TREE GRAMMARS (= Δ- GRAMMARS)

- Gladky A. V. (Novosibirsk) & Mel'čuk I. A. (Moscow) -

1. This paper suggests a new kind of formal grammar (hereafter called Δ- grammar*) which in some respects is closely related to Chomsky's grammars but differs from these in that it is meant to process trees (in the sense of graph theory) and not to process strings as Chomsky's grammars do. More precisely, we aim at a type of grammar with rewriting rules of the "X → Y" where X and Y are trees (N.B. : with no linear order imposed on their nodes !**).

Linguistically, the trees under consideration are <u>dependency</u> (not phrase structure) trees representing natural sentences at different levels of "depth" : roughly speaking, "surface" syntax, "deep" syntax, semantics.

Δ- grammars are designed to be used not for generating sentences but rather for transforming given trees into other trees; this covers transitions from one abstract representation of a natural sentence to another (deeper or more superficial) representation of the same sentence as well as transitions from an abstract representation of one sentence to a representation on the same level of another sentence, synonymous to the given one. The conversion of a "ready" surface tree into an actual sentence - a conversion consisting of a) inflexion and b) determination of word order - must be carried out by some autonomous device not included in the conception of Δ-grammar.

---

*. Δ from the Greek δένδρον (tree).

**. The limitations of place and time prevent us from comparing tree grammars with those of Chomsky as well as from referring to other works dealing with more or less analogous matters, such as studies by M. Arapow and V. Borschtschow; G. Veillon, J. Veyrunes and B. Vauquois; Ch. Hockett; and others.

The authors are glad to acknowledge here the friendly help and useful suggestions by O.S. Kulagina and A.Y. Dikovsky.
All shortcomings in the paper are, of course, ours.

The Δ-grammar embodies an attempt to formalize the linguistic "Meaning —Text Model" described, e.g., in [1]. In this model, the starting point for producing a sentence is a detailed semantic description of its meaning conceived as a rather involved graph (not merely a tree) consisting of "semantic atoms" and "semantic links" connecting them. The semantic description is generated outside of the linguistic model and constitutes the input of that model; it is then subsequently "lingualized" (anglicized, russianized etc.) by means of formally specified transformations: 1) extracting from the given semantic description (of a family of synonymous sentences conveying the meaning represented by that description) the deepest admissible tree-like structures; 2) proceeding in a multi-step fashion from the deeper trees to the more superficial ones; 3) linearizing the most superficial syntactic trees (with simultaneous inflexion where needed) to produce actual sentences. The Δ-grammars deal with the second phase of this process only.

2. We shall consider trees with labelled branches; nodes are not labelled. The labels can be interpreted as names of the types of syntactic link at the corresponding level. For brevity's sake such trees will here be referred to just as "trees".

A tree is called minimal if all its nodes, except the root, are terminal (i.e., with no branches growing out of them). A tree with but one node is called an empty tree and is denoted as ϵ. The composition of trees is defined as follows: let t_0, t_1, t_2, ..., t_n be trees, and let in t_0 some nodes α_1, α_2, ..., α_n (not necessarily pairwise different) be marked. Then the result of the composition of the tree t_0 with the trees t_1, t_2, ..., t_n will be any tree isomorphic to the tree which can be obtained from t_0 by identifying the roots of the trees t_1, t_2, ..., t_n with the nodes α_1, α_2, ..., α_n, respectively in t_0.

The composition of t_0 in which the nodes α_1, α_2, ..., α_n are marked with t_1, t_2, ..., t_n is denoted

$$T = C\,(t_0;\ \alpha_1,\ \alpha_2,\ \ldots,\ \alpha_n \mid t_1,\ t_2,\ \ldots,\ t_n) \qquad (1)$$

A tree is a subtree of T if T can be represented as:

$$T = C\,(T_0;\ \alpha_0 \mid C(t\ ;\ \alpha_1,\ \alpha_2,\ \ldots,\ \alpha \mid T_1,\ T_2,\ \ldots\ T_n)) \qquad (2)$$

where α_0 is a terminal node of T_0, and α_1, α_2, ..., α_n a repetitionless enumeration of all nodes of t.

Now, <u>an elementary transformation</u> (ET) of trees is an ordered triple $< t_1, t_2, f >$, where t_1 and t_2 are trees and f is a mapping of the set of all nodes of t_1 into the set of all nodes of t_2. Instead of $<t_1, t_2, f>$, we shall write $t_1 \Rightarrow t_2 | f$. The tree T' is said to be the result of the application of the ET $t_1 \Rightarrow t_2 | f$ to the tree T if T and T' can be represented in the form:

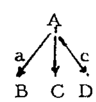
$$T = C\,(T_0;\ \alpha_0 |\ C\,(t_1;\ \alpha_1,\ \alpha_2,\ \ldots,\ \alpha_n |\ T_1,\ T_2,\ \ldots,\ T_n\,)) \qquad (3)$$

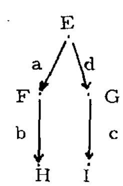
and

$$T' = C\,(T_0;\ \alpha_0 |\ C\,(t_1;\ f(\alpha_1),\ f(\alpha_2),\ \ldots,\ f(\alpha_n)\ |\ T_1,\ T_2,\ \ldots,\ T_n)) \qquad (4)$$

where α_0 is a terminal node of T_0, and $\alpha_1, \alpha_2, \ldots, \alpha_n$ a repetionless enumeration of all nodes of t_1. Informally, an application of certain ET to a tree T consists of the substituting of t_2 for an occurence of t_1 in T ; if α (a node of t_1) is mapped on β(a node of t_2), i.e., $\beta = f(d)$, then all "untouched" nodes of T "pending" from α are transferred to β with the same labels on corresponding branches.

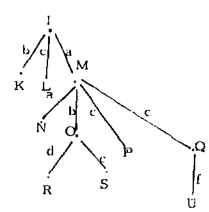
<u>Example</u>:

Let t_1 = 

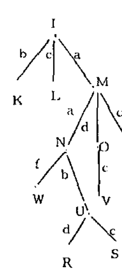
, t_2 = 

and let f be specified

as follows: f(A) = E, f(B) = H, f(D) = F. Then, applying the ET $t_1 \Rightarrow t_2 | f$ to the tree

T = 

we can obtain the tree



T contains three occurences of t ; the replaced one is the subtree of T with the nodes M, N, O, Q.

3. A syntactic Δ-grammar is an ordered pair $\Gamma = \langle V, \Pi \rangle$ where V is a finite set of symbols (branch labels) and a finite set of ET's, called rules of grammar Γ. A derivation in a syntactic Δ-grammar is a finite sequence of trees where each subsequent tree is obtainable from the preceding one by application of an ET of Π. A tree t' is derivable from T in Γ if there exists a derivation in Γ beginning with T and ending with T'.

For linguistic applications, it may prove to be of interest to define some specific types of syntactic Δ-grammars.

A syntactic Δ-grammar will be called expanding if each rule it contains has in its left side no more nodes than in its right.

An expanding syntactic Δ-grammar will be called minimal if in each of its rules of the form "$t_1 \Rightarrow t_2 \mid f$" the trees t_1 and t_2 can be represented in the form

$$t_1 = C(\tau_0 ; \alpha_0 \mid C(\mu ; \alpha_1, \alpha_2, \ldots, \alpha \mid \tau_1, \tau_2, \ldots, \tau_n)) \qquad (5)$$

and

$$t_2 = C(\tau_0 ; \alpha_0 \mid C(V ; f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n) \mid \tau_1, \tau_2, \ldots \tau_n)) \qquad (6)$$

where 1) μ is a minimal tree, 2) $\alpha_1, \alpha_2, \ldots, \alpha_n$ is a repetitionless enumeration of all nodes of μ, 3) α_1 is the root of μ, 4) $f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n)$ are pairwise different, 5) $f(\alpha_2), f(\alpha_3), \ldots, f(\alpha_n)$ are terminal nodes of V, 6) for every $i = 2, 3, \ldots, n$ the label on the branch of μ ending in coincides with the label on the branch of V ending in $f(\alpha_i)$, 7) for all nodes of t differing from $\alpha_1, \alpha_2, \ldots, \alpha_n$, the mapping is identical.

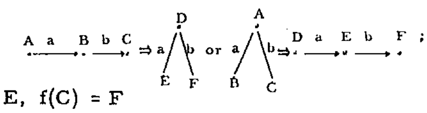
A minimal expanding syntactic Δ-grammar will be called context-free if in the expressions (5) and (6) the trees $\tau_0, \tau_1, \tau_2, \ldots, \tau_n$ are unity trees.

4. Linguistic considerations dealt with in "Meaning↔Text Model" (see, e.g., [1]) imply the introduction of a subset of ET's, special elementary transformations (SET's). A SET is an ET of one of the following three types:

1) Splitting of one node - a transformation of the form $A \Rightarrow B \xrightarrow{a} C$ where either $f(A) = B$ or $f(A) = C$.

Notation: $A \Rightarrow a(B, C) \mid f(A) = B$ and $A \Rightarrow a(B, C) \mid a(A) = C$.

2) Transfer of one node - a transformation of the form



in both cases f(A) = D, f(B) = E, f(C) = F

(Notation : a(A, B). b(B, C) ⇒ a(D, E). b(D, F) <u>and</u> a(A, B). b(A, C) ⇒ a(D, E). b(E, F)).

3) Lumping two nodes into one - a transformation of the form $B \xrightarrow{a} C \Rightarrow A$, where f(B) = f(C) = A

Notation : a(B, C) ⇒ A.

Let $t_1 \Rightarrow t_2 \mid f$ be an ET and let M be a set of ET's. Then the statement "The ET $t_1 \Rightarrow t_2 \mid f$ <u>can be simulated by ET's of M</u>" means that there exists some finite sequence $m_1, m_2, \ldots, m_n$ of ET's in M such that for any trees T and T' where T' can be obtained from T by application of the ET $t_1 \Rightarrow t_2 \mid f$ the tree T' can be obtained from T by applying $m_1, m_2, \ldots, m_n$ in tandem.

<u>Theorem 1</u>. Any elementary transformation can be simulated by special elementary transformations.

5. For the representation of natural sequences it is reasonable to assume not arbitrary syntactic trees but rather a subset of those - namely, those with limited branching. The precise meaning of limited branching is as follows: for each branch label a_i there is fixed an integer n_i such that each node can be a starting ppint at most for n_i branches labelled a_i . The trees meeting this restriction are called $(n_1, n_2, \ldots, n_k)$-<u>regular</u> (<u>k</u> being the number of different branch labels); for brevity we shall call these trees simply regular trees.

Now, a slight modification of the notion of the application of an ET suggests itself: if we suppose that the trees T and T' in (3) - (4) are regular, we need consider only ET's with regular left and right sides; such ET's will also be called regular.

<u>A regular syntactic Δ-grammar</u> is an ordered triple $< V, \zeta, \Pi >$, where $V = \{ a_1, a_2, \ldots, a_k \}$ is a finite set of symbols (branch labels), ζ is a mapping of V into the set of positive integers (for every $a \in V$ the integer $\zeta(a)$ being the maximum number of branches labelled a which can grow out of any single node) and Π is a finite set of $(\zeta(a_1), \zeta(a_2), \ldots, \zeta(a_k))$-regular ET's.

The set of all regular syntactic Δ-grammars may be divided into hierarchical subsets which are fully analogous to the corresponding subsets of the syntactic Δ-grammars as defined above. Special elementary transformations (SET's) can be defined here too.

Theorem 1'. Any $(n_1, n_2, \ldots, n_k)$-regular elementary transformation can be simulated by $(n_1, n_2, \ldots, n_k, 1)$-regular SET's.

Theorem 2. a) If $n_1 + \ldots + n_k \geq 3$ or if $n_1 + \ldots + n_k = 1$, then any $(n_1, n_2, \ldots, n_k)$-regular ET can be simulated by $(n_1, n_2, \ldots, n_k)$-regular SET's.

b) There exists (1, 1)-regular and (2)-regular ET's which cannot be simulated by (1, 1)-regular and (2)-regular SET's respectively.

6. If a regularity characteristics $(n_1, n_2, \ldots, n_k)$ is fixed on the basis of some empirical (linguistic) evidence, then a "universal syntax" can be constructed as an abstract calculus of all possible syntactic structures and all possible transformations of these. Choosing (1, 1, 1, 1, 10, 1) - regularity* as a first approximation to the deep syntactic description of natural languages, we obtain a universal (1, 1, 1, 1, 10, 1)-regular Δ-grammar, $\langle V_\upsilon, \xi_\upsilon, \Pi_\upsilon \rangle$, where $V_\upsilon = \{a_1, a_2, \ldots, a_6\}$ is the set of types of deep syntactic connections and where

$$\xi_\upsilon(a_1) = \xi_\upsilon(a_2) = \xi_\upsilon(a_3) = \xi_\upsilon(a_4) = \xi_\upsilon(a_6) = 1 \; ; \; \xi_\upsilon(a_5) = 10 \; .$$

consists of the following 80 rules:

1) 12 "splitting" rules of the form $A \Rightarrow a_i(B, C) \mid f(A) = B$ and $A \Rightarrow a_i(B, C) \mid f(A) = C \quad (i = 1, \ldots, 6)$

2) 62 "transfer" rules of the form $a_i(A, B) \,.\, a_j(B, C) \Rightarrow a_i(D, E) \,.\, a_j(D, F)$ and $a_i(A, B) \,.\, a_j(A, C) \Rightarrow a_i(D, E) \,.\, a_j(E, F)$;

---

*)The description of deep syntax suggested in [1] is meant here. 6 types of syntactic connections are differentiated and interpreted as follows: connections 1 through 4 link a predicate with its arguments (only predicates with no more than 4 places are considered), connection 5 formalizes the general attributive relation, and connection 6 expresses coordination; a node can be a starting point for only one branch of each of types 1, 2, 3, 4, 6 and for several branches of type 5 (we have set the number of the latter at 10 as a sufficient upper limit).

The set of all regular syntactic Δ-grammars may be divided into hierarchical subsets which are fully analogous to the corresponding subsets of the syntactic Δ-grammars as defined above. Special elementary transformations (SET's) can be defined here too.

Theorem 1'. Any $(n_1, n_2, \ldots, n_k)$-regular elementary transformation can be simulated by $(n_1, n_2, \ldots, n_k, 1)$-regular SET's.

Theorem 2. a) If $n_1 + \ldots + n_k \geq 3$ or if $n_1 + \ldots + n_k = 1$, then any $(n_1, n_2, \ldots, n_k)$-regular ET can be simulated by $(n_1, n_2, \ldots, n_k)$-regular SET's.

b) There exists (1, 1)-regular and (2)-regular ET's which cannot be simulated by (1, 1)-regular and (2)-regular SET's respectively.

6. If a regularity characteristics $(n_1, n_2, \ldots, n_k)$ is fixed on the basis of some empirical (linguistic) evidence, then a "universal syntax" can be constructed as an abstract calculus of all possible syntactic structures and all possible transformations of these. Choosing (1, 1, 1, 1, 10, 1) - regularity* as a first approximation to the deep syntactic description of natural languages, we obtain a universal (1, 1, 1, 1, 10, 1)-regular Δ-grammar, $<V_u, \xi_u, \Pi_u>$, where $V_u = \{a_1, a_2, \ldots, a_6\}$ is the set of types of deep syntactic connections and where

$$\xi_u(a_1) = \xi_u(a_2) = \xi_u(a_3) = \xi_u(a_4) = \xi_u(a_6) = 1 \; ; \; \xi_u(a_5) = 10 \; .$$

consists of the following 80 rules:

1) 12 "splitting" rules of the form $A \Rightarrow a_i(B, C) \mid f(A) = B$ and $A \Rightarrow a_i(B, C) \mid f(A) = C \quad (i = 1, \ldots, 6)$
2) 62 "transfer" rules of the form $a_i(A, B) \,.\, a_j(B, C) \Rightarrow a_i(D, E) \,.\, a_j(D, F)$ and $a_i(A, B) \,.\, a_j(A, C) \Rightarrow a_i(D, E) \,.\, a_j(E, F)$;

---

*) The description of deep syntax suggested in [1] is meant here. 6 types of syntactic connections are differentiated and interpreted as follows: connections 1 through 4 link a predicate with its arguments (only predicates with no more than 4 places are considered), connection 5 formalizes the general attributive relation, and connection 6 expresses coordination; a node can be a starting point for only one branch of each of types 1, 2, 3, 4, 6 and for several branches of type 5 (we have set the number of the latter at 10 as a sufficient upper limit).

here i, j = 1, ..., 6 and either $i \neq j$ or $i = j = 5$.

3) 6 "lumping" rules of the form $a_i(A.B) \Rightarrow C$ ($i = 1, \ldots, 6$).

7. It may be useful, in view of possible linguistic applications, to consider also such regular trees where the branches as well as the nodes are labelled <u>filled regular trees</u>. The node labels may be interpreted as <u>characterized lexemes</u>, i.e., symbols denoting words, idioms and so-called lexical functions with morphological subscripts attached to them ([1], p. 186) The notion of regular ET and that of regular syntactic Δ-grammar can in an obvious manner be modified accordingly. As a result, we obtain <u>regular lexico-syntactic Δ-grammars</u>. For these grammars (see p. 6-7) we can define SET's of the types "splitting","transfer" and "lumping" in a manner analogous to the one above; in addition another type of SET must be introduced:

4) "renaming" of a node - a transformation of the form $\overset{w_i}{\cdot} \Rightarrow \overset{w_j}{\cdot}$,
where w_i <u>and</u> w_j are node labels.

If SET's are understood as transformations of the types 1-4, the theorems 1' and 2 will hold also for this case.

---

[1] Жолковский А К, Мелъчук И А О семантическом синтезе.- Проблемы кибернетики, вып. 19 1967, 177 - 238.