STUDIA UNIV. "BABEŞ-BOLYAI", INFORMATICA, Volume I, Number 1, March 1996

COMPILING DEFINITE CLAUSE GRAMMARS

D. TĂTAR

Abstract. The original motivation for Chomsky's phrase structure grammar was the description of natural languages (NL). The most frequently used of them, CFG's, are not in general regarded as adequate for all the aspects that occurs in NL processing. A generalisation of CFG's, definite clause grammars (DCG) is better: in this paper we introduce some methods of the association of a DCG to a CFG and the connections between the queries addressed to first and the language generated by the second.

1. Introduction

It is very well known how a CFG can describe a subset of a NL, as in the following example

Example 1.1. *Sample grammar 1.*

$$G = (I_N, I_T, S, P)$$
$$I_N = \{sentence, verb, noun - phrase, article, adjective, noun, preposition\},$$
$$I_T = \{'give', 'the', 'a', 'best', 'last', 'definition', 'notion', 'of'\},$$
$$S = sentence,$$
$$P = \{sentence \to verb, noun - phrase,$$
$$noun - phrase \to article, adjective, noun,$$
$$noun - phrase \to article, adjective, noun, preposition, noun - phrase$$
$$verb \to' give'$$
$$article \to' a'$$
$$article \to' the'$$

Received by the editors: September 21, 1996.

1991 *Mathematics Subject Classification.* 68N17, 68S05.

1991 *CR Categories and Descriptors.* D.1.6 [Programming Techniques]: Logic programming; F.4.1 [Mathematical Logic and Formal Languages]: Mathematical Logic; I.2.7 [Artificial Intelligence]: Natural Language Processing.

$$adjective \rightarrow' best'$$
$$adjective \rightarrow' last'$$
$$noun \rightarrow' definition'$$
$$noun \rightarrow' notion'$$
$$preposition \rightarrow' of'\}$$

The production rules allow us to verify that the string:

$$'give','the','best','definition','of','the','last','notion'$$

is a legal sentence of CFG in this sample.

Let us, on the other hand, regard a Horn clause in a definite program:

$$h : -a_1, a_2, \cdots, a_n$$

or

$$h \leftarrow a_1, a_2, \cdots, a_n$$

We read the implication $h \leftarrow a_1, a_2, \cdots, a_n$ as a production rule : $h \rightarrow a_1, a_2, \cdots, a_n$ which is interpreted as "*h is well-formed if* " $a_1, a_2, \cdots, a_n$ *is well-formed* and such a production rule corresponds to each Horn clause.

A first apparent difference between a production rule and a clause is the order in which the element of the right-hand part of the production rule: the order of elements in the body of a clause is immaterial, due to the commutativity of the conjunction operation. In fact, the precise order of elements in the body of a clause is pushed in the precise order of elements of a list.

We will define a first type of DCG associated with a CFG, and we will call them "definite clause grammar of first type" (DCG-FT).

Definition 1.2. *Let* $G = (I_N, I_T, S, P)$ *a CFG for a sequence of NL, as in sample 1, where* I_N *is a set of syntactic categories and* I_T *is a set of words terminal. A DCG-FT associated with* G *is a* **definite program** P*, defined as follows:*

- *for each non-terminal* X *in* I_N *there exists a predicate* p_X *with one argument such that* $p_X(Y)$ *succeeds iff* Y *is a list of syntactic category* X.
- *for each production rule* $X \rightarrow X_1, X_2 \cdots X_n$ *there exists a Horn clause :*

$$p_X([Y_1|Y_2|\cdots Y_n]) : -p_{X_1}(Y_1), \cdots, p_{X_n}(Y_n)$$

- *for each production rule* $X \to t, t \in I_T$ *there exists a Horn clause* $p_X([t])$.

In this condition, the following theorem is proved.

Theorem 1.3. *Let* $G = (I_N, I_T, S, P)$ *be a CFG. A word* $w = w_1, w_2, \cdots, w_n \in L(G)$, *where* $w_i \in I_T$,$i = 1, \cdots, n$, *iff*

$$p_S([w_1, w_2, \cdots, w_n])$$

succeeds in DCG-FT, associated with G *as in above definition.*

Proof. We will prove by induction on the length of the deduction a more general result:

For $\forall X$, $X \Rightarrow *w_1, w_2, \cdots, w_n$, where $w_i \in I_T$,$i = 1, \cdots, n$, iff $p_X([w_1, w_2, \cdots, w_n])$ *succeeds* in DCG-FT, associated with G as in above definition.

If the length of the deduction is 1, then it is $X \to t$, and by definition $p_X([t])$ succeds. Let us suppose that $\forall X$, $X \Rightarrow *w_1, w_2, \cdots, w_n$, is a deduction of length $k+1$.This deduction is of the form

$$X \to X_1, X_2 \cdots X_m \Rightarrow *w_1, \cdots, w_n$$

Then, there exist the lists $L_1, L_2, \cdots, L_m$ such that:

$$1.[w_1, w_2, \cdots, w_n] = [L_1|L_2|\cdots|L_m]$$

and

$$2.X_i \Rightarrow *L_i, i = 1, \cdots, m$$

are the deductions of length $\leq k$. By induction assumption $p_{X_i}(L_i)$ succeds, $i = 1, \ldots, m$.

Since for the rule

$$X \to X_1, X_2 \cdots X_m$$

there exists a Horn clause : $p_X([Y_1|Y_2|\cdots|Y_m]) : -p_{X_1}(Y_1), \cdots, p_{X_m}(Y_m)$ and $p_{X_i}(L_i)$ succeds, $i = 1, \cdots, m$, then that $p_X([L_1|\cdots|L_m])$ succeds too.

As $[w_1, w_2, \cdots, w_n] = [L1|L2|\cdots|Lm]$, that means that $p_X([w_1, \cdots, w_n])$ succeds.

For the converse implication, we will proceed by induction on the length n of the argument list. If $n = 1$ than $p_X([w_1])$ succeeds and, by above definition, $X \to w_1$. Let us suppose that

$$p_X([w_1, w_2, \cdots, w_n])$$

succeeds. Accordingly with the above definition, there exists a Horn clause

$$p_X([Y_1|Y_2|\cdots Y_m]) : -p_{X_1}(Y_1), \cdots, p_{X_m}(Y_m)$$

and $[w_1, \cdots, w_n]$ is obtained by concatenation of an instantiation of the list $[Y_1|\cdots|Y_m]$ .Let $[L_1|\cdots|L_m]$ be this instantiation. As $p_{X_i}(L_i)$ succeeds, $i = 1, \ldots, m$,and the length of L_i is $< n$, by induction assumption we can say that $X_i \Rightarrow *L_i$,$i = 1, \ldots, m$.

As $X \to X_1, X_2 \cdots X_m$, then $X \to X_1, X_2 \cdots X_m \Rightarrow *L_1, \cdots, L_m = w_1, \cdots, w_n$ □

Example 1.4. *The DCG-FT associated with the sample grammar 1 is the following:*

```
domains
  lista=symbol*
predicates
  sentence(lista)
  verb(lista)
  noun_ph1(lista)
  noun_ph2(lista)
  article(lista)
  adjective(lista)
  noun(lista)
  prep(lista)
clauses
  sentence([VE|NP]):-verb([VE]),noun_ph1(NP).
  sentence([VE|NP]):-verb([VE]),noun_ph2(NP).
  noun_ph1([AR,AD,NO]):-article([AR]),
                        adjective([AD]),
                        noun([NO]).
  noun_ph2([AR,AD,NO,PP|NP]):-article([AR]),
                        adjective([AD]),
                        noun([NO]),
                        prep([PP]),
                        noun_ph1(NP).
  verb([give]).
  article([the]).
```

```
adjective([best]).
adjective([last]).
noun([definition]).
noun([notion]).
prep([of]).
```

If the goal addressed to this DCG is : $sentence(X)$,then we will obtain: $X = [give, the, best, definition, of, the, last, notion\ ]$ and others 19 solutions. In the set of solution we find the following: $[give, the, best, notion, of, the, last, definition]$, which is, surely, not semantically correct. In this point another tool that is introduced by DCG can be useful, namely the " procedure calls". [3]. These are some relations between the arguments of symbols from I_N. The translation of this condition is easily realized by a DP,where the relational operators are permitted. Another generalisation realised by a DCG, apart from CFG, is the possibility of realisation of the number and the person agreement.

Example 1.5. *Sample grammar 2.*

$$sentence \rightarrow pronoun, verb, article, noun$$
$$pronoun \rightarrow' I'$$
$$pronoun \rightarrow' you'$$
$$pronoun \rightarrow' he'$$
$$pronoun \rightarrow' we'$$
$$pronoun \rightarrow' you'$$
$$pronoun \rightarrow' they'$$
$$article \rightarrow' a'$$
$$noun \rightarrow' teacher'$$
$$noun \rightarrow' teachers'$$
$$verb \rightarrow' am'$$
$$verb \rightarrow' are'$$
$$verb \rightarrow' is'$$
$$verb \rightarrow' are'$$

In this CFG grammar we can obtain the following incorrect sentence: $'We','is','the','teachers'$.

Let us transform CF production rules as:

$$sentence(X,Y) \rightarrow pronoun(X,Y), verb(X,Y), article, noun(X).$$

$$pronoun(1,1) \rightarrow' I'$$

$$pronoun(1,2) \rightarrow' you'$$

$$pronoun(1,3) \rightarrow' he'$$

$$pronoun(2,1) \rightarrow' we'$$

$$pronoun(2,2) \rightarrow' you'$$

$$pronoun(2,3) \rightarrow' they'$$

$$article \rightarrow' a'$$

$$noun(1) \rightarrow' teacher'$$

$$noun(2) \rightarrow teachers'$$

$$verb(1,1) \rightarrow' am'$$

$$verb(1,2) \rightarrow' are'$$

$$verb(1,3) \rightarrow' is'$$

$$verb(2,1) \rightarrow' are'$$

$$verb(2,2) \rightarrow' are'$$

$$verb(2,3) \rightarrow' are'$$

In this DCG grammar, the sentence as above cannot be obtained. For example, we can obtain the following correct sentence:

$$sentence(1,1) \rightarrow pronoun(1,1), verb(1,1), article, noun(1) \Rightarrow *'I','am','a','teacher'$$

A rewriting of a word in this DCG grammar is obtained by a double process: an application of the relation $\Rightarrow *$ as in the usual definition and a *unification* process. This is a common fact with the *unification grammar* as in [1].

The DCG-FT obtained as in the above procedure is the following:

```
domains
 lista=symbol*
predicates
 sentence(integer,integer,lista)
 pronoun(integer,integer,lista)
 verb(integer,integer,lista)
 noun(integer,lista)
```

```
article(lista)

clauses
sentence(X,Y,[Pn,V,Ar,N]):-pronoun(X,Y,[Pn]),
                           verb(X,Y,[V]),
                           article([Ar]),
                           noun(X,[N]).

pronoun(1,1,[i]).
pronoun(1,2,[you]).
pronoun(1,3,[he]).
pronoun(2,1,[we]).
pronoun(2,2,[you]).
pronoun(2,3,[they]).
article([a]).
noun(1,[teacher]).
noun(2,[teachers]).
verb(1,1,[am]).
verb(1,2,[are]).
verb(1,3,[is]).
verb(2,1,[are]).
verb(2,2,[are]).
verb(2,3,[are]).
```


The goal: $sentence(1,1,X)$ addressed to this DCG-FT obtains,
$X = ['I','am','a', teacher']$.

2. Obtaining the parse tree

If we want to obtain the syntactic tree (the parse tree) of a sentence,we must modify the rules of association of DCG to a CFG.

Let us consider the following sentence in the Sample grammar 1:

$$'give','the','best', definition','of','the','last','notion'$$

The above sentence can be obtained by the concatenation of all the leaves from the parse tree, from left to right. This tree can also be represented by means of the below

construction, which suggests better the process of generation of a word:

$$sentence(v('give'), np(art('the'), adj('best'), n('definition'),$$
$$prep('of'), np(art('the'), adj('last'), n('notion'))))$$

We will call such a tree an *annotated* tree. It is obtained by reading the tree top-down, such that each production rule $X \to X_1, X_2, \cdots, X_n$, where $s_X, s_{X_1}, s_{X_2}, \cdots, s_{X_n}$ are syntactic categories of $X_1, X_2, \cdots, X_n$, is *annotated* with $s_X(s_{X_1}, \cdots, s_{X_n})$.

A DCG-FT which obtains the syntactic trees is defined as follows.

Definition 2.1. *Let $G = (I_N, I_T, S, P)$ a CFG for a sequence of NL. A DCG-FT associated with G for the syntactic tree is a definite program P defined as follows:*

- *for each non-terminal X in I_N which occurs in a production rule $X \to X_1, \cdots, X_n$ and which is of the syntactic category s_X, there exists a predicate*

$$p_X(s_X(V_1, \cdots, V_n)), [V_1', \cdots, V_n'])$$

such that $p_X(Y, Z)$ succeeds iff Y is of the syntactic category of p_X and Z is a list of the syntactic categories of its descendents.

- *for each production rule $X \to X_1, X_2 \cdots X_n$ there exists a Horn clause:*

$$p_X(s_X(V_1, V_2, \cdots V_n), [V_1', V_2', \cdots, V_n']) : -p_{X_1}(V_1, [V_1']), \cdots, p_{X_n}(V_n, [V_n'])$$

- *for each production rule $X \to t, t \in I_T$ there exists a Horn clause*

$$p_X(s_X(t), [t]).$$

In this case, the following theorem is proved.

Theorem 2.2. *Let $G = (I_N, I_T, S, P)$ be a CFG . A word $w = w_1, w_2, \cdots, w_n \in L(G)$ where $w_i \in I_T$,$i = 1, \ldots, n$, iff $p_S(A, [w_1, w_2, \cdots, w_n])$ succeeds in DCG-FT for the syntactic tree , associated with G as in the above definition, with A being the annotated tree of w.*

The DCG-FT for the syntactic tree, for CFG in *Sample grammar 1*, obtained as in the above procedure, is the following:

```
domains
  dom=v(symbol);art(symbol);adj(symbol);no(symbol);
  np(dom,dom,dom);sn(dom,dom);
```

```
  npc(dom,dom,dom,dom,dom);p(symbol)
  lists=symbol*
predicates
  sentencet(dom,lists)
  verbt(dom,lists)
  noun_pht1(dom,lists)
  noun_pht2(dom,lists)
  articlet(dom,lists)
  adjectivet(dom,lists)
  nount(dom,lists)
  prept(dom,lists)
clauses
  sentencet(sn(VE,NP),[V|N]):-verbt(VE,[V]),noun_pht1(NP,N).
  sentencet(sn(VE,NP),[V|N]):-verbt(VE,[V]),noun_pht2(NP,N).
  noun_pht1(npc(AR,AD,NO,PP,NP),[Art,Adj,Nn,Prep|Nph]):-
                        articlet(AR,[Art]),
                        adjectivet(AD,[Adj]),
                        nount(NO,[Nn]),
                        prept(PP,[Prep]),
                        noun_pht2(NP,Nph).
 noun_pht2(np(AR,AD,NO),[Art,Adj,N]):-articlet(AR,[Art]),
                        adjectivet(AD,[Adj]),
                        nount(NO,[N]).
  verbt(v(give),[give]).
  articlet(art(the),[the]).
  adjectivet(adj(best),[best]).
  adjectivet(adj(last),[last]).
  nount(no(definition),[definition]).
  nount(no(notion),[notion]).
  prept(p(of),[of]).
```

For the goal: *sentence*(X, Y) this program obtains 20 solutions, in which Y is a correct sentence, and X is the *annotated* syntactic tree associated with it.

3. Definite clause grammar of second type

Another sort of DCG associated with a CFG, which can describe a syntactic tree of a sentence, if it is correct, is the DCG of the second type (DCG-ST) obtained from the below procedure.

Definition 3.1. *Let $G = (I_N, I_T, S, P)$ a CFG . A DCG-ST associated with G is a definite program P defined as follows:*

- *for each non-terminal X in I_N there exists a predicate p_X with two arguments such that $p_X(Y, Z)$ succeeds iff Y is a list whose head is of the syntactic category X, and whose tail is the list Z.*
- *for each production rule $X \to X_1, X_2 \cdots X_n$ there exists a Horn clause:*

$$p_X(Y, Z) : -p_{X_1}(Y, Y_1), \cdots, p_{X_{n-1}}(Y_{n-2}, Y_{n-1}), p_{X_n}(Y_{n-1}, Z)$$

- *for each production rule $X \to t, t \in I_T$ there exists a Horn clause*

$$p_X(Y, Z) : -Y = [t|Z]).$$

The following theorem can be proved.

Theorem 3.2. *Let $G = (I_N, I_T, S, P)$ be a CFG. A word $w = w_1, w_2, \ldots, w_n \in L(G)$, where $w_i \in I_T$,$i = 1, \cdots, n$, iff*

$$p_S([w_1, w_2, \cdots, w_n], [])$$

succeeds in DCG-ST, associated with G as in the above definition.

In the following we will build the DCG-ST associated with the sample grammar 3.

Example 3.3. *Sample grammar 3.*

$$sentence \rightarrow subject, verb, complement$$
$$sentence \rightarrow article, noun$$
$$complement \rightarrow adjective$$
$$article \rightarrow ['the']$$
$$noun \rightarrow ['work']$$
$$noun \rightarrow ['child']$$
$$adjective \rightarrow ['nice']$$
$$adjective \rightarrow ['sage']$$
$$verb \rightarrow ['is']$$

```
 lista=symbol*
predicates
 sentence(lista,lista)
 subject(lista,lista)
 verb(lista,lista)
 complement(lista,lista)
 article(lista,lista)
 adjective(lista,lista)
 noun(lista,lista)
clauses
 sentence(Y,Z):-subject(Y,Y1),
                verb(Y1,Y2),
                complement(Y2,Z).
 subject(Y,Z):-article(Y,Y1),noun(Y1,Z).
 complement(Y,Z):-adjective(Y,Z).
 article(Y,Z):-Y=[the|Z].
 noun(Y,Z):-Y=[work|Z].
 noun(Y,Z):-Y=[child|Z].
 verb(Y,Z):-Y=[is|Z].
 adjective(Y,Z):-Y=[nice|Z].
```

If the goal is: $sentence(Y, [])$, then the solution is: $Y = ['the','child','is','nice']$

References

[1] M. Johnson, *Attribute-value logic and the theory of grammar*, CSLI, 1988.

[2] D. Tatar, *Logic grammars as formal languages*, Studia Universitas "Babes-Bolyai", Mathematica, XXXIX (1994), pp. 75-82.

[3] A. Thayse (ed), *From standard logic to logic programming*, John Wiley & Sons, 1988.

"BABEŞ-BOLYAI" UNIVERSITY, FACULTY OF MATHEMATICS AND COMPUTER SCIENCE, RO-3400 CLUJ-NAPOCA, ROMANIA

E-mail address: dtatar@cs.ubbcluj.ro