STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **XLV**, Number 1, 2000

PHRASE GENERATION IN LEXICAL FUNCTIONAL GRAMMARS AND UNIFICATION GRAMMARS

DOINA TĂTAR, DANA AVRAM

ABSTRACT. In this paper we compare the process of deriving a phrase structure in a lexical functional grammars with the process of obtaining feature structure for the symbol S of an unification grammar. If the $c-structure(D, C, e)$ generates the feature structure F, then F is the feature structure obtained as $MGSat(\psi)$, where ψ is a conjunction of a set of descriptions from $Desc$.

1. LEXICAL FUNCTIONAL GRAMMAR-*LFG*

LFG is a lexical theory, this means that the lexicon contains a lot of information about lexical entries. *LFG* grammars present two separate levels of syntactic representation: *c-structure*, about constituent structures (in much the same way as derivation trees in CFG grammars) and *f-structure* , which is used to hold information about functional relations, encoded using equations between feature structures (see the next section). We will introduce here the design of the grammar rules and the lexicon, as well as the process applied to derive a phrase.

Definition

A *LFG* grammar over a set *Feats* of attributes and a set *Types* of types is a 5-uple (N,T,P,L,S) where:

- N is a finite set of symbols, called nonterminals;
- T is a finite set of symbols called terminals;
- P is a finite set of production rules

$$A_0 \rightarrow A_1, \cdots, A_n$$
$$E_1, \cdots, E_n.$$

where $n \geq 1$, $A_1, \cdots, A_n \in N$ and E_i, $1 \leq i \leq n$, is a finite set of equations of the forms:

$$\uparrow | \downarrow \phi = \uparrow | \downarrow \phi'$$

2000 *Mathematics Subject Classification.* 68Q42.

1998 *CR Categories and Descriptors.* F.4.2 [**Theory of computation**]: Mathematical Logic and Formal languages – *Grammars and other rewriting systems.*

$$\uparrow|\downarrow \phi'' = v$$

with $\phi, \phi' \in Feats^*, \phi'' \in Feats^+$ and $v \in Types$;

- L is a finite set of lexicon rules

$$A \to t$$
$$E$$

where $A \in N,\ \ t \in T \cup \varepsilon$ and E is a finite set of equations of the form

$$\uparrow|\downarrow \phi = v$$

with $\phi \in Feats^+$ and $v \in Types$;

- $S \in N$ is the start symbol.

As an example let us consider the rule:

$$S \to NP \qquad VP$$
$$\uparrow subj = \downarrow \uparrow = \downarrow$$

The equations (or functional schemes) are interpreted as referring to the feature structures (section 2) associated, in the following way: the meta-variable $\uparrow$ refers to the f-structure that is associated with the head of the rule, $\downarrow$ refers to the f-structure associated with the daughter to which the equation is attached.

The $c-structure$ based on a LFG grammar G is a tree, in much the same way as derivation trees in a CFG grammar, but the nodes are annotated not only with elements from $N \cup T$ but also with sets of equations E. More exactly:

Definition

A tree domain D is a set $D \subseteq N^*$, (where N is the set of natural numbers, and N^* is the Kleene closure of N) such that if $x \in D$ then all prefixes of x are also in D. The out degree $d(x)$ of an element x in tree domain D is the cardinality of the set $\{i \mid xi \in D, i \in N\}$. Let us denote by $term(D)$ the set $\{x \mid x \in D, d(x) = 0\}$.

We can now define a *c-structure* based on a LFG grammar :

Definition[2]

A constituent structure (*c-structure*) based on a LFG grammar $G = (N, T, P, L, S)$ is a triple (D, C, e) where

- D is a finite tree domain;
- C is a function $C : D \longrightarrow N \cup T \cup \{\varepsilon\}$;
- e is a function $e : D \setminus \{\varepsilon\} \longrightarrow \Gamma$ where Γ is the set of all equation sets in P and L, such that $C(x) \in T \cup \{\varepsilon\}$ if $x \in term(D)$, $C(\varepsilon) = S$ and for all $x \in (D - term(D))$, if $d(x) = n$ then

$$C(x) \to C(x_1) \cdots C(x_n)$$
$$e(x_1) \cdots e(x_n)$$

is a production or lexical rule in G.

Definition

A terminal string for a *c-structure* is the string $C(x_1)\cdots C(x_n)$, with $x_1,\cdots,x_n \in term(D)$ and $x_i \leq_{lex} x_{i+1}$ for $i=1,\cdots,n-1$.

The existence of a *c-structure* is a necessary but not sufficient condition as terminal string belongs to the $L(G)$. Nodes of the *c-structure* are associated with feature structures (denoted by f_i), and the equations induce some equations between f_i as unknowns. The minimal solution of this set of equations (if a solution exists) represents a feature structure F.

Definition

The *c-structure*(D,C,e) generates the feature structure F if F is the minimal solution of the set of equations e. We denote this by

$$F \models' \bigcup_{x\in D} e(x).$$

In the next section we will present unification grammars and will illustrate the connection between unification grammars and LFG grammars.

2. Unification Based Phrase Structure Grammars.

The unification grammars are phrase structure grammars in which non-terminal and terminals symbols are replaced by feature structures. Intuitively, a feature structure (FS) is a description of some linguistic object, specifying some or all of the information that is asserted to be true of it [3, 5]. We will present shortly two definitions of (untyped) feature structures.

Definition:

A feature structure over a signature *Types* and *Feats* is a labeled rooted directed graph represented by the tuple:

$$F = < Q, \bar{q}, \theta, \delta >$$

where :

- Q is the finite set of nodes of the graph;
- $\bar{q} \in Q$ is the root node;
- $\theta : Q \longrightarrow$ **Type** is a *partial* node typing function;
- δ : **Feat** $\times Q \longrightarrow Q$ is a partial value function, which associates with a node i the nodes $i_1,\cdots,i_n$ if $\delta(FEAT_1,i)=i_1,\cdots,\delta(FEAT_n,i)=i_n$.

In the rewriting relations two notions about FS's are important: subsumption relation and unification operation.

Definition

A feature structure F *subsumes* another feature structure G or $F \sqsubseteq G$ iff:

• if a feature $f \in \mathbf{Feat}$ is defined in F then f is also defined in G and its value in F *subsumes* the value in G;

• if the values of two paths are shared in F , then they are also shared in G.

Thus, $F \sqsubseteq G$ if G contains more information than F or F is *more general* than G.

The notion of subsumption can be used to define the notion of unification, the main information combining operation in unification based grammars. Unification conjoins the information in two feature structures into a single result if they are consistent and detects an inconsistency otherwise.

Definition

The result of the unification of two FS's F and F' is an other FS (if it exists), denoted $F \sqcup F'$ which is *the most general* FS (in the sense of relation $\sqsubseteq$) subsumed by both input FS's.

Thus, $F \sqcup F'$ is the l. u. b of F and F' , if it exists, on the ordering relation $\sqsubseteq$.

The FS's can be described, as an other modality, by a logical expression, which is denoted "description". The big advantage of this kind of representing FS's is the linearity of displaying.

Definition [1] The set of descriptions over the set **Types** of types and **Feats** of features is the least set, $Desc$, such that:

$\sigma \in Desc$, if $\sigma \in \mathbf{Types}$

$\pi : \phi \in Desc$ if π is a path, $\phi \in Desc$

$\pi_1 \doteq \pi_2 \in Desc$, if π_1 and π_2 are paths

$\phi \wedge \psi, \phi \vee \psi \in Desc$, if $\phi, \psi \in Desc$

The priority among the operations is:

$$\doteq | \; : | \; \wedge \; | \; \vee \; |$$

A satisfaction relation between FS's and the set $Desc$ is defined as:

Definition The relation $\models$ is the least relation such that:

$F \models \sigma$ if $\sigma \in \mathbf{Types}$, $\sigma \sqsubseteq \theta(\overline{q})$

$F \models \pi : \phi$ if $F@\pi$ is defined and $F@\pi \models \phi$

$F \models \pi_1 \doteq \pi_2$ if $\delta(\overline{q}, \pi_1) = \delta(\overline{q}, \pi_2)$

$F \models \phi \wedge \psi$ if $F \models \phi$ and $F \models \psi$

$F \models \phi \vee \psi$ if $FF \models \phi$ or $F \models \psi$.

The following theorem establishes the duality between a (non-disjunctive) description and the most general FS which satisfies this description:

Theorem ([1]). There is a partial function (algorithm)

$$MGSat : Non - Disj - Desc \rightarrow \mathcal{TFS}$$

such that for each ϕ and F

$$F \models \phi \ iff \ \ MGSat(\phi) \sqsubseteq F.$$

($MGSat(\phi)$ is constructed as most general total well typed FS which satisfies ϕ.)

Remark : The algorithm considers recursively the cases of descriptions: σ, $\pi : \phi$, $\pi_1 = \pi_2$, $\phi \wedge \psi$ and construct (learn) $MGSat(\phi)$. The most important case is:

$$MGSat(\phi \wedge \psi) = MGSat(\phi) \sqcup MGSat(\psi).$$

The UBPSG's are phrase structure grammars in which non-terminal or category symbols are replaced by FS's in rewriting rules, the lexical entries are terminals, and an inheritance hierarchy $<$ **Types**, $\sqsubseteq>$ is associated.

UBPSG's was introduced by Shieber (1988) [5] , Gazdar and Mellich (1989) [4].

Definition. (UBPSG) For an inheritance hierarchy $<$ **Types**, $\sqsubseteq>$ with an appropriateness specification, a set **Feats** of features, a set Lex of terminals (lexical entries), a UBPSG is a set of rewriting rules:

$$E_0 \rightarrow E_1 \dots E_n,$$

where each E_i is either a feature structure or a terminal (and in this case $n = 1$).

The interpretation of such a rule is: the category E_0 can consist of an expression of category E_1, followed by the category E_2, etc.

Alternatively, the rewriting rule can be given as:

$$D_0 \rightarrow D_1 \dots D_n$$

where D_i are descriptions, such that

$$E_i = D_i, \text{ if } D_i \text{ is a terminal}, E_i = (\textit{total well}-\textit{typed}\ )MGSat(D_i), \text{ if } D_i \in Desc.$$

Remarks:

If the *c-structure*(D, C, e) generates the feature structure F, then F is the feature structure obtained as $MGSat(\psi)$, where ψ is obtained as conjunction of the set of $Desc$ as follows:

- If an equation refers to a single unknown (with the form: $f_i\pi = v$, f_i being an unknown, π being a path from $Feats^*$, $v \in Types$), then $\pi : v \in Desc$;
- If two equations are as $f_i\pi = v$ and $f_i\pi' = v'$ then $\pi : v \wedge \pi' v' \in Desc$;
- If an equation is of the form $f_i = f_j$, and $f_i \models \phi_i$ and $f_j \models \phi_j$, then $\phi_i \wedge \phi_j \in Desc$.

These remarks can be summarized in the following:

Theorem

If $F \models' \bigcup_{x \in D} e(x)$ then $F \models \psi$, where $\psi = \bigwedge_{\phi \in Desc} \phi$, and ϕ are the descriptions obtained as above.

3. Example

The lexical rules of this example from [3] are:

$$N \longrightarrow' Raluca'$$

$$\uparrow pred =' Raluca', \uparrow pers =' 3', \uparrow nr =' sing'$$

$$N \longrightarrow' marea'$$

$$\uparrow pred =' marea', \uparrow pers =' 3', \uparrow nr =' sing'$$

$$V \longrightarrow' priveste'$$

$$\uparrow pred =' priveste', \uparrow pers =' 3', \uparrow nr =' sing'$$

The nonlexical rules let be:

$$S \rightarrow NP \qquad VP$$

$$\uparrow subj =\downarrow, \uparrow=\downarrow$$

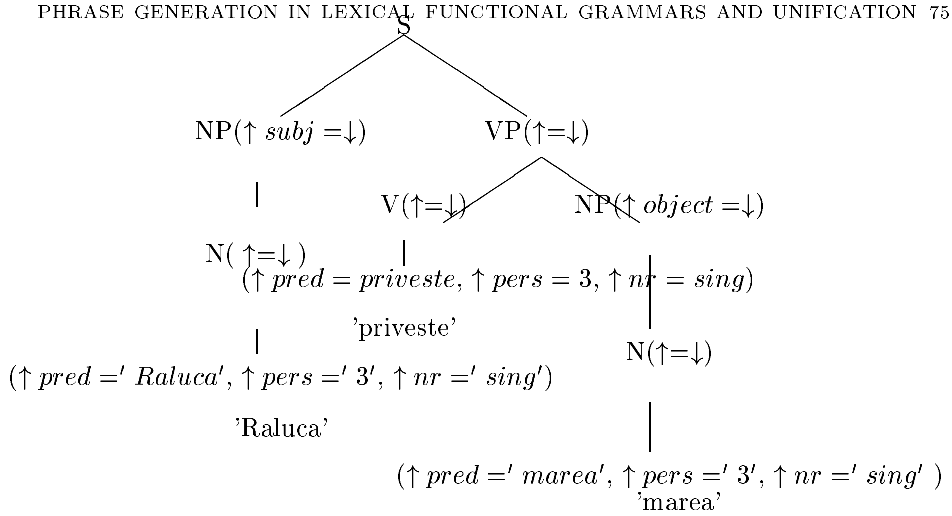
$$VP \rightarrow V \qquad NP$$

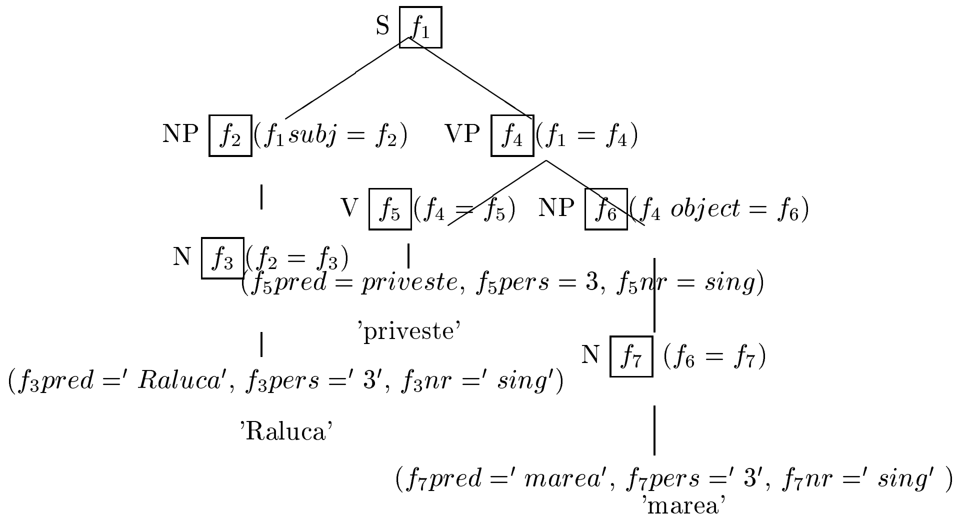
$$\uparrow=\downarrow, \uparrow obj =\downarrow$$

$$NP \rightarrow N$$

$$\uparrow=\downarrow$$

We will construct the $c - structure$ based on the above LFG grammar, than we will proceed to decorate the $c - structure$ by names of feature structures f_i and will apply the equation between them. The decorated $c - structure$ with the instantiated equations attached to its nodes for the above example is also presented as bellow.





We will proceed in the following to obtain the (minimal) solution of the set of equation (or to determining the unsolvability of it).

The steps of this procedure are:

1. Solving the set of equations referring to a single unknown (with the form: $f_i\pi = v$, f_i being an unknown, π being a path from $Feats^*$, $v \in Types$).

2. Interpreting equal unknowns with different values as results of an unification ($f_i \pi v$ and $f_i \pi' v'$ induce the feature structure $\boxed{f_i}\begin{bmatrix}\pi & \text{v} \\ \pi\text{'} & \text{v'}\end{bmatrix}$).

3. Removing the unknowns which are not used effectively by their equals (if $f_i = f_j$ and f_i is not defined, one use f_j).

4. Solving the equations with two feature structure names (if $f_i = a \;\; f_j$, then the feature structure $\boxed{f_i}\begin{bmatrix}\text{a} & \boxed{f_j}\begin{bmatrix} \; \end{bmatrix}\end{bmatrix}$ is obtained).

5. Solving the equations of the form $f_i = f_j$, where both feature structures f_i and f_j are defined, by unification of the values of f_i and f_j and denoting the result as: $\boxed{f_i}\boxed{f_j}\begin{bmatrix}....\end{bmatrix}$

6. As $\boxed{f_1}$ is associated with S , the feature structure for f_1 (if exists), is the feature structure of the entire *correct* phrase.

For the above example, the set of equations is:

$f_1 subj = f_2$
$f_1 = f_4$
$f_2 = f_3$
$f_3 pred =' Raluca'$
$f_3 pers = 3$
$f_3 nr = sing$
$f_4 = f_5$
$f_4 object = f_6$
$f_5 pred =' priveste'$
$f_5 pers = 3rd$
$f_5 nr = sing$
$f_6 = f_7$
$f_7 pred =' marea'$
$f_7 pers = 3rd$
$f_7 nr = sing$

By execution of the above calculus 1-4 steps we obtain the following feature structures:

$$\boxed{f_1}\begin{bmatrix}\text{subj:} & \boxed{f_2}\end{bmatrix}$$

$$\boxed{f_3}\begin{bmatrix}\text{pred:} & \text{'Raluca'} \\ \text{nr:} & \text{sing} \\ \text{pers:} & 3\end{bmatrix}$$

$$\boxed{f_4}\begin{bmatrix}\text{object:} & \boxed{f_6}\end{bmatrix}$$

$$\boxed{f_5}\left[\begin{array}{ll}\text{pred:} & \text{'priveste'}\\ \text{nr:} & \text{sing}\\ \text{pers:} & 3\end{array}\right]$$

$$\boxed{f_7}\left[\begin{array}{ll}\text{pred:} & \text{'marea'}\\ \text{nr:} & \text{sing}\\ \text{pers:} & 3\end{array}\right]$$

From equations $f_1 = f_4$, $f_2 = f_3$, $f_4 = f_5$, $f_6 = f_7$, we obtain the following feature structures:

$$\boxed{f_1}\left[\text{subj: } \boxed{f_2}\boxed{f_3}\left[\begin{array}{ll}\text{pred:} & \text{'Raluca'}\\ \text{nr:} & \text{sing}\\ \text{pers:} & 3\end{array}\right]\right]$$

$$\boxed{f_4}\left[\text{object: } \boxed{f_6}\boxed{f_7}\left[\begin{array}{ll}\text{pred:} & \text{'marea'}\\ \text{nr:} & \text{sing}\\ \text{pers:} & 3\end{array}\right]\right]$$

$$\boxed{f_5}\left[\begin{array}{ll}\text{pred:} & \text{'priveste '}\\ \text{nr:} & \text{sing}\\ \text{pers:} & 3\end{array}\right]$$

For the equations $f_1 = f_4$, $f_4 = f_5$, we apply the step 5 as above and we obtain:

$$\boxed{f_1}\boxed{f_4}\boxed{f_5}\left[\begin{array}{l}\text{pred: 'priveste'}\\[1em] \text{subj:}\boxed{f_2}\ \boxed{f_3}\left[\begin{array}{ll}\text{pred:} & \text{'Raluca'}\\ \text{nr:} & \text{sing}\\ \text{pers:} & 3\end{array}\right]\\[1em] \text{object:}\boxed{f_6}\ \boxed{f_7}\left[\begin{array}{ll}\text{pred:} & \text{'marea'}\\ \text{nr:} & \text{sing}\\ \text{pers:} & 3\end{array}\right]\end{array}\right]$$

The same feature structure can be obtained from descriptions as at the end of section 2.

4. Conclusions.

In this paper we replace the construction of a feature structure, given as the most general satisfier of a conjunction of descriptions, by obtaining the solution of a set of lexical rules equations. The bases of this replacing are the remarks expressed by the theorem at end of section 2.

References

[1] B.Carpenter: "The logic of typed feature structures",Cambridge University Press, 1992.
[2] T. Burheim: "Indexed languages and unification grammars" , Univ. Bergen, Norway, Internal Raport.
[3] N.Francez, S. Wintner: " Feature structure based linguistic formalisms", draft 1998, http.
[4] G. Gazdar, C. Mellish: " NLP in Prolog. An introduction to CL", Addison Wesley, 1989.
[5] M.Shieber: "Introduction to unification-based approaches to grammars", CSLI, 1986.
[6] D. Tatar, M. Lupea: "Indexed grammars and unification grammars", Studia Univ. "Babeş-Bolyai", Informatica, 1998, nr 1 , pp 39-46.

Department of Computer Science, Faculty of Mathematics and Computer Science, "Babeş-Bolyai" University, Cluj-Napoca, Romania
E-mail address: `dtatar|davram@cs.ubbcluj.ro`