CONTEXTUAL GRAMMARS IN PROLOG

Paul SABATIER

Laboratoire d'Automatique et de Linguistique
Universite Paris 7
Tour Centrale 9 eme
2 Place Jussieu
75005 PARIS

ABSTRACT

We present a formalism and a technique by which left and/or right contextual constraints can be easily expressed and computed efficiently in Prolog grammars (avoiding transport of variables): the Contextual Grammars (CG), interpreted in PROLOG II.

Each rule has the form:

NT -> CONTEXT BODY.

where NT is a non-terminal symbol. BODY is a sequence of one or more items separated by blanks. Each item is either a non-terminal symbol, a terminal symbol or a condition. Symbols and conditions are terms (as in Metamorphosis or Definite Clauses grammars); BODY may be empty.

If CONTEXT is not empty, it has the form:

{ L # R }

L and R are sequences of non-terminal and/or terminal symbols separated by points. We read it as:

Apply NT if, in the derivation tree,
1) L precedes NT, and
2) R follows NT.

L or R may be empty.

For example, the following is a sample contextual grammar (terminal symbols are in brackets, and conditions are preceded by "+"):

```
sentence(S)   -> np(_)  vp(S).
np(X.Y)       -> noun(X)  [and]  noun(Y).
np(X)         -> noun(X).
noun(day)     -> [day].
noun(night)   -> [night]
vp(S)         -> verb(S).
vp(S)         -> verb(S)  preposition  np(_).
preposition   -> [with].
```

```
(a)  verb(alternate(X,Y)) -->  { np(X.Y) # }
                               +different(X,Y)
                               [alternate].

(b)  verb(alternate(X,Y)) -->  { noun(X) # [with].noun(Y) }
                               +different(X,Y)
                               [alternates].
```

The sentences produced/analysed from (a) are:

day and night alternate.
night and day alternate.

and from (b):

day alternates with night.
night alternates with day.

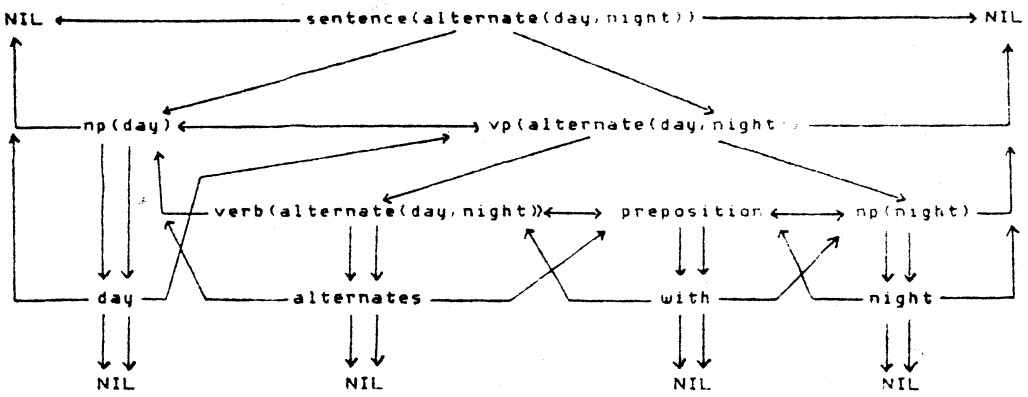
The technique consists in building, during the parsing, an internal derivation graph G containing the sufficient information to recover the context whenever a contextual constraint must be satisfied before the rule must be applied. To each node Ni (corresponding to a non-terminal or terminal symbol) of G, are associated four nodes Nj, Nk, Nl and Nm:

- Nj is the left sibling of Ni; Nj is the left sibling of the parent of Ni if Ni has no left sibling;
- Nk is the first child of Ni; Nk is NIL if Ni has no children;
- Nl is the last child of Ni; Nl is Nk if Ni has one child; Nl is Nk is NIL if Ni has no children;
- Nm is the right sibling of Ni; Nm is the right sibling of the parent of Ni if Ni has no right sibling;

The right sibling and the left sibling of the axiom-symbol of the grammar are NIL.

Here, for example, is the final derivation graph of the sentence:

day alternates with night.



Contextual constraints are computed directly from G. When any part of a context is not yet known (as for example right context of a symbol in a left-to-right parser), the computation is delayed by means of the GELER (FREEZE) predicate.