On fully initial grammars with regulated rewriting

T. BĂLĂNESCU, M. GHEORGHE, GH. PĂUN

The Computer Centre of the University of Bucharest, Str. Academiei 14, Bucureşti, 70109 ROMANIA

We investigate the fully initial version of context-free grammars added with various control devices: regular control, matrices, programming, random context, Indian parallelism and ordering, each of them with or without λ-rules and (when appropriate) appearance checking. It is shown that the fully initial feature decreases the generative power of programmed, random context λ-free grammars with or without appearance checking, and of ordered and Indian parallel ones. In all remaining cases the generative capacity is not modified. On the other hand, regulated rewriting increases the generative capacity of fully initial context-free grammars.

1. Definitions and notations

The fully initial (fi, for short) variant of context-free grammars was defined by S. Horváth and investigated in [2], [3]. Such a grammar is a usual context-free grammar (cfg, for short) having no distinguished start symbol. The language generated in this way by a grammar $G=(V_N, V_T, P)$ is $L(G)=\{x\in V_T^*|A\overset{*}{\Rightarrow} x$ for some $A\in V_N\}$. (As usual, V_N is the nonterminal vocabulary, V_T is the terminal vocabulary and P is the set of rewriting rules; V^* denotes the free monoid generated by V under the operation of concatenation and λ is the null element.) Inclusion and strict inclusion are denoted by $\subseteq$ and $\subset$, respectively.

Similar to regulated rewriting for context-free grammars [1], [4], we consider here the languages generated by fi regular control, matrix, programmed, random context, Indian parallel and ordered cfg's. We give only informal definitions and refer to [1], [4] for details.

Given a grammar G as above, Lab (P) denotes the set of labels of rules in G (each rule has a distinct label).

A fi regular control (fic, for short) grammar $G=(V_N, V_T, P, K, F)$ consists of a fi cfg (V_N, V_T, P), a regular control language K over Lab (P) and a set F of labels. We write $A \overset{*}{\Rightarrow} y$ in G if there exists a string $p_1 p_2 \dots p_n \in K$, $p_i \in \text{Lab}(P)$, such that $A=x_0 \underset{p_1}{\Rightarrow} x_1 \dots \underset{p_n}{\Rightarrow} x_n=y$, and for each i we have either $x_{i-1} \underset{p_i}{\Rightarrow} x_i$ or $x_{i-1}=x_i$, the rule p_i is not applicable to x_{i-1} and $p_i \in F$.

A fi matrix (fim, for short) grammar $G=(V_N, V_T, P, M, F)$ consists of a cfg (V_N, V_T, P), a finite set M of matrices and a finite set F of occurrences of productions in matrices of M. A matrix is a sequence $m=(A_1 \to u_1, \dots, A_n \to u_n)$, $n \geqq 1$, of productions in P. We write $x \underset{m}{\Rightarrow} y$ for a matrix m as above if there are $x_1=x$, $x_2, \dots$ $\dots, x_n=y$ such that either $x_j=x_{j+1}$, the rule r_j: $A_j \to u_j$ is in F and it is not applicable to x_j or $x_j \underset{r_j}{\Rightarrow} x_{j+1}$.

In a programmed (fip, for short) grammar $G=(V_N, V_T, P)$ the rules are of the form $(b: A \to u, S(b), F(b))$, where b is the label of the production, $S(b)$ and $F(b)$ are sets of labels referred to as the success and the failure field. If $A \to u$ is applicable to a string x, then, after applying it, we continue the derivation with a rule having the label in $S(b)$; if $A \to u$ is not applicable to x, then we pass to a rule with its label in $F(b)$ (the string x remains unchanged).

A fi random context (firc, for short) grammar $G=(V_N, V_T, P)$ has the rules of the form $(A \to u, Q, R)$, where Q, R are subsets of V_N, referred to as permitting and forbidding sets of symbols, respectively. Such a rule is applicable to a string x iff x contains all nonterminals of Q and contains no nonterminal in R.

A fi Indian parallel (fiip, for short) grammar is a cfg grammar in which each rule $A \to w$ is used in a derivation $u \Rightarrow v$ for rewriting all occurrences of A in w, thus obtaining v.

A fi ordered (fio, for short) grammar $(G, >)$ consists of a fi cfg G and a partial order $>$ on P. A rule $A \to u$ is applicable to a string x iff no rule $B \to v$ is applicable to x and $B \to v > A \to u$.

We denote by FI_λ, $\text{FIC}_{ac,\lambda}$, $\text{FIM}_{ac,\lambda}$, $\text{FIP}_{ac,\lambda}$, $\text{FIRC}_{ac,\lambda}$, FIIP_λ, and FIO_λ the families of languages generated by fi, fic, fim, fip, firc, fiip and fio grammars, respectively. The corresponding families generated in the usual mode are denoted by $\text{C}_{ac,\lambda}$, $\text{M}_{ac,\lambda}$, $\text{P}_{ac,\lambda}$, $\text{RC}_{ac,\lambda}$, IP_λ, O_λ, respectively. When the appearance checking feature is not present, that is when $F=\emptyset$ for fic and fim, $F(b)=\emptyset$ for fip and $R=\emptyset$ for firc grammars, we erase the subscript ac; when no λ-rules are allowed we erase also the subscript λ. As usual, the families of recursively enumerable, context sensitive, context-free and regular languages are denoted by $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$, respectively.

Two languages are identified if they differ by at most the empty string.

2. The generative capacity of fully initial regulated grammars

Lemma 1. $\text{FIC}_{ac,\lambda}=\text{C}_{ac,\lambda}$, $\text{FIC}_\lambda = \text{C}_\lambda$, $\text{FIC}_{ac}=\text{C}_{ac}$, $\text{FIC}=\text{C}$.

Proof. Let $G=(V_N, V_T, S, P, K, F)$ be a regular control grammar. We consider the fic grammar $G'=(V_N, V_T, P, K', F)$, where $K'=K \cap I \cdot \text{Lab}(P)^*$, I being the set of labels of rules of the form $S \to u$ in P. Clearly, $L(G)=L(G')$ and G' is of the same type as G.

Conversely, for a fic grammar $G=(V_N, V_T, P, K, F)$ we consider the regular control grammar $G'=(V_N\cup\{S\}, V_T, S, P', K', F)$, where S is a new nonterminal, $P'=P\cup\{S\to A|A\in V_N\}$, $K'=I\cdot K$ and I is the set of labels of rules $S\to A$, $A\in V_N$. Obviously, $L(G)=L(G')$.

Lemma 2. $\mathrm{FIM}_{ac,\lambda}=\mathrm{M}_{ac,\lambda}$, $\mathrm{FIM}_{\lambda}=\mathrm{M}_{\lambda}$, $\mathrm{FIM}_{ac}=\mathrm{M}_{ac}$, $\mathrm{FIM}=\mathrm{M}$.

Proof. Let $G=(V_N, V_T, P, M, F)$ be a fim grammar. We construct the grammar $G'=(V_N\cup\{S\}, V_T, S, P, M', F)$, where S is a new symbol and $M'=M\cup\{(S\to A)| A\in V_N\}$. Clearly, $L(G)=L(G')$, hence we have the inclusions $\subseteq$.

Conversely, let $L\subseteq V^*$ be a matrix language in a family $\mathrm{M}_{\alpha,\beta}$, $\alpha=ac$ or it is empty, $\beta=\lambda$ or it is empty. We write

$$L = \bigcup_{a\in V} \{a\}\partial_a(L)\cup\{x\in L|\,|x|\leqq 1\}$$

($\partial_a(L)$ is the left derivative of L with respect to a). Each language $\partial_a(L)$ is a matrix language of the same type as L; let $G_a=(V_{N,a}, V, S_a, P_a, M_a, F_a)$ be a matrix grammar for each. Without loss of generality we may suppose that the vocabularies $V_{N,a}$ are pairwise disjoint and that each M_a contains matrices $m=(r_1, ..., r_n)$ with at least one occurrence of productions not in F_a (otherwise we remove m and the corresponding occurrences of rules from M_a and F_a, respectively, and we introduce all matrices $m_i=(r_1, ..., r_n)$, $1\leqq i\leqq n$, containing the same rules as m but with the rule occurring on the position i not in F_a).

A fim grammar generating L is $G'=(V_N', V, P', M', F')$, where

$$V_N' = \bigcup_{a\in V} (V_{N,a}\cup\{[\alpha]\})\cup\{S\}, \quad S \text{ is a new symbol},$$

$$P' = \bigcup_{a\in V} (P_a\cup\{S\to[a]S_a, [a]\to[a], [a]\to a\})\cup\{S\to x|x\in L,\ |x|\leqq 1\},$$

and M' is constructed as follows:

a) $(S\to x)$, $x\in L$, $|x|\leqq 1$, is in M',

b) for each $a\in V$ we introduce in M' the matrices

b.1) $(S\to [a]S_a)$,

b.2) $([a]\to[a],\ r_1, ..., r_n)$, for $(r_1, ..., r_n)\in M_a$,

b.3) $([a]\to a,\ r_1, ..., r_n)$, for $(r_1, ..., r_n)\in M_a$.

Finally, $F'=\bigcup_{a\in V} F_a$.

It is easy to see that in each derivation of a string $x\in L$, $|x|>1$, all sentential forms are of the form $[a]w$; moreover, no derivation can start from a symbol different from S (remember that for all $a\in V$, each matrix in M_a contains a rule not in F_a). In conclusion, $L(G')=L$, hence $\mathrm{M}_{\alpha,\beta}\subseteq\mathrm{FIM}_{\alpha,\beta}$, α, β as above.

Lemma 3. $\mathrm{FIP}_{ac,\lambda}=\mathrm{P}_{ac,\lambda}$, $\mathrm{FIP}_{\lambda}=\mathrm{P}_{\lambda}$, $\mathrm{FIP}_{ac}\subseteq\mathrm{P}_{ac}$, $\mathrm{FIP}\subseteq\mathrm{P}$.

Proof. Let $G=(V_N, V_T, P)$ be a fip grammar and consider the programmed grammar $G'=(V_N\cup\{S\}, V_T, S, P')$, where S is a new symbol and $P'=P\cup\{(r_A\colon S\to A,\ \mathrm{Lab}(P), \emptyset)|A\in V_N\}$. We have $L(G)=L(G')$, hence $\mathrm{FIP}_{\alpha,\beta}\subseteq\mathrm{P}_{\alpha,\beta}$, $\alpha=ac$ or it is empty, $\beta=\lambda$ or it is empty.

Conversely, let $G=(V_N, V_T, S, P)$ be a programmed grammar. We construct the fip grammar $G'=(V'_N, V_T, P)$, where
$V'_N=V_N\cup\{X, Y, N\}$, X, Y, N are new symbols, and P' contains the next rules:

a) $(s\colon X\to SY,\ S(s),\ \emptyset)$, $s\notin \mathrm{Lab}(P)$, $S(s)=\{i|(i\colon S\to u,\ S(i),\ F(i))\in P\}$,

b) $(r\colon A\to uN,\ S(r)\cup\{f\},\ F(r))$, $f\notin \mathrm{Lab}(P)$ and

$(r\colon A\to u,\ S(r),\ F(r))\in P$,

c) $(f\colon Y\to\lambda, \{f_N\}, \emptyset)$,

d) $(f_N\colon N\to\lambda, \{f_N\}, \emptyset)$, $f_N\notin \mathrm{Lab}(P)$.

It is easy to see that the symbol N cannot be erased without erasing first symbol Y. Therefore, no rule in group b) can be successfully used without starting the derivation by the rule of type a). In consequence, $L(G)=L(G')$, hence $P_{\alpha,\lambda}\subseteq \mathrm{FIP}_{\alpha,\lambda}$, where α is as above.

Lemma 4. $\mathrm{FIRC}_{ac,\lambda}=\mathrm{RC}_{ac,\lambda}$, $\mathrm{FIRC}_{\lambda}=\mathrm{RC}_{\lambda}$, $\mathrm{FIRC}_{ac}\subseteq R_{ac}$, $\mathrm{FIRC}\subseteq\mathrm{RC}$.

Proof. Given a firc grammar $G=(V_N, V_T, P)$, we construct the random context grammar $G'=(V_N\cup\{S\}, V_T, S, P')$, where S is a new symbol and $P'=P\cup\{(S\to A, \emptyset, \emptyset)|A\in V_N\}$. We have $L(G)=L(G')$, hence $\mathrm{FIRC}_{\alpha,\beta}\subseteq\mathrm{RC}_{\alpha,\beta}$, $\alpha=ac$ or it is empty, $\beta=\lambda$ or it is empty.

Conversely, for a random context grammar $G=(V_N, V_T, S, P)$, we construct the firc grammar $G'=(V_N\cup\{X, Y\}, V_T, P')$, where X, Y are new symbols and P' contains the following rules:

a) $(X\to SY, \emptyset, \emptyset)$,

b) $(Y\to\lambda, \emptyset, \emptyset)$,

c) $(A\to u,\ Q\cup\{Y\},\ R)$, for $(A\to u, Q, R)\in P$.

Obviously, $L(G)=L(G')$, which completes the proof.

Lemma 5. $\mathscr{L}_3 - \mathrm{FIO}_\lambda\neq\emptyset$.

Proof. Let us consider the regular language $L=\{ab^na|n\geqq 0\}$ and suppose that L is generated by the fio grammar $(G, >)$, $G=(V_N, \{a, b\}, P)$. Define $k=\max\{|u|\ |A\to u\in P\}$ and consider a derivation $A=u_0\Rightarrow u_1\Rightarrow\ldots\Rightarrow u_p=ab^ka$ in $(G, >)$, $A\in V_N$. As $|ab^ka|>k$, we have $p\geqq 2$. Let i be the greatest index such that $u_i=u'_iBu''_i$ and $u'_i\overset{*}{\Rightarrow}\lambda$, $u''_i\overset{*}{\Rightarrow}\lambda$ and $B\overset{*}{\Rightarrow}ab^ka$ in $(G, >)$. It follows that $B\Rightarrow uCv\overset{*}{\Rightarrow}ab^ka$, $ab^ka=xyz$, $u\overset{*}{\Rightarrow}x$, $C\overset{*}{\Rightarrow}y$, $v\overset{*}{\Rightarrow}z$ and $y\neq\lambda$. Clearly, $y\neq ab^ka$, hence $y\in L(G, >)$ and y is a proper subword of ab^ka, contradiction.

Corollary. $\mathrm{FIO}_\lambda\subset\mathrm{O}_\lambda$, and $\mathrm{FIO}\subset\mathrm{O}$.

Lemma 6. $\mathscr{L}_3-(\mathrm{FIP}_{ac}\cup\mathrm{FIRC}_{ac})\neq\emptyset$.

Proof. Let us consider the language $L=\{ab^na|n\geqq 0\}$ as above and suppose that L is generated by a fip (firc) grammar $G=(V_N, V_T, P)$ without λ-rules. Let $k=\max\{|u|\,|A\to u\in P\}$ and take $x=ab^ka\in L(G)$. There exists a derivation $A\Rightarrow x_1\Rightarrow\ldots$

$\ldots \Rightarrow x_n = ab^k a$, $A \in V_N$. The lastly used rule is $B \to u$, with $u = ab^t$, or $u = b^q a$, or $u = b^s$, $0 \leqq t$, $q < k$, $1 \leqq s \leqq k$. It follows that $u \in L(G)$, a contradiction.

Corollary. (i) $\mathscr{L}_2 - (\mathrm{FIP}_{ac} \cup \mathrm{FIRC}_{ac}) \neq \emptyset$, (ii) $\mathrm{FIP}_{ac} \subset \mathrm{P}_{ac}$, $\mathrm{FIP} \subset \mathrm{P}$, $\mathrm{FIRC}_{ac} \subset \mathrm{RC}_{ac}$, $\mathrm{FIRC} \subset \mathrm{RC}$.

Lemma 7. Let L be a language over a vocabulary V and let c be a symbol not in V. a) If $L \in \mathrm{P}_{ac}$ ($L \in \mathrm{P}$), then $L\{c\} \cup V \cup \{c\} \in \mathrm{FIP}_{ac}$ (FIP, respectively). b) If $L \in \mathrm{RC}_{ac}$ ($L \in \mathrm{RC}$), then $L\{c\} \cup \{c\} \in \mathrm{FIRC}_{ac}$ (FIRC, respectively).

Proof. a) For a programmed λ-free grammar $G = (V_N, V, S, P)$ generating L, we construct the fip grammar $G' = (V_N', V \cup \{c\}, P')$ with $V_N' = V_N \cup \{a' \mid a \in V\} \cup \{X, Y\}$ where X, Y are new symbols, and with P' containing the next productions:

a) $(s\colon X \to SY, S(s), \emptyset)$, with $s \notin \mathrm{Lab}(P)$, $S(s) = \{i \mid (i\colon S \to u, S(i), F(i)) \in P\}$

b) $(r\colon A \to u', S(r) \cup \{f\}, F(r))$, for each $(r\colon A \to u, S(r), F(r)) \in P$; $f \notin \mathrm{Lab}(P)$ and u' is obtained from u by replacing each $a \in V$ by $a' \in V_N'$ in u,

c) $(f\colon Y \to c, \{f_a \mid a \in V\}, \emptyset)$,

d) $(f_a\colon a' \to a, \{f_b \mid b \in V\}, \emptyset)$, for all $a \in V$; $f_a \notin \mathrm{Lab}(P)$.

The equality $L(G') = L\{c\} \cup V \cup \{c\}$ is obvious, hence we have proved the first part of the lemma.

b) If $G = (V_N, V, S, P)$ is a random context grammar generating L, then we construct the firc grammar $G' = (V_N', V \cup \{c\}, P')$, where

$$V_N' = V_N \cup \{X, Y\}, \quad \text{with new symbols } X \text{ and } Y,$$

$$P' = \{(X \to SY, \emptyset, \emptyset), (Y \to c, \emptyset, \emptyset)\} \cup$$

$$\cup \{(A \to u, Q \cup \{Y\}, R) \mid (A \to u, Q, R) \in P\}.$$

We obviously have $L(G') = L\{c\} \cup \{c\}$, which completes the proof.

Corollary 1. $\mathrm{FIP} - \mathscr{L}_2 \neq \emptyset$, $\mathrm{FIRC} - \mathscr{L}_2 \neq \emptyset$.

Proof. Follows from $\mathrm{P} - \mathscr{L}_2 \neq \emptyset$, $\mathrm{RC} - \mathscr{L}_2 \neq \emptyset$, the above lemma and the closure properties of $\mathscr{L}_2$.

Corollary 2. $\mathrm{FIRC} - \mathrm{FIP}_{ac} \neq \emptyset$.

Proof. The language $L = \{ab^n a \mid n \geqq 0\}\{c\} \cup \{c\}$ is in FIRC, but not in FIP_{ac} (this follows as in the proof of Lemma 6).

Lemma 8. $\mathrm{FIP} - \mathrm{FIO}_\lambda \neq \emptyset$.

Proof. The language $L = \{ab^n ac \mid n \geqq 0\} \cup \{a, b, c\} \in \mathrm{FIP} - \mathrm{FIO}_\lambda$. The relation $L \in \mathrm{FIP}$ follows from Lemma 7, and $L \notin \mathrm{FIO}_\lambda$ can be proved as in the proof of Lemma 5.

Corollary. $\mathrm{FIRC} - \mathrm{FIO}_\lambda \neq \emptyset$, $\mathrm{FIP}_{ac} - \mathrm{FIO} \neq \emptyset$.

Lemma 9. $\mathrm{FIO} \subset \mathrm{FIP}_{ac}$.

Proof. Let $(G, >)$, $G = (V_N, V_T, P)$, be a fio grammar. Without loss of generality we may assume that whenever $A \to u$ and $A \to v$ are both in P, then these rules are incomparable. We construct the fip grammar $G' = (V_N \cup \{X\}, V_T, P')$, where X is a new symbol and P' is constructed as follows. For any rule $r\colon A \to u \in P$ write $g(r) = \{A_1 \to u_1, \ldots, A_n \to u_n\}$, where $A_i \to u_i > A \to u$, $1 \leqq i \leqq n$.

For every rule r: $A \to u$ in P, introduce in P' all the rules $(r^{(i)}: A_i \to u_i X, \emptyset, \{r^{(i+1)}\})$, $1 \leqq i \leqq n-1$, as well as the rule $(r^{(n)}: A_n \to u_n X, \emptyset, \{r'\})$; then, add also to P' the rule $(r': A \to u, E, \emptyset)$, with $E = \{p^{(1)} | p: B \to v \in P, \ g(p) \neq \emptyset\} \cup \{p' | p: B \to v \in P, \ g(p) = \emptyset\}$.

A derivation in G' develops as follows: the use of a rule $(r': A \to u, E, \emptyset)$ is preceeded by the application with appearance checking of all the rules $r^{(i)}$, $1 \leqq i \leqq \text{card}(g(r))$; if such a rule $r^{(i)}$ can be applied, then the derivation is blocked. Therefore $L(G, >) = L(G')$, hence $\text{FIO} \subseteq \text{FIP}_{ac}$. The proper inclusion follows from the corollary to Lemma 8.

Lemma 10. $\text{FIP}_{ac} \subset \text{FIRC}_{ac}$.

Proof. Let $G = (V_N, V_T, P)$ be a fip grammar. We construct the firc grammar $G' = (V'_N, V_T, P')$, where

$$V'_N = \{[A, r] | A \in V_N, \ r \in \text{Lab}(P)\} \cup \{(u, r) | (r: A \to u, S(r), F(r)) \in P\}$$

and, for every rule $(r: A \to u, S(r), F(r)) \in P$, the set P' contains the following random context rules:

a) $([A, r] \to (u, r), \emptyset, C_r)$, for any $s \in S(r)$,

b) $([B, r] \to [B, s], \{(u, s)\}, C_{r,s} - \{(u, s)\})$, for any $s \in S(r)$ and $B \in V_N$,

c) $((u, s) \to [u, s], \emptyset, C_s - \{(u, s)\})$, for any $s \in S(r)$,

d) $([B, r] \to [B, f], \emptyset, C_{r,f} \cup \{[A, r]\})$, for any $f \in F(r)$, $B \in V_N$ and $B \neq A$,

with $C_r = V'_N - \{[X, r] | X \in V_N\}$, $C_{r,s} = C_r - \{[X, s] | X \in V_N\}$ and if $u = x_1 A_1 x_2 \dots x_n A_n x_{n+1}$, $x_i \in V_T^*$, $A_i \in V_N$, $n \geqq 0$, then $[u, s] = x_1 [A_1, s] \dots x_n [A_n, s] x_{n+1}$.

An arbitrary derivation $v \underset{r}{\Rightarrow} w$ in G is simulated in G' as follows. If r is not applicable to v, then simply apply the rules of the form d) and continue according to the failure field $F(r)$. Otherwise, a rule of the form a) is applied, provided all nonterminals are marked with the label r. The new by introduced nonterminal, (u, s), enables us to continue the derivation according to the success field $S(r)$; it assists the application of the rules of the form b) until all nonterminals are marked by s. Next, the rewriting of A by u is simply accomplished by a rule of the form c); note that all nonterminals of the sentential form must be marked by s. The process continues with the rules derived from the rule $s \in S(r)$. Obviously, $L(G) \subseteq L(G')$. Similarly, each derivation in G' corresponds to one in G, hence $L(G') \subseteq L(G)$, hence $\text{FIP}_{ac} \subseteq \text{FIRC}_{ac}$. The inclusion is proper, as it follows from Corollary 2 of Lemma 7.

Let us investigate now the Indian parallel grammars.

Similarly to the equality $\text{IP} = \text{IP}_\lambda$, we also have $\text{FIIP} = \text{FIIP}_\lambda$.

Lemma 11. $\text{FIIP} \subset \text{IP}$ and $\text{FIIP} \subset \text{FIP}_{ac}$.

Proof. If $G = (V_N, V_T, P)$ is a fiip grammar, we construct $G' = (V_N \cup \{S\}, V_T, S, P')$, S a new symbol, $P' = P \cup \{S \to A | A \in V_N\}$, for proving $\text{FIIP} \subseteq \text{IP}$, and

$$G'' = (V_N \cup \{A' | A \in V_N\}, V_T, P''),$$

$$P'' = \{(r: A \to x_A, \{r\}, \{r_A\}) | r: A \to x \in P\} \cup \{(r_A: A' \to A, \{r_A\}, \text{Lab}(P)) | A \in V_N\},$$

for proving $\mathrm{FIIP} \subseteq \mathrm{FIP}_{ac}$ (x_A is the string obtained by replacing each occurrence of A in x by A').

As $\{ab^n a \mid n \geq 0\} \in \mathrm{IP} - \mathrm{FIP}_{ac}$ and the Dyck language over $\{a, b\}$ is in FI (it is generated by the cfg $(\{S\}, \{a, b\}, \{S \to SS, S \to \lambda, S \to aSb\})$), but not in IP, both inclusions above are proper.

Corollary 1. $\mathscr{L} - \mathrm{FIIP} \neq \emptyset$ for all families $\mathscr{L} \in \{\mathrm{FI}, \mathscr{L}_2, \mathrm{FIO}\}$.

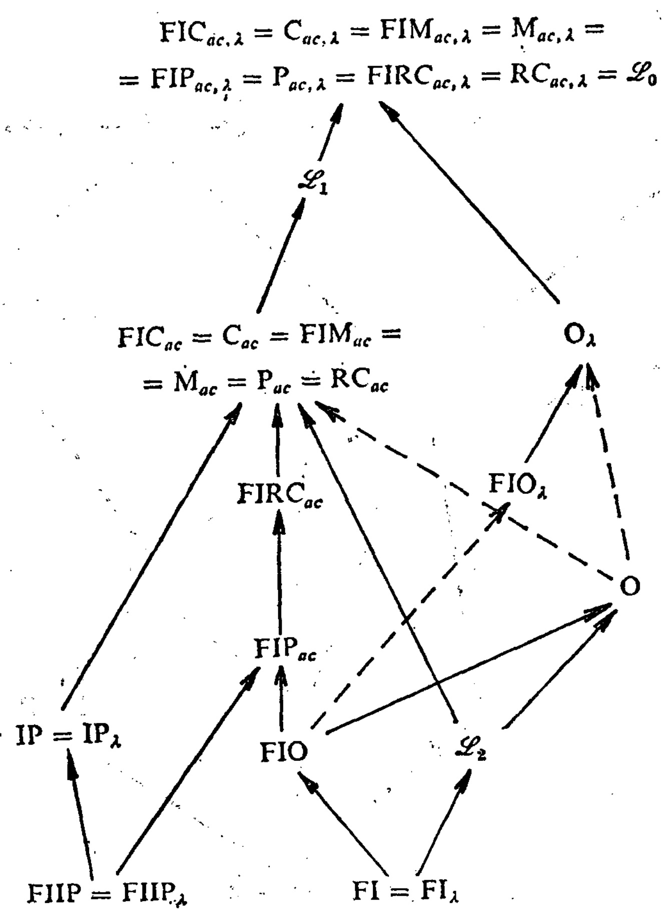
Corollary 2. IP is incomparable with all families FI, $\mathscr{L}_2$, FIO, FIP, FIP_{ac}, FIRC, FIRC_{ac}.

Lemma 12. $\mathrm{FIIP} - \mathscr{L}_2 \neq \emptyset$, $\mathrm{FIIP} - \mathrm{FI} \neq \emptyset$.

Proof. $L = \{a^{2^n} \mid n \geq 0\}$ is in FIIP as it is generated by the grammar $(\{S\}, \{a\}, \{S \to SS, S \to a\})$, but it is not context-free, hence nor is it in FI.

Summarizing the results in the previous lemmas, we obtain:

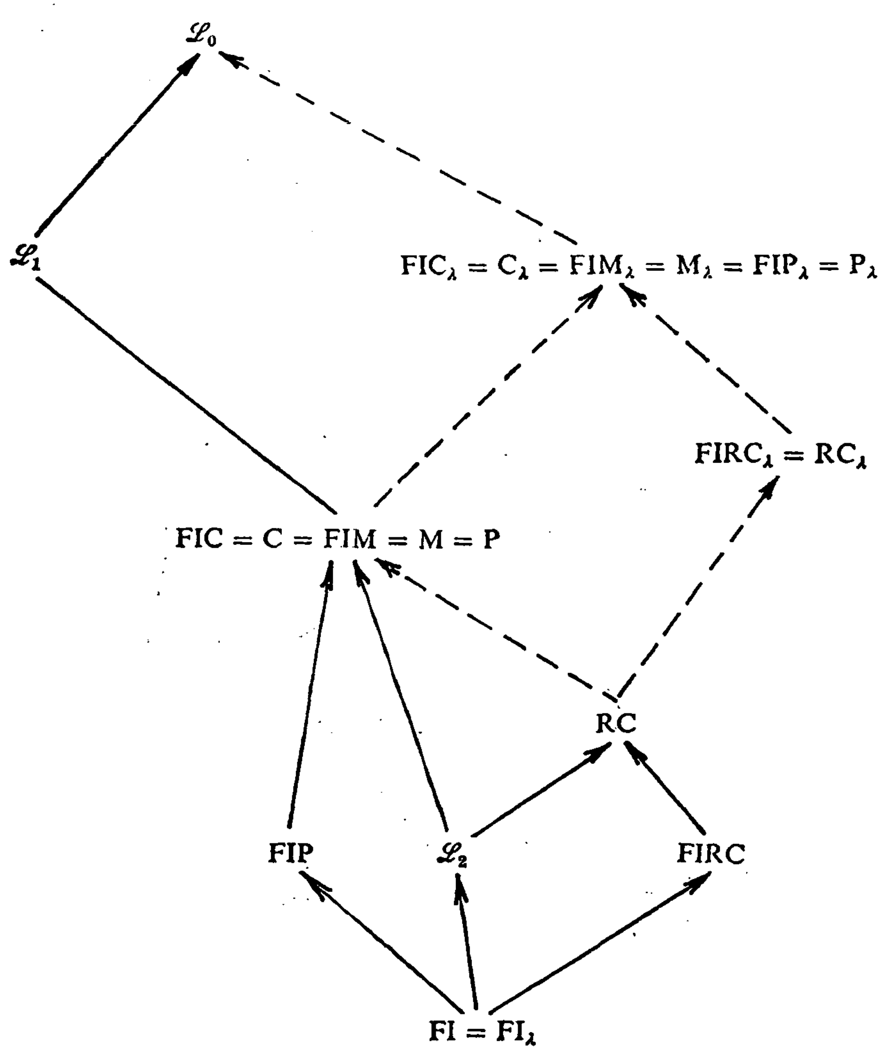
Theorem 1. The following diagram holds:



where ——→ indicates strict inclusion and - - → points out an inclusion which is not known to be strict.

Theorem 2. a) The families in the next pairs are incomparable: $(\mathscr{L}_2, \mathrm{FIO})$, $(\mathscr{L}_2, \mathrm{FIO}_\lambda)$, $(\mathscr{L}_2, \mathrm{FIP}_{ac})$, $(\mathscr{L}_2, \mathrm{FIRC}_{ac})$, $(\mathscr{L}_2, \mathrm{FIIP})$, (FI, FIIP), (IP, FI), (IP, FIO), (IP, FIP), $(\mathrm{IP}, \mathrm{FIP}_{ac})$, (IP, FIRC), $(\mathrm{IP}, \mathrm{FIRC}_{ac})$, $(\mathrm{IP}, \mathscr{L}_2)$. b) The following relations hold: $\mathrm{FIP}_{ac}-\mathrm{FIO}_\lambda \neq \emptyset$, $\mathrm{FIRC}_{ac}-\mathrm{FIO}_\lambda \neq \emptyset$, $\mathscr{L}_3-\mathrm{FIRC}_{ac} \neq \emptyset$, $\mathscr{L}_3-\mathrm{FIP}_{ac} \neq \emptyset$, $\mathscr{L}_3-\mathrm{FIO}_\lambda \neq \emptyset$, $\mathrm{FIO}-\mathrm{FIIP} \neq \emptyset$.

Theorem 3. The following diagram holds



Theorem 4. a) The families in the next pairs are incomparable: $(\mathscr{L}_2, \mathrm{FIP})$ $(\mathscr{L}_2, \mathrm{FIRC})$. b) The following relations hold: $\mathrm{FIRC}-\mathrm{FIP}_{ac}\neq\emptyset$, $\mathrm{FIP}-\mathrm{FIO}_\lambda\neq\emptyset$ $\mathrm{FIRC}-\mathrm{FIO}_\lambda\neq\emptyset$.

3. Final remarks and open problems

As it may be noticed from the previous results, any recursively enumerable set can be generated by fully initial context-free grammars with the following regulated rewriting: matrices, programming, regular control and random context, provided that the appearance checking mode of derivation is present. If λ-rules are not allowed, then the fully initial regular control and matrix grammars are weaker than the context sensitive grammars and they are stronger than the context-free ones. Moreover, the fully initial context-free ordered, programmed, random context and matrix λ-free grammars give a hierarchy of languages (appearance checking is supposed). The family of context-free languages strictly includes the fully initial corresponding family, but it is strictly contained in the family of fully initial regular control and matrix languages. Both the families of regular and context-free languages are incomparable with the families of fully initial ordered and of Indian parallel languages, as well as, with the families of fully initial λ-free programmed and random context languages. The incomparability of the fully initial ordered family (with λ-rules) with the fully initial random context and programmed families is only partially solved: we said nothing about $\mathrm{FIO}_\lambda-\mathrm{FIP}_{ac}$ and $\mathrm{FIO}_\lambda-\mathrm{FIRC}_{ac}$. Without appearance checking but with λ-rules, it seems that the fully initial random context grammars are weaker than the regular control, the matrix and the programmed grammars. Moreover, in the λ-free case, the fully initial programmed and random context grammars are stronger than the fully initial context-free grammars, but the relation between them remains open (we know only that $\mathrm{FIRC}-\mathrm{FIP}\neq\emptyset$). As these open problems correspond to some unsettled questions about usual regulated grammars, the answers are not expected to be easy.

Similarly to the usual case, the Indian parallel family has a "lateral" position (incomparable with FI, $\mathscr{L}_2$ etc.).

References

[1] J. Dassow, Gh. Păun, *Regulated rewriting in formal language theory*, Akademie Verlag, Berlin, 1989.

[2] J. Dassow, On fully initial context-free languages, *Papers on automata and languages* (Ed. I. Peák), X (1988), 3—6.

[3] Gh. Păun, A note on fully initial context-free languages, *Papers on automata and languages* (Ed. I. Peák), X (1988), 7—11.

[4] A. Salomaa, *Formal languages*, Academic Press, New York, London, 1973.

(Received June 21, 1988)