $R_s = \{r, S_r \mid r \in R\}$.²

Figure 1 illustrates the f -structure encoding of a terminal string generated by a simple string grammar. By induction on the depth of the derivation trees, it can easily be shown that G and $String(G)$ have the same language and that the f -structure assigned to a terminal string w encodes w , as stated more precisely in the following Lemma:

1 We separated this construction out of the main proof, since it might be useful for analyzing other problems.

2 We used PATR-style notation, since it facilitates the construction of string grammars. For LFG grammars where we do not have the possibility to refer from one daughter to her sister (necessary for $(x_1 \text{ OUT}) \approx (x_2 \text{ IN})$) we need a slightly more complex construction. If $w = BC$ then B has to be annotated by $(\uparrow B_1) \approx \downarrow$ and $(\uparrow \text{ IN}) \approx (\downarrow \text{ IN})$ and C by $(\uparrow C_2) \approx \downarrow$, $(\uparrow \text{ OUT}) \approx (\downarrow \text{ OUT})$, and $(\uparrow B_1 \text{ OUT}) \approx (\downarrow \text{ IN})$. If $w = a$ we need $(\uparrow \text{ IN FIRST}) \approx a$ and $(\uparrow \text{ IN REST}) \approx (\uparrow \text{ OUT})$ and for $w = \epsilon$ the equation $(\uparrow \text{ IN}) \approx (\uparrow \text{ OUT})$. With this construction we get the same undecidability results for classical LFG grammars. The only difference is that the constructed grammars are tree grammars rather than string grammars.

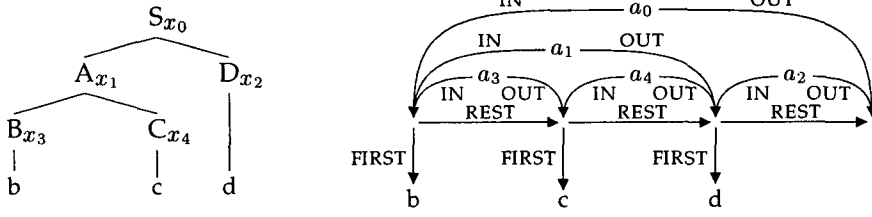


Figure 1

A sample constituent structure and the associated f-structure provided by a simple string grammar. The metavariables of the rules are instantiated by the variables attached to the nodes of the constituent structure. To each variable x_i is assigned the f-structure element a_i .

Lemma

Let $String(G)$ be a string grammar. Then $L(G) = L(String(G))$ and if there is a derivation of a terminal string w with root S_{x_0} and f-structure Φ then the substructure of Φ which comprises the elements accessible from a_0 in Φ is a minimal solution of $\{(x_0 \text{ IN REST}^{i-1} \text{ FIRST}) \approx w_i \mid 1 \leq i \leq |w|\} \cup \{(x_0 \text{ OUT}) \approx (x_0 \text{ IN REST}^{|w|})\}$.³

If we combine two arbitrary string grammars in such a way that the string encodings of the derived terminal strings get unified, we can show the undecidability of the emptiness problem by a simple reduction to the emptiness problem of the intersection of arbitrary context-free languages.

Theorem

It is undecidable for an arbitrary unification grammar G whether $L(G) = \emptyset$.

Proof

Let $G^1 = \langle V_N^1, V_T^1, S^1, R^1 \rangle$ and $G^2 = \langle V_N^2, V_T^2, S^2, R^2 \rangle$ be context-free grammars for two arbitrary context-free languages. Without loss of generality, we can assume that $V_N^1 \cap V_N^2 = \emptyset$ and that each rule in R^i ($i = 1, 2$) is in Chomsky normal form. On the basis of $String(G^1)$ and $String(G^2)$ we construct a unification grammar $G = \langle V_N, V_T, S, R \rangle$ with

$$V_N = V_N^1 \cup V_N^2 \cup \{S\} \text{ and } S \notin V_N^1 \cup V_N^2$$

$$V_T = V_T^1 \cup V_T^2$$

$$R = \{\langle S \rightarrow S^1 S^2, \left\{ \begin{array}{l} x_0 \approx x_1, x_0 \approx x_2, \\ (x_0 \text{ OUT FIRST}) \approx \# \end{array} \right\} \rangle\} \cup R_s^1 \cup R_s^2$$

such that $\#$ is a new atomic value not in V_T . If we assume for G constant-consistency (i.e., axioms of the form $\vdash a \not\approx b$ for all atomic values $a, b \in V_T \cup \{\#\}$ with $a \neq b$) then the problem whether $L(G) = \emptyset$ reduces to the undecidable problem whether $L(G^1) \cap L(G^2) = \emptyset$. In order to get a derivation of a well-formed terminal string $w^1 w^2$ from S with w^1 derived from S^1 and w^2 from S^2 , w^1 must be identical with w^2 , since both string encodings get unified by the S -rule and $(x_0 \text{ OUT FIRST}) \approx \#$ ensures that one string is not a proper prefix of the other.⁴ Thus, $L(G) = \{w w \mid w \in L(G^1) \cap L(G^2)\}$ and $L(G) = \emptyset$ iff $L(G^1) \cap L(G^2) = \emptyset$. ■

³ The whole f-structure encodes the complete difference list derivation of $wx - x$, which is induced by the derivation tree by relabeling each (nonterminal) node dominating substring v of $uvz = w$ by $vzx - zx$, since the annotations of each rule of the form $A \rightarrow BC$ encode the difference list of the mother as the concatenation of the lists of its daughters ($X - X_2 = X - X_1 + X_1 - X_2$).

⁴ The annotation $(x_0 \text{ OUT FIRST}) \approx \#$ is not necessary if acyclicity is assumed.

By taking the smallest f-structure \perp as an input the undecidability of our generation problem reduces trivially to the undecidability of the emptiness problem, since

$$L(G) = \{w \mid \exists \Phi(\Delta_G(w, \Phi))\} = \{w \mid \exists \Phi(\perp \sqsubseteq \Phi \wedge \Delta_G(w, \Phi))\}.$$

That is, if the emptiness problem of $L(G)$ is undecidable for a unification grammar G then G 's generation problem in (3) must be undecidable too. (The other direction does not hold, of course.)

Corollary

For an arbitrary unification grammar G and an arbitrary f-structure Φ' it is undecidable whether there is an f-structure Φ and a terminal string w such that $\Phi' \sqsubseteq \Phi$ and $\Delta_G(w, \Phi)$.

Although it might be argued that we show the undecidability on the basis of a rather special case, namely the smallest f-structure, the undecidability of the emptiness problem is nevertheless sufficient, since we always get a (superficially) less trivial direct proof of the corollary by using *any* proof of the theorem and adding some (new) nontrivial input information to the S-rule. If we add, for example, the equation $(x_0 \text{ SEM}) \approx 1$ to the S-rule of our proof

$$\langle S \rightarrow S^1 S^2, \left\{ \begin{array}{l} x_0 \approx x_1, x_0 \approx x_2, \\ (x_0 \text{ OUT FIRST}) \approx \#, \\ (x_0 \text{ SEM}) \approx 1 \end{array} \right\} \rangle$$

then the problem whether we can find for $[SEM \ 1]$ ($= \Phi'$) an f-structure Φ and a terminal string w such that $[SEM \ 1] \sqsubseteq \Phi$ and $\Delta_G(w, \Phi)$ reduces to the undecidable problem whether $L(G) = \emptyset$ as well.⁵

Our construction shows that an LFG or PATR grammar G can simulate the valid computations of an arbitrary Turing machine M , since they are known to be specifiable by the intersection of two context-free languages. Since $L(M) = \emptyset$ is undecidable, the emptiness problem of $L(G)$ must be undecidable too. By adding a bit of semantic representation Φ' to the S-rule these properties are trivially carried over from $L(G)$ to the set of possible realizations assigned to Φ' by G , given by the language $\{w \mid \exists \Phi(\Phi' \sqsubseteq \Phi \wedge \Delta_G(w, \Phi))\}$. Our proof construction works, of course, even if the grammatical formalisms satisfy the off-line parsability restriction.⁶ Thus, the decidability of the membership problem—similar to context-sensitive grammars—does not imply the decidability of the emptiness (and the semantic-driven generation) problem.⁷

From a cognitive point of view it seems quite unrealistic that our language generation capabilities require mathematical models of Turing machine power. Hence, natural language grammars (of the LFG and PATR formalisms) must satisfy conditions that do not allow us to show the undecidability of the problem. We assumed the semantic representations to be structurally unrelated to the f-structures they subsume. It seems more plausible that there is a proportion k that bounds the size of an

⁵ Van Noord (1993) used the equation $(x_0 \text{ SOLUTION}) \approx \text{yes}$ in his proof.

⁶ If the context-free grammars G^1 and G^2 are off-line parsable then the unification grammars G used in the undecidability proofs are off-line parsable as well. Since we can decide $\epsilon \in L(G')$ for any context-free grammar G' and can reduce G' to an off-line parsable grammar G'' with $L(G') - \{\epsilon\} = L(G'')$, $L(G^1) \cap L(G^2) = \emptyset$ and hence $L(G) = \emptyset$ must be undecidable even if the grammars satisfy the off-line parsability restriction.

f-structure Φ assigned to a string by the size of its subsuming semantic representation Φ' : $|\Phi| \leq k|\Phi'|$. This would force the f-structures of the surface realizations of a semantic representation Φ' given by $\{\Phi \mid \Phi' \sqsubseteq \Phi \wedge \exists w(\Delta_G(w, \Phi))\}$ to be included in a finite and computable set of structurally related f-structures $\{\Phi \mid \Phi' \sqsubseteq \Phi \wedge |\Phi| \leq k|\Phi'|\}$. Since the generation problem is decidable (Wedekind 1995), i.e., $\{w \mid \Delta_G(w, \Phi)\} = \emptyset$ is decidable for any given f-structure Φ , and only a finite number of structurally related f-structures Φ has to be tested for $\{w \mid \Delta_G(w, \Phi)\} = \emptyset$, semantic-driven generation must be decidable. But we must, of course, admit that it is far from being evident yet, how this structural relation is realized in natural language grammars.

Acknowledgments

Ron Kaplan had the idea that the proof construction which we used in Wedekind and Kaplan (1996) might be useful for other purposes. Thanks to him for valuable suggestions.

References

- Dymetman, M. 1991. Inherently reversible grammars, logic programming and computability. In *Proceedings of the ACL Workshop: Reversible Grammar in Natural Language Processing*. Berkeley, CA, pages 20–30.
- Kaplan, R. M. and J. Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA, pages 173–281.
- Nishino, T. 1991. Mathematical analysis of lexical-functional grammars. *Language Research*, 27(1): 119–141.
- Pollard, C. and I. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago, IL.
- Roach, K. 1983. LFG languages over a one-letter alphabet. Manuscript, Xerox PARC, Palo Alto, CA.
- 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, editors, *Research on Interactive Acquisition and Use of Knowledge*. SRI Final Report 1894. SRI International, Menlo Park, CA, pages 39–79.
- van Noord, G. 1993. *Reversibility in Natural Language Processing*. Ph.D. thesis, Rijksuniversiteit Utrecht.
- Wedekind, J. 1995. Some remarks on the decidability of the generation problem in LFG- and PATR-style unification grammars. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics*. Dublin, pages 45–52.
- Wedekind, J. and R. M. Kaplan. 1996. Ambiguity-preserving generation with LFG- and PATR-style grammars. *Computational Linguistics*, 22(4): 555–558.

⁷ This fact is already illustrated by the languages of the grammars we used in the undecidability proofs; they all have a decidable membership problem, since $w \in L(G^1) \cap L(G^2)$ is decidable for arbitrary context-free grammars G^1 and G^2 .

