ANALELE ŞTIINŢIFICE ALE UNIVERSITĂŢII "AL. I. CUZA" IAŞI
Tomul II, Informatică, 1993

GRAMMARS WITH ORACLES[1]

BY

L. KARI, AL. MATEESCU, A. SALOMAA, G. PĂUN

Abstract. We define a class of grammars with regulated rewriting based on the idea of oracles: certain nonterminals act as "bifurcation symbols", having associated language oracles; if the left/right neighbouring (maximal) terminal string for a given symbol belongs to the corresponding oracle, then a precisely identified nonterminal is produced, otherwise another one is introduced. We investigate mainly the (rather large) generative power of such grammars and the closure properties of one of the obtained families (it is closed under "hard" operations, like intersection and complement, but not under arbitrary morphisms).

1. Introduction. The study of regulated rewriting is a significant branch of formal language theory, developed (some decades ago) mainly with the aim of increasing the power of context-free grammars (to generate as large families of languages as possible using as simple machineries as possible, extensions of context-free grammars). Many variants of regulating mechanisms were considered (the reader can find details in [1]), but still there is room for further ones. We consider here such a new-old restriction on derivations of context-free grammars, based on the idea of oracles. Special nonterminals are able to ask whether or not a well-defined part of the current sentential form belongs to an associated oracle. Depending on the answer, the derivation continues on one path or another (possibly it is blocked). The construct is somewhat related to the conditional grammars considered in [2], [3], with differences we shall discuss in the last section of this paper. The results we find show the large generative capacity of these mechanisms. (One-letter non-regular languages can be generated with grammars with context-free rules and regular oracles, without probably achieving the power of context-sensitive grammars in this case. This proper inclusion in the family of context-sensitive languages is true in the case of languages generated by grammars with regular rules and context-free oracles, a family with surprising closure properties: it is closed under intersection and complement, but not under λ-free morphisms).

2. Definitions and examples. The reader is referred to [5] for basic elements of formal language theory. We denote by λ the empty string and by $| x |$ the length

---

[1] Research supported by the Academy of Finland, grant 11281, and the Alexander von Humboldt Foundation. All correspondence to Gheorghe Păun.

of a string $x \in V^*$.

Definition. An *oracle grammar* is a construct $G = (N, T, S, P, Q)$, where

$N = N_1 \cup \bigcup_{X \in N_1} \{X_?^L, X_?^R, X_t, X_f\}$, for N_1 a finite alphabet,

$T =$ finite alphabet, $T \cap N = 0$,

$S \in N_1$,

$P \subseteq (T \cup N - \{X_?^d \mid X \in N_1, d \in \{L, R\}\})^+ \times (N \cup T)^*$,

$Q = \{Q_X \subseteq T^* \mid X \in N_1\}$.

(N is the nonterminal alphabet in which the set of basic noterminals, N_1, is distinguished; for each basic nonterminal, a left–asking nonterminal $X_?^L$, a right–asking one $X_?^R$, an answer X_t (*true*), an answer X_f (*false*) are associated; T is the terminal alphabet, S is the axiom, P is the set of rewriting rules, Q is the set of oracles, lanuages over T associated with symbols $X \in N_1$, hence to quadruples $X_?^L, X_X^R, X_t, X_f$; the symbols $X_?^L, X_?^R$ cannot appear on left–hand sides of rules in P.)

For $x, y \in (N \cup T)^*$, we define the relation $x \Longrightarrow y$ if one of the next cases holds:

1. $x = x_1 \alpha x_2, y = x_1 \beta x_2, \alpha \rightarrow \beta \in P$;

2. $x = x_1 A x_2 B_?^L x_3$ (or $x = x_2 B_?^L x_3$), $y = x_1 A x_2 B_a x_3$ (respectively, $y = x_2 B_a x_3$), for $A \in N$, $x_1, x_3 \in (N \cup T)^*$, $x_2 \in T^*$, $B \in N_1$, $a \in \{t, f\}$, and (if $x_2 \in Q_B$ then $a = t$ else $a = f$);

3. $x = x_1 A_?^R x_2 B x_3$, (or $x = x_1 A_?^R x_2$), $y = x_1 A_a x_2 B x_3$ (respectively, $y = x_1 A_a x_2$), for $A \in N_1, x_1, x_3 \in (N \cup T)^*$, $x_2 \in T^*$, $B \in N$, $a \in \{t, f\}$, and (if $x_2 \in Q_A$ then $a = t$ else $a = f$).

(Therefore, $X_?^L$, $X_?^R$ is replaces by X_t when the neighbouring terminal string, to the left of $X_?^L$ and to the right of $X_?^R$, respectively, belongs to Q_X, and by X_f otherwise.)

As usual, $L(G) = \{x \in T^* \mid S \Longrightarrow^* x\}$.

Convention. When specifying an oracle grammar, only the nonterminals used in rules are written (in general, from each qualdruple $X_?^L, X_?^R, X_t, X_f$ only two symbols are used), and only the associated oracle languages are given.

Examples.

$$\begin{aligned} G_1 = \ & (\{S, X, Y, X_?^R, X_t, Y_?^L, Y_f\}, \quad \{a, b\}, S, P_1, Q_1) \\ & P_1 = \{S \rightarrow XY, X \rightarrow X_?^R, X_t \rightarrow aXb, \\ & Y \rightarrow Y_?^L, Y_f \rightarrow aYb, \\ & X \rightarrow ab, Y \rightarrow ab\} \\ & Q_X = Q_Y = \{b^n a^m \mid n + m \text{ even }\}. \end{aligned}$$

Assume we have a sentential form $a^i X b^i a^i Y b^i$, $i \geq 0$(initially $i = 0$). Both X and Y can introduce asking symbols, namely $X_?^R, Y_?^L$, respectively. If the length of the string between the two nonterminals is even, then $X_?^R$ can be replaced by X_t and if this string is of odd length, then $Y_?^L$ can be replaced by Y_f. Then X_t introduces aXb, Y_f introduces aYb, which both modify the parity. Therefore we must first work on the first nonterminal and then on the second one. We obtain

$$\begin{aligned} a^i X b^i a^i Y b^i &\Longrightarrow a^i X_?^R b^i a^i Y b^i \Longrightarrow a^i X_t b^i a^i Y b^i \\ &\Longrightarrow a^{i+1} X b^{i+1} a^i Y b^i \Longrightarrow a^{i+1} X b^{i+1} a^i Y_?^L b^i \\ &\Longrightarrow a^{i+1} X b^{i+1} a^i Y_f b^i \Longrightarrow a^{i+1} X b^{i+1} a^{i+1} Y b^{i+1} \end{aligned}$$

The procedure can be iterated, hence we can obtain one of the following strings

$$a^{i+j} X b^{i+j} a^{i+j} Y b^{i+j}, \qquad i \geq 0, j \geq 0,$$
$$a^{i+j+1} X b^{i+j+1} a^{i+j} Y b^{i+j}, \qquad i \geq 0, j \geq 0.$$

If in one of these strings we use $Y \to ab$, then the checked string will no more be of the form $b^n a^m$, hence introducing $X_?^R$ will block the derivation. Thus we have to use $X \to ab$, too, and we obtain

$$a^{i+j+1} b^{i+j+1} a^{i+j+1} b^{i+j+1}, \qquad i, j \geq 0,$$
$$a^{i+j+2} b^{i+j+2} a^{i+j+1} b^{i+j+1}, \qquad i, j \geq 0,$$

respectively. If we use first the rule $X \to ab$, then we can also use $Y \to ab$ and we obtain strings as above, but we can also follow the cycle $Y, Y_?^L, Y_f, Y$, an arbitrary number of times, because in all moments the string to the left of $Y_?^L$ is not in Q_Y. In conclusion,

$$L(G_1) = \{a^n b^n a^m b^m \mid n = m + 1 \text{ or } m \geq n, \text{ for } n, m \geq 1\},$$

a language which is not context–free although the rules of G_1 are context–free and the oracle languages (the same for X and Y) are regular.

$$\begin{aligned} G_2 = \;& (\{S, A, B, X, Y, X_?^L, X_t, Y_?^L, Y_t\}, \{a, b, c\}, S, P_2, Q_2), \\ & P_2 = \{S \to aS, S \to aA, A \to bA, A \to X_?^L, \\ & \qquad X_t \to B, B \to cB, B \to Y_?^L, Y_t \to c\}, \\ & Q_X = \{a^n b^n \mid n \geq 1\}, \\ & Q_Y = \{a^n b^m c^{m-1} \mid n \geq 1, \quad m \geq 1\}. \end{aligned}$$

It is easy to see that

$$L(G_2) = \{a^n b^n c^n \mid n \geq 1\},$$

again a non–context–free language (this time the rules of the grammar are regular and the oracle languages are context–free; moreover, the oracles are asked only once each, and only for left strings and only the true answer is used).

3. Generative capacity. Denote by REG, CF, CS, RE the families in the Chomsky hierarchy (regular, context–free, context–sensitive, recursively enumerable languages, respectively) and by $OR(F, F')$ the families of languages generated by oracle grammars with rules of type F, $F \in \{REG, CF, CF^\lambda, CS, RE\}$, with the following meaning: REG = right–linear, CF = λ–free context–free, CF^λ = arbitrary context–free, CS = length–increasing, RE = unrestricted, and the oracle languages of type F', $F' \in \{REG, CF, CS, RE\}$.

Theorem 1. *If $L_1, L_2 \subseteq V^*, L_1, L_2 \in F, F \in \{REG, CF, CS, RE\}$, then $L_1, L_1 \cap L_2, V^* - L_1$ are in $OR(REG, F) \cap OR(CF, F)$.*

Proof. Write

$$L_i = (L_i \cap \{\lambda\}) \cup \bigcup_{a \in V} \partial_a^r(L_i)\{a\}, \quad i = 1, 2.$$

All the above families F are closed under the right derivative (denoted here by ∂_x^r), hence $\partial_a^r(L_i) \in F, i = 1, 2, a \in V$. Then construct

$$\begin{aligned}
G_1 = \ & (\{S\} \cup \{S_a, S_{a,?}^L, S_{a,t} \mid a \in V\}, V, S, P_1, Q_1),\\
& P_1 = \{S \to \lambda \mid \ \text{if} \ \lambda \in L_1\} \cup \\
& \quad \cup \{S \to S_a, S_a \to S_{a,?}^L, S_{a,t} \to a \mid a \in V\} \cup \\
& \quad \cup \{S_a \to bS_a \mid a, b \in V\},\\
& Q_{S_a} = \partial_a^r(L_1), \quad a \in V,\\
G_2 = \ & (\{S\} \cup \{S_a, S_{a,?}^L, S_{a,f} \mid a \in V\}, V, S, P_2, Q_2),\\
& P_2 = \{S \to \lambda \mid \ \text{if} \ \lambda \notin L_1\} \cup \\
& \quad \{S \to S_a, S_a \to S_{a,?}^L, S_{a,f} \to a \mid a \in V\} \cup \\
& \quad \cup \{S_a \to bS_a \mid a, b \in V\},\\
& Q_{S_a} = \partial_a^r(L_1), \quad a \in V,\\
G_3 = \ & (\{S, T\} \cup \{S_a, T_a, S_{a,?}^L, T_{a,?}^L, S_{a,t}, T_{a,t} \mid a \in V\}, V, S, P_3, Q_3),\\
& P_3 = \{S \to \lambda \mid \ \text{if} \ \lambda \in L_1 \cap L_2\} \cup \\
& \quad \cup \{S \to S_a, S_a \to S_{a,?}^L, S_{a,t} \to T_{a,?}^L, T_{a,t} \to a \mid a \in V\} \cup \\
& \quad \cup \{S_a \to bS_a \mid a, b \in V\},\\
& Q_{S_a} = \partial_a^r(L_1),\\
& Q_{T_a} = \partial_a^r(L_2), \quad a \in V.
\end{aligned}$$

It is easy to see that $L(G_1) = L_1$, $L(G_2) = V^* - L_1$, $L(G_3) = L_1 \cap L_2$ and G_1, G_2, G_3 contain only λ-free right–linear rules (hence they are of the desired type). ∎

Corollary. *All the inclusions $F' \subset OR(F, F')$, $F \in \{REG, CF, CF^\lambda, CS, RE\}$, $F' \in \{CF, RE\}$, and $CF \subset OR(CF, REG)$ are proper.*

Proof. Neither of the families CF and RE is closed under complement, hence we obtain the inclusions $F' \subset OR(F, F')$ from the theorem. The inclusion $CF \subset OR(CF, REG)$ is proper in view of the first example in the previous section. ∎

Theorem 2. $OR(F, CS) = CS, F \in \{REG, CF, CS\}$.

Proof. The inclusions $\supseteq$ are proved in Theorem 1, the converse inclusions can be proved by a straightforward construction, based on the fact that CS is closed under complement ([4], [6]): start from an oracle grammar G and construct a type-0 grammar with a linearly bounded workspace simulating both the rules of G and the oracles; the passing from some $X_?^L$ to X_t is done when the terminal string to left of $X_?^L$ can be reduced nondeterministically to the axiom of the grammar for the corresponding oracle, whereas the passing to X_f is done when the axiom of the grammar for the complement of the oracle is reached; similary for symbols $X_?^R$. The details of such a construction are left to the reader. ∎

Theorem 3. $OR(F, F') \subseteq RE$ for $F \in \{CF^\lambda, RE\}, F' \in \{REG, CF, CS\}$.

Proof. A similar construction as the previously sketched one, without concern about the workspace.

Theorem 4. $OR(REG, REG) = REG$.

Proof. We have to prove only the inclusion $\subseteq$.

Take an oracle grammar $G = (N, T, S, P, Q)$. For every nonterminal $A \in N_1$, besides the possible symbols $A_?^L, A_t.A_f$ (because the grammar is right–linear, symbol $A_?^R$ are no interest), consider also the symbols $\overline{A}_t, \overline{A}_f$. Construct the right–linear grammar

$$G' = (N, T \cup \{\overline{A}_t, \overline{A}_f \mid A \in N_1\}, S, P'),$$

$$\begin{aligned} P' = \quad & \{A \to x \mid A \to x \in P, x \in (N_1 \cup T)^*\} \cup \\ & \cup \{A \to x\overline{B}_t B_t \mid A \to xB_?^L \in P\} \cup \\ & \cup \{A \to x\overline{B}_f B_f \mid A \to xB_?^L \in P\} \end{aligned}$$

Consider also the languages

$$L_1 = \bigcup_{A \in N_1} (T^* - Q_A)\overline{A}_t T^*,$$

$$L_2 = \bigcup_{A \in N_1} Q_A \overline{A}_f T^*,$$

as well as the morphism $h: (T \cup \{\overline{A}_t, \overline{A}_f \mid A \in N_1\})^* \longrightarrow T^*$ defined by $h(\overline{A}_t) = h(\overline{A}_f) = \lambda, A \in N_1, h(a) = a, a \in T$, and the regular substitution defined by

$$s(a) = \{a\}\{\overline{A}_t, \overline{A}_f \mid A \in N_1\}^*, \quad a \in T.$$

Then we have

$$L(G) = h(L(G') \cap ((T \cup \{\overline{A}_t, \overline{A}_f \mid A \in N_1\})^* - s(L_1 \cup L_2))).$$

Indeed, the strings in L_1 represent cases when A_t is illegally introduced, the strings in L_2 represent cases when A_f is illegally introduced. Taking the complement (after nondeterministically inserting further symbols $\overline{B}_t, \overline{B}_f$) we ensure the correct use of symbols A_t, A_f. The grammar G' follws the derivations in G without involving oracles. By intersection, both the restrictions of rules in G and of oracles are observed. Finally, h erases the auxiliary symbols.

All the operations involved preserve the regularity $L_1, L_2 \in REG$, hence the proof is complete.

Corollary. *The inclusions* $OR(REG, REG) \subset OR(REG, CF)$, $OR(REG, REG) \subset OR(CF, REG)$ *are proper.*

Proof. Obvious, as $OR(REG, CF)$ and $OR(CF, REG)$ strictly include the family CF.

Theorem 5. *The family $OR(CF, REG)$ contains one–letter non–semilinear (hence non–regular) languages.*

Proof. Let us consider the following oracle grammar

$$G = (N, \{a\}, S, P, Q),$$

with

$$N = \{S, A, B, C, D, A', B', B'', C'', A_?^R, A_t, B_?^L, B_t, C_?^L, C_t,$$
$$A_?'^R, A_t', B_?'^R, B_t', B_?''^L, B_t'', C_?''^L, C_t''\},$$

and P containing the subsequent rules

1. $S \to A_?^R a^5 B_?^L D$,
2. $D \to a^5 B_?^L D$,
3. $D \to a^5 B_?^L a^5 C_?^L$,
4. $A_t \to A_?'^R a$,
5. $B_t \to a^2 B_?'^R a$,
6. $C_t \to a^3 C_?^L$,
7. $C_t \to a^3 C_?''^L$,
8. $B_t' \to a B_?^L a$,
9. $B_t' \to a B_?''^L a$,
10. $A_t' \to A_?^R a$,
11. $A_t' \to a^8$,
12. $B'' \to a^5$,
13. $C_t'' \to a^5$.

The set Q contains the following oracle languages:

$$Q_A = \{a^{5k} \mid k \geq 1\},$$
$$Q_B = Q_C = \{a^{5k+1} \mid k \geq 1\},$$
$$Q_{A'} = Q_{B'} = \{a^{5k+4} \mid k \geq 1\},$$
$$Q_{B''} = Q_{C''} = \{a^{5k+2} \mid k \geq 1\}.$$

We obtain

$$L(G) = \{a^{5nm+2} \mid n, m \geq 2\},$$

a language which is not semilinear (by a gsm we can map $L(G)$ into $\{a^{nm} \mid n, m \geq 2\} = \{\ a^p \mid p \text{ composite number}\}$, which is obviously non–semilinear.)

Let us look at the way the grammar G works.

After generating a string $A_?^R(a^5 B_?^L)^i a^5 C_?^L$, $i \geq 2$, using rules 1, 2, 3, the only oracle which is satisfied is Q_A, hence we can pass to $A_t a^6 B_?^L (a^5 B_?^L)^{i-1} a^5 C_?^L$. Now the oracle Q_B of the leftmost $B_?^L$ is satisfied, which makes possible the passing to $A_t a^8 B_t a^6 B_?^L (a^5 B_?^L)^{i-2} a^5 C_?^L$. We can go in this way to the right, from one occurrence of $B_?^L$ to the next one, and this is the only possibility we have. When reaching the right end, the same oracle $Q_B = Q_C$ is satisfied, hence we can pass from $C_?^L$ to $a^3C_?^L$. The obtained string is $A_?'^R a^8 B_?'^R a^8 \ldots a^8 B_?'^R a^9 C_?^L$.

Now the only oracle which allows a continuation is $Q_{B'}$ for the last occurrence of $B_?'^R$. From the right to the left we can replace step by step $B_?'^R$ by B_t', which introduces $aB_?^L a$; when reaching the left end we can replace $A_?'^R$ by A_t' and this one by $A_?^R a$. Thus we obtain $A_?^R a^{10} B_?^L a^{10} \ldots a^{10} C_?^L$.

By iterating this scanning of the string in both directions, in turn, we can produce $A_?^R (a^{5k} B_?^L)^i a^{5k} C_?^L$, $k \geq 1$.

The derivation can be ended only by rules 11, 12, 13, hence after replacing each B and C occurrence by double-primed symbols (rules 7, 8). After introducing $C_?''^L$ we must again go to the left until replacing $A_?'^R$ by A_t' and this one by a^8. The symbol A_t' is introduced having a^{5k+4} in its right hand side; plus a^8 gets $a^{5(k+2)+2}$. This satisfies the next $B_?''^L$ and only this symbol can continue the derivation. After replacing $B_?''^L$ by a^5, the left prefix is again of the form a^{5r+2}, hence we can proced step by step to the right. The derivation can be finished when only symbols $B_?''^L$ are present. Finally, also rule 13 can be used, hence we obtain a string of the form $a^{5(k+2)+2} a^{5(k+2)} \ldots a^{5(k+2)} = a^{5i(k+2)+2}$, for $i \geq 2$, $k \geq 0$.

From the form of the rules and of oracles one can see that only the described strings can be obtained, which concludes the proof.

Corollary. *The inclusion $OR(REG, CF) \subset OR(CF, CF)$ is proper, moreover, $OR(CF, REG) - OR(REG, CF) \neq \emptyset$.*

Proof. If G is an oracle grammar with $T = \{a\}$ and of type (REG, CF), then it is in fact of type (REG, REG), because the one-letter oracle languages are regular. This implies $L(G) \in OR(REG, REG) = REG$. On the other hand, $OR(CF, REG)$, hence also $OR(CF, CF)$, contains one-letter non-regular languages.

Open problems. Are the inclusions $OR(CF, REG) \subseteq OR(CF, CF) \subseteq CS$ proper ? Is the difference $OR(REG, CF) - OR(CF, REG)$ non-empty ?

We *conjecture* that the answer is affirmative in all cases. For instance, we feel that

$$\{a^n b^m a^n \mid m \leq n \leq 2m\} \notin OR(CF, REG),$$

but cleary this language is the intersection of two context-free languages, hence it is in $OR(REG, CF)$.

4. Closure properties. We shall investigate only the family $OR(REG, CF)$, for which surprising results are obtained.

Theorem 6. *The family $OR(REG, CF)$ is closed under union, intersection and complement.*

Proof. The closure under union can be proved in the standard way (make the nonterminal alphabets disjoint and put together all symbols, rules and oracles, respectively)..

Take now $G = (N, T, S, P, Q)$ an oracle grammar with right-linear rules in P and context-free oracle languages. Construct the right-linear grammar $G' = (N, T, S, P')$, with

$$P' = P \cup \{A_?^L \to A_t, A_?^L \to A_f \mid A \in N_1\}.$$

The language $L(G')$ contains all strings which can be generated by G without observing the oracles restriction. Then $T^* - L(G')$ contains strings in the complement of $L(G)$, namely those violating the restrictions imposed by the rules of G. The language $T^* - L(G')$ is regular, hence in $OR(REG, CF)$. We have

$$T^* - L(G) = (T^* - L(G')) \cup L,$$

where $L \subseteq L(G')$ contains the strings generated by G' but violating at least once the oracles. The language L can be generated by the oracle grammar

$$G'' = (N'', T, (S, 0), P'', Q''),$$

where

$$N'' = N \times \{0,1\} \cup \{(A,i)_?^L, (A,i)_t, (A,i)_f \mid A \in N_1, i \in \{0,1\}\},$$

$$\begin{aligned} P'' = \; & \{(A,i) \to x(B,i) \mid A \to xB \in P, i \in \{0,1\}, B \in N_1\} \cup \\ & \cup \{(A,i) \to x(B,i)_?^L \mid A \to xB_?^L \in P, i \in \{0,1\}\} \cup \\ & \cup \{(B,i)_t \to (B_t, i), (B,i)_f \to (B_f, i) \mid i \in \{0,1\}, B \in N_1\} \cup \\ & \cup \{(B,i)_t \to (B_f, j), (B,i)_f \to (B_t, j) \mid (i,j) \in \{(0,1),(1,1)\}, B \in N_1\} \cup \\ & \cup \{(B,1) \to x \mid B \to x \in P, x \in T^*\} \end{aligned}$$

A derivation can be finished if and only if one "crossing rule" $(B,i)_t \to (B_f, j)$, $(B,i)_f \to (B_t, j)$ is used, hence at least one oracle is violated.

As $OR(REG, CF)$ is closed under union, it follows that $T^* - L(G) \in$ $\in OR(REG, CF)$, hence we have the closure under complement.

The closure under intersection follows now from deMorgan laws.

Corollary 1. *The family $OR(REG, CF)$ includes the Boolean closure of the family CF.*

Because every recursively enumerable language is the morphic image of the intersection of two context-free languages, it follows that $OR(REG, CF)$ is not closed under arbitrary morphisms. Moreover, we have

Corollary 2. *The family $OR(REG, CF)$ is not closed under λ-free morphisms (hence it is neither an AFL nor an anti-AFL).*

Proof. Consider the context-free languages

$$L_1 = \{a^n b^n \mid n \geq 1\}^*,$$

$$L_2 = \{a^n b^{m_1} a^{m_1} b^{m_2} a^{m_2} \dots b^{m_{n-1}} a^{m_{n-1}} b^p \mid n, p \geq 1, m_i \geq 1, 1 \leq i \leq n-1\}.$$

We have

$$L_1 \cap L_2 = \{(a^n b^n)^n \mid n \geq 1\}.$$

Take now the morphism $h: \{a, b\}^* \to \{a\}^*$ defined by $h(a) = h(b) = a$. We obtain

$$h(L_1 \cap L_2) = \{a^{2n^2} \mid n \geq 1\},$$

which is not in $OR(REG, CF)$, because this family does not contain non-regular one-letter languages. However, $L_1 \cap L_2 \in OR(REG, CF)$ in view of the previous theorem.

Corollary 3. *All undecidable problems for CF are undecidable for $OR(REG, CF)$ too; moreover, the emptiness and the finiteness problems are undecidable for $OR(REG, CF)$.*

Proof. It is known that the emptiness and the finiteness of the intersection of two context-free languages are undecidable problems.

5. Discussions and further problems

Some interrelations can be observed between our oracle grammars and the conditional grammars in the sense of [3] and [4] (details can be found also in [2], [6]). However, here we check only one terminal subword of the current sentential form. (In the case of right-linear grammars this is in fact practically the whole string, hence Theorem 4 corresponds to the similar results for regular conditional grammars with regular condition languages. Another difference between our grammars and the conditional ones is that here the "condition" concerns about certain nonterminals, not the use of the grammar rules; moreover, we can have here two possibilities to continue the derivation.

The λ - free context-free grammars with regular condition languages (possibly the same for all rules, [4]) characterize the context-sensitive languages. The oracle grammars with context-free rules seems to be weaker, as apparently they cannot simulate context-sensitive derivations. However, results like Theorems 5 and 6 show that the families $OR(CF, REG), OR(REG, CF)$ have properties different from these of basic families obtained by regulated rewriting. For instance, the family of matrix (hence programmed, controlled etc.) languages is not closed under intersection and complement [1], but it is closed under λ -free morphisms and it is conjuctured not to contain one-letter non-regular languages ([2], [6])

In the previous sections we have already pointed out open problems. Many other directions of research remain to be explored.

For instance, we can impose the restriction that all languages coincide (for all $A, B \in N_1, Q_A = Q_B$); in the first example considered in Section 2 this is the case. We can also consider grammars in which either only left-asking symbols or only right-asking symbols are present (the case of the second example in Section 2). Moreover, we can impose the restriction that the questions must be answered immediately after introducing a symbol $A^L_?, A^R_?$ (thus considering that the oracle has priority over rewriting). Another possible modification is to use different answering symbols for left and right questions: X_t^L, X_f^L for $X_?^L$ and X_t^R, X_f^R for $X_?^R$.

All these variants remain to be investigated.

Another natural question is whether or not the number of oracles leads to hierarchies. There are two possible variants: considering the number of oracles in the grammar and considering the number of times they are asked.

Another possibly frutful approach to oracle grammars is to consider complexity. We have to deal with usual derivation steps and recognizing steps (with respect to regular or context-free oracles, hence of known complexity).

REFERENCES

1. F.J. Brandenburg — *A note on matrix grammars.* Manuscript, 1993.
2. J. Dassow, Gh. Păun — *Regulated Rewriting in Formal Language Theory.* Springer-Verlag, 1989.
3. J. Fris — *Grammars with partial ordering of the rules.* Inform. Control, 12 (1968), 415 - 425.
4. J. Kral — *A note on grammars with regular restrictions.* Kybernetika, 9 (1973), 159 - 161.
5. N. Immerman — *Nondeterministic space is closed under complement,* Tech. Rep. Yale Univ., Dept. Computer Sci., 1987.
6. A. Salomaa — *Formal Languages.* Academic Press, 1973.
7. R. Szelepcsenyi — *The method of forcing for nondeterministic automata,* Bulletin EATCS, 33 (1987)

LILA KARI, ALEXANDRU MATEESCU, ARTO SALOMAA
ACADEMY OF FINLAND AND UNIVERSITY OF TURKU
DEPARTMENT OF MATHEMATICS
SF-20500 Turku, FINLAND

GHEORGHE PĂUN
INSTITUTE OF MATHEMATICS
ROMANIAN ACADEMY OF SCIENCES
Str. ACADEMIEI 14, 17109 BUCUREŞTI
ROMÂNIA