PUBLICATIONS DE L'INSTITUT MATHÉMATIQUE
Nouvelle série, tome 56 (70), 1994, 1–6

REPRESENTATION OF METAMORPHOSIS GRAMMAR IN LOGIC GRAMMAR: PROOF TREES AND THEIR LENGTHS

Marica D. Prešić

Communicated by Žarko Mijajlović

Abstract. We consider the representations of metamorphosis grammar in logic grammar, more precisely in Horn predicate logic, developed in [C078] on the Colmerauer idea of defference lists. For the proofs in metamorphosis grammar in normal form the so called *normal length* of the proof is defined and it is shown that this length equals to the length of the corresponding proof in logic grammar.

If V is some vocabulary by V^* we denote the set of all words on V (including the empty word e) and by $V^{[\,]}$ the set of all lists on V (including the empty list $[\,]$). The word has been defined in some standard way, for example as a finite sequence $a_1, a_2, \dots, a_n$ (we omit the comas in the case there is no possibility of confusion). The list $[a_1, a_2, \dots, a_n]$ has been defined as the following term: $a_1 \cdot (a_2 \cdot \dots (a_n \cdot nil) \dots)$ built up from a binary functional symbol $\cdot$ so called *list constructor* and a constant symbol *nil* playing a role of empty list $[\,]$. It is supposed that neither $\cdot$ nor *nil* belong to V. The singleton list $[a]$, i.e. the term $a \cdot nil$ is also denoted by $\underline{a}$.

For the length of either a word or a list α we use the denotation $|\alpha|$. With words and lists the operation of *concatenation* written as product is defined recursively in the standard way.

Let F be a set of functional symbols containing a binary functional symbol $\cdot$ and a constant symbol *nil*, and let X be a set of variables. Then Term(F, X) or shortly Term is the set of terms built up from F, X and $H(F)$ or shortly H is the *Herbrand universe*, i.e. the set of terms containing no variables (we ad a new constant symbol to form $H(F)$ if necessary). Following [CO78] with a slight modification a *metamorphosis grammar* on F is defined as a quadruple $G = (N, T, S, P)$, where: $T \subseteq H$ is a vocabulary of *terminals*, $N \subseteq H$ is a vocabulary of *non-terminals* [We suppose that $T \bigcap N \neq \emptyset$ and define $V = T \bigcup N$.], $S \subseteq N$ is the set of *starting* non-terminals, P is the set of *rewriting rules* on $V^{[\,]}$ [With the restriction: if $\alpha \longrightarrow \beta$ belongs

AMS Subject Classification (1991): Primary 03 B 65 to P then $\alpha \neq nil$.]. The *language* generated by G is the set: $L(G) = \{t \in T^{[]} \mid$ There exists $s \in S$ with $\underline{s} \longrightarrow t\}$.

We recall that a metamorphosis grammar G is in *normal* form if each its rewriting rule is of the form:

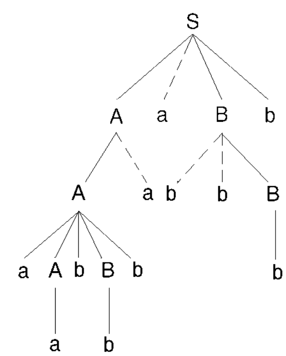
$$\underline{A}\alpha \longrightarrow \beta \tag{1}$$

where A is a non-terminal symbol and α is a *terminal* list, i.e. belongs to $T^{[]}$. In the case α is empty list for each rewriting rule, G is called *definite clause grammar*. In [C078] it has be proved that every metamorphosis grammar G can be transformed into the corresponding grammar G' in normal form such that the language $L(G)$ equals to $L(G')$. The similar property holds for Chomsky grammars.

The proofs in grammars in normal form (either Chomsky one or metamorphosis one) have the appropriate tree representations similar to those for context-free and definite clause grammars. For example the proof:

$$S \longrightarrow AaBb \longrightarrow AaaBb \longrightarrow AaabbBb \longrightarrow aAbBbBb \longrightarrow aabbbbb \tag{2}$$

in the Chomsky grammar $G = (\{S, A, B\}, \{a, b\}, S, P)$ with the set P of rewriting rules: $S \longrightarrow AaBb$, $A \longrightarrow Aa$, $B \longrightarrow bbB$, $Aaabb \longrightarrow aAbBb$, $A \longrightarrow a$, $B \longrightarrow b$ can be represented with the given tree. The dashed edges appear after application of the rule $Aaabb \longrightarrow aAbBb$.



We say that the nodes on the right side: a, A, b, B, b have *grown up* after application of this rule, and that the terminal nodes on the left side: a, a, b, b have *dried up* and therefore their edges become dashed.

As both Chomsky and metamorphosis grammar are a kind of formal theories, for a given proof π in either of these grammar the *length* $|\pi|$ of the proof would be the total number of atoms occurring in π (empty atoms are not counted). Particularly in the case of context-free of definite clause grammar this number equals to the number of nodes in the corresponding proof tree.

Consider a rewriting rule in a context free (definite clause) grammar:

$$A \longrightarrow b_1 b_2 \ldots b_n \quad [\underline{A} \longrightarrow \underline{b}_1 \underline{b}_2 \ldots \underline{b}_n]$$

where A is a non-terminal and $b_1, b_2, \ldots, b_n$ are either terminals or non-terminals. If we consider it as a proof (obtained by one application of the rule), its length is $1 + n$, which is in accordance with the previous discussion. This proof is a special kind of proofs which "start with A" and it can in a natural way be expanded to a proof of the form $\pi(A)$:

$$A \longrightarrow^* \pi(b_1)\pi(b_2)\ldots\pi(b_n) \quad [\underline{A} \longrightarrow^* \pi(b_1)\pi(b_2)\ldots\pi(b_n)]$$

where $\pi(b_1), \pi(b_2), \ldots, \pi(b_n)$ are some proofs which start with $b_1, b_2, \ldots, b_n$ respectively. If b_i is a terminal, $\pi(b_i)$ reduces to b_i [to $\underline{b}_i$] in which case we have $|\pi(b_i)| = 1$. It is easy to verify the following equality

$$|\pi(A)| = 1 + |\pi(b_1)| + |\pi(b_2)| + \ldots + |\pi(b_n)| \tag{3}$$

Let now G be a grammar in normal form (Chomsky one or metamorphosis one) and π a proof in G. Apart from the length $|\pi|$ we introduce another kind of length called *normal* and denote it by $\|\pi\|$, in which the nodes of both kinds (grown-ups and dried-ups) are taken into account.

Definition. Let $\pi(S)$ be a proof in a Chomsky grammar (the only difference in the case of metamorphosis grammar is the first equality which becomes $\underline{S} = \pi_0$) built up step by step by the sequence of proofs:

$$S=\pi_0, \pi_1, \ldots, \pi_i, \;\; \pi_{i+1}, \ldots, \pi_m = \pi(S)$$

the members of which *expand each other* meaning that each member has been obtained from the preceding one by one application of the rule $A\alpha \longrightarrow \beta$, where $A \in N$, $\alpha \in T^*$, $\beta \in V^*$ [$A \in N, \alpha \in T^{[]}, \beta \in V^{[]}$], the corresponding normal length is defined by the following recursion:

$$\|\pi_0\| = 1, \quad \|\pi_{i+1}\| = \|\pi_i\| + |\alpha| + |\beta|. \quad \square \tag{4}$$

If, for example, in the previous proof in a Chomsky grammar the following m rules ($S, A_1, \ldots, F_1$ are non-terminals and $\alpha_1, \beta_1, \ldots, \phi_1$ are terminal words):

$$S \longrightarrow \alpha, A_1\alpha_1 \longrightarrow \beta, B_1\beta_1 \longrightarrow \gamma, \ldots, F_1\phi_1 \longrightarrow \theta$$

have respectively been employed. Then the number $\|\pi(S)\|$ satisfies the equality:

$$\|\pi(S)\| = 1 + |\alpha| + (|\alpha_1| + |\beta|) + (|\beta_1| + |\gamma|) + \ldots + (|\phi_1| + |\theta|) \tag{5}$$

As the proof $\pi(S)$ starts with the non-terminal S (or with singleton list $\underline{S}$), every terminal which has been dried up in some step π_i of the proof has to be previously grown up in some step which comes before π_i. Thus, as an important consequence of the definition, we have that each dried node is counted twice in $\|\pi(S)\|$: first time it has been grown up, second time it has been dried up. For example, the normal length of the proof (2) is determined by the following equality:

$$\|\pi(S)\| = 1 + 4 + 2 + 3 + (4 + 5) + 1 + 1 + 1 = 22$$

We now consider the representation of metamorphosis grammar in logic grammars, more precisely in Horn logic grammars, based on the idea of difference lists developed in [CO78]. Suppose that $G = (N, T, S, P)$ is a metamorphosis grammar on F which is supposed to be in normal form. The corresponding first order Horn logic grammar G_c is constructed in the following way:

The set of its functional symbols is $F = T \cup \{\cdot, nil\}$, where nil is a new constant symbol and $\cdot$ is a new binary functional symbol (empty list and list constructor). The set X of variables is some countable set of symbols for example:

$X = \{x_0, x_1, x_2, x_3, \dots\}$. The corresponding sets of terms and Herbrand universe are *Term* and H respectively. The set of predicate symbols of G_c is determined as follows: For each non-terminal S, A, B, C,... from N one predicate symbol $\mathbf{s}$, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$,... is introduced. If $R = r(t_1, \dots, t_r)$ is some non-terminal from N built up from a functional symbol r and terms $t_1, \dots, t_r$, the length of the corresponding predicate symbol $\mathbf{r}$ is determined as $length(\mathbf{r}) = length(r) + 2$. The sets of axioms and rewriting rules of G_c is constructed in following way:

(i) If $\underline{A}\alpha \longrightarrow \beta$ is a rule from P where $A \in N$, $A = a(t_1, \dots, t_a)$ and α, β are terminal lists, then the following formula belongs to the axioms of G_c (ν is a variable from X):

$$\mathbf{a}(t_1, \dots, t_a, \alpha\nu, \beta\nu) \tag{6}$$

(ii) If $\underline{A}\alpha \longrightarrow \beta_0\underline{B}\beta_1\underline{C}\beta_2 \dots \underline{D}\beta_n$ is a rule from P, where non-terminals $A, B, \dots, D$ ($n + 1$ of them) are of the form:

$$A = a(t_{10}, \dots, t_{a0}),\ \ B = b(t_{11}, \dots, t_{b1}),\ \ C = c(t_{12}, \dots, t_{c2}), \dots,$$
$$D = d(t_{1n}, \dots, t_{dn})$$

and $\alpha, \beta_0, \beta_1, \dots, \beta_n \in T^{[]}$ then the following Horn predicate formula belongs to the the rules of G_c ($\nu_0, \nu_1, \dots, \nu_n$ are variables from X):

$$\mathbf{b}(t_{11}, \dots, b_{b1}, \beta_1\nu_1, \nu_0) \wedge \mathbf{c}(t_{12}, \dots, t_{c2}, \beta_2\nu_2, \nu_1) \wedge \dots \wedge$$
$$\mathbf{d}(t_{1n}, \dots, t_{dn}, \beta_n\nu_n, \nu_{n-1}) \Rightarrow \mathbf{a}(t_{10}, \dots, t_{a0}, \alpha\nu_n, \beta_0\nu_0)$$

for which we use the standard grammatical denotation in the form of rewriting rule:

$$\begin{aligned} &\mathbf{a}(t_{10}, \dots, t_{a0}, \alpha\nu_n, \beta_0\nu_0) \longrightarrow \mathbf{b}(t_{11}, \dots, t_{b1}, \beta_1\nu_1, \nu_0), \\ &\mathbf{c}(t_{12}, \dots, t_{c2}, \beta_2\nu_2, \nu_1), \dots, \mathbf{d}(t_{1n}, \dots, t_{dn}, \beta_n\nu_n, \nu_{n-1}) \end{aligned} \tag{7}$$

We recall that in [CO78] for each non-terminal $R = r(t_1, \dots, t_r)$ from N the following equivalence: $\underline{R} \longrightarrow^* t$ iff $\vdash \mathbf{r}(t_1, \dots, t_p, [\,], t)$ in logic grammar G_c has been proved, where $t \in T^{[]}$ and $\mathbf{r}$ is the predicate symbol corresponding to the non-terminal $R = r(t_1, \dots, t_r)$.

We now confine to the problem of efficiency of the above representation. One natural way of treating the efficiency of a logic grammar is to count the length of (the shortest) proof of a theorem, which in the case of Horn logic grammar is equal to the total number of atomic formulae occurring in the proof tree.

As in the rule (7) this number equals to the number of the non-terminals occurring in the rule $\underline{A}\alpha \longrightarrow \beta_0\underline{B}\beta_1\underline{C}\beta_2 \dots \underline{D}\beta_n$ from which (7) has been born, it seems at the first sight that the total number of atomic formulae occurring in a proof of G_c would be the total number of the non-terminals in the corresponding proof tree of the original grammar G. But the things are not as simple as that. For in the rule (7) we also have concatenation of lists which has not been taken into account. If we use for concatenation a standard recursive definition by means of predicate *append*:

$$append\,([\,], X, X)$$

$$append\ ([X|Y], Z, [X|U]) \longrightarrow\ append(Y, Z, U)$$

then, for example, if $\underline{A}[a,b] \longrightarrow [a,a]\underline{A}[b,b]\underline{B}[a]$ is a rule of G, the corresponding rule of G_c would be:

$$\mathbf{a}([a,b]w, [a,a]u) \longrightarrow \mathbf{a}([b,b]\nu, u),\ \ \mathbf{b}([a]w, \nu)$$

or after introduction the predicate *append*:

$$\begin{aligned}\mathbf{a}(w'', u') \longrightarrow\ & append\ ([a,a], u, u'),\ \ \mathbf{a}(\nu', u),\ \ append\ ([b,b], \nu, \nu'),\\ & \mathbf{b}(w', \nu),\ \ append\ ([a], w, w'),\ \ append\ ([a,b], w, w'')\end{aligned}$$

Further, as *append* has been defined recursively on the first list, the number of atoms which are necessary to prove a formula of the form *append* (X, Y, Z) is equal to the length of list X. Which in the case of our example means that the total number of new nodes introduced after application of the above rule equals to:

$$|[a,a]| + |\mathbf{a}(\nu', u)| + |[b,b]| + |\mathbf{b}(w', \nu)| + |[a]| + |[a,b]| = 2+1+2+1+1+2 = 9$$

This sequence of equalities corresponds to the following sequence which in the original grammar G gives the total number of new nodes (grown-ups and dried-ups) introduced after application of the rule $\underline{A}[a,b] \longrightarrow [a,a]\underline{A}[b,b]\underline{B}[a]$:

$$|[a,a]\underline{A}[b,b]\underline{B}[a]| + |[a,b]| = 7 + 2 = 9$$

Such correspodence holds generally. Namely, if we transform the rules (6), (7) into the form in which the concatenation of lists occurs explicitly, we obrain:

$$\mathbf{a}(t_1, \ldots, t_a, \nu'', \nu') \longrightarrow\ \ append\ (\beta, \nu, \nu'),\ append\ (\alpha, \nu, \nu'') \tag{8}$$

$$\begin{aligned}\mathbf{a}(t_{10}, \ldots, t_{a0}, \nu''_n, \nu'_0) \longrightarrow\ & append\ (\beta_0, \nu_0, \nu'_0),\ append\ (\alpha, \nu_n, \nu''_n)\\ & \mathbf{b}(t_{11}, \ldots, t_{b1}, \nu'_1, \nu_0),\ append\ (\beta_1, \nu_1, \nu'_1),\\ & \mathbf{c}(t_{12}, \ldots, t_{c2}, \nu'_2, \nu_1),\ append\ (\beta_2, \nu_2, \nu'_2),\\ & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ & \mathbf{d}(t_{1n}, \ldots, t_{dn}, \nu'_n, \nu_{n-1}),\ append\ (\beta_n, \nu_n, \nu'_n),\end{aligned} \tag{9}$$

If we further introduce the atomic formulae born on the basis of recursive application of *append*, it is easy to see that the total number of atomic formulae which would appear on the right side of the sign $\longrightarrow$ equals to:

$$|\beta| + |\alpha|, \quad |\beta_0| + \cdots + |\beta_n| + n + |\alpha|$$

respectively. Thus, in the corresponding proof trees in the grammar G_c the nodes $\mathbf{a}(t_1, \ldots, t_a, \nu'', \nu')$, $\mathbf{a}(t_{10}, \ldots, t_{a0}, \nu''_n, \nu'_0)$ have exactly $|\beta| + |\alpha|$, i.e. $|\beta_0| + \cdots + |\beta_n| + n + |\alpha|$ children respectively, which equals to the number of nodes (grown-ups and dried-ups) which appear after one application of the rules:

$$\underline{A}\alpha \longrightarrow \beta, \quad \underline{A}\alpha \longrightarrow \beta_0\underline{B}\beta_1 \ldots \underline{D}\beta_n \tag{10}$$

in the proof tree of the original metamorphosis grammar G (in normal form). This means that between proof trees in metamorphosis grammar and the corresponding trees in logic grammars there is a natural one-to-one correspondence.

THEOREM. *Let G be a metamorphosis grammar in normal form and G_c the corresponding Colmerauer Horn logic grammar. Then the total number of atomic formulae occurring in a proof in the definite clause grammar G_c equals to the normal length of the corresponding proof in the original grammar G. Thus let*

$$S \longrightarrow^* t, \quad \vdash \mathbf{s}(t_1, \dots, t_s, [\,], t)$$

be corresponding proofs in the grammars G, G_c, where $t \in T^{[\,]}$, $S{=}s(t_1,\dots,t_s)$ is a starting term of G and $\mathbf{s}$ is the related predicate of the logic grammar G_c. Denote these proofs by $\pi(S)$, $\pi_c(S)$ respectively. Then:

$$|\pi_c(S)| = \|\pi(S)\| \tag{11}$$

Proof. The equality can be proved by induction on length of the proof $\pi_c(S)$. Thus consider the sequence of proofs in the grammar G_c:

$$\mathbf{s}(t_1, \dots, t_s, [\,], t) = \pi_0,\ \ \pi_1, \dots, \pi_i,\ \ \pi_{i+1}, \dots, \pi_m = \pi_c(S)$$

the members of which expand each other. Supposing that π_{i+1} has been obtained by application of the rule corresponding to $\underline{A}\alpha \longrightarrow \beta$, where β is terminal list or $\beta = \beta_0\underline{B}\beta_1 \dots \underline{D}\beta_n$ where $A,B,C,\dots,D$ are non-terminals and $\alpha, \beta_0, \beta_1, \dots, \beta_n$ are terminal lists from $T^{[\,]}$, then on the basis of the rules (8), (9) it is easy to check that the following equalities:

$$|\pi_{i+1}| = |\pi_i| + |\alpha| + |\beta|,\ \ |\pi_{i+1}| = |\pi_i| + |\beta_0| + \cdots + |\beta_n| + n + |\alpha| = |\pi_i| + |\alpha| + |\beta|$$

hold respectively, for in the case $\beta = \beta_0\underline{B}\beta_1 \dots \underline{D}\beta_n$ we have:

$$\begin{aligned} |\beta| = |\beta_0\underline{B}\beta_1 \dots \underline{D}\beta_n| &= |\beta_0| + |\underline{B}| + |\beta_1| + |\underline{C}| + |\beta_2| + \cdots + |\underline{D}| + |\beta_n| \\ &= |\beta_0| + |\beta_1| + \cdots + |\beta_n| + n \end{aligned}$$

Thus for the above sequence of proofs in the grammar G_c we immediately obtain the recurrence formulae:

$$|\pi_0| = 1, \quad |\pi_{i+1}| = |\pi_i| + |\alpha| + |\beta| \tag{12}$$

which are of the same kind as the formulae (4) by which the normal length for the corresponding sequence of proofs in the original grammar G has been defined, wherefrom the claim of the theorem follows immediately. ☐

REFERENCE

[CO78] A. Colmerauer, *Metamorphisis grammars*, In: L. Bolc, ed., *Natural Language Communication with Computers*, Lecture Notes in Computer Science, **63**, Springer-Verlag, Berlin, Heidelberg, New York, pp. 133–189

Matematički fakultet
Univerzitet u Beogradu
Studenski trg 16
11000 Beograd, Jugoslavija
E-mail: epresic@ubbg.etf.bg.ac.yu

(Received 06 04 1994)