ZESZYTY NAUKOWE UNIWERSYTETU JAGIELLOŃSKIEGO
DLXXX PRACE MATEMATYCZNE, ZESZYT 22 1981

On F-languages and their grammars

by Maria Foryś

As a result of the generalization of a finite automaton the theory of tree languages was created. The first papers dealt with the problems connected with the notion of tree automaton. The question of generation tree languages was worked out later, viz. in 1969 by W. S. Brainerd [1].

In this paper the definitions of context-free F-grammar and finite state F-grammar are formulated and families of F-languages generated by those grammars are characterized. So it is a proposition of the generalization problem considered by W. S. Brainerd to F-languages, i.e. languages composed of finite sequences of trees.

We recall all the definitions and the terminology from [4]. Furthermore, we introduce some new notions concerning F-languages.

Definition 1. Let $l \in \Sigma^F$. *A frontier* of a forest l $(\mathrm{fr}(l))$ is a word over Σ defined as follows

$$\mathrm{fr}(e) = \mathrm{fr}(\bar{e}) = \lambda,$$

where λ is the unit of Σ^*,

$$\mathrm{fr}(l) = \sigma$$

for $l = \underset{\downarrow}{\sigma}$ or $l = \underset{\cdot}{\sigma}$,

$$\mathrm{fr}(l) = \begin{cases} \mathrm{fr}(l_2)\mathrm{fr}(l_3) & \text{if } l_2 \neq e,\ l_3 \neq \bar{e}; \\ \mathrm{fr}(l_1)\mathrm{fr}(l_3) & \text{otherwise} \end{cases}$$

for any $l = (X, \Gamma, v)$ such that $|X| > 1$ and $l = [l_1; l_2; l_3]$.

By a frontier of any F-language we mean the set of frontiers of all forest from this F-language.

Now we formulate the definition of F-grammar, the basic notion of our consideration.

Definition 2. Let S and Σ are any disjoint alphabets. *A F-grammar* is a system $FG = \langle S_1^F, \Sigma_1^F, P^F, \underset{\downarrow}{s_0} \rangle$ where

$$S_1^F = \{\underset{\downarrow}{s} \in S^F\colon\ s \in S\}$$

is a set of nonterminals,

$$\Sigma_1^F = \{\underset{\downarrow}{\sigma} \in \Sigma^F\colon\ \sigma \in \Sigma\}$$

is a set of terminals,

$$P^F \subset S_1^F \times (S \cup \Sigma)^F$$

is finite relation named a set of productions and

$$\underset{\downarrow}{s_0} \in S_1^F$$

is distinguished one-node forest, named an initial forest.

The fact that $(\underset{\downarrow}{s}, l) \in P^F$ we will write as $\underset{\downarrow}{s} \Rightarrow l$. The process of generation in F-grammar is now defined.

Definition 3. Let $l, \bar{l} \in (S \cup \Sigma)^F$, $l = (X, \Gamma, v)$, $\bar{l} = (\bar{X}, \bar{\Gamma}, \bar{v})$ and $l = [l_1; l_2; l_3]$. A forest l *generates directly* a forest $\bar{l}$ in F-grammar $FG = \langle S_1^F, \Sigma_1^F, P^F, \underset{\downarrow}{s_0} \rangle$ (symbolically $l \underset{FG}{\Rightarrow} \bar{l}$) if and only if there are exists production $\underset{\downarrow}{s} \Rightarrow l_0 \in P^F$ and $x \in X$ such that $v(x) = s$ and for a forest $\bar{l}$ one of the following conditions holds:

1° if $l_1 = \underset{\downarrow}{s}$ then

$$\bar{l} = l_0 \downarrow l_2 \rightarrow l_3 ,$$

2° if $l_1 \neq \underset{\downarrow}{s}$ then we construct the forest $\bar{l}$ in two steps:

step I: we define a sequence of forests

$$l_{k_1}, l_{k_1 k_2}, \ldots, l_{k_1 k_2 \ldots k_m} ,$$

where $l_{k_1 \ldots k_i} = (X_{k_1 \ldots k_i}, \Gamma_{k_1 \ldots k_i}, v_{k_1 \ldots k_i})$ for $k_i \in \{2, 3\}$ and $i = 1, \ldots, m$ putting

(i) $l_{k_1 \ldots k_i} = [l_{k_1 \ldots k_i 1}; l_{k_1 \ldots k_i 2}; l_{k_1 \ldots k_i 3}]$,

(ii) $l_{k_1} \in \{l_2, l_3\}$ and for $i = 1, \ldots, m$

$$l_{k_1 \ldots k_i} \in \{l_{k_1 \ldots k_{i-1} 2}, l_{k_1 \ldots k_{i-1} 3}\}$$

(iii) $x \in X_{k_1 \ldots k_i}$ for $i = 1, \ldots, m$

(iv) m is a non-negative integer such that $l_{k_1 \ldots k_m} = \underset{\downarrow}{s}$.

step II: using the notation

$$\boldsymbol{k}_i = \begin{cases} 2 & \text{if } k_i = 3 , \\ 3 & \text{if } k_i = 2 \end{cases}$$

we define the following forests

$$l_{k_1 \ldots k_m} = l_0 \downarrow l_{k_1 \ldots k_m 2} \rightarrow l_{k_1 \ldots k_m 3}, \qquad l_{k_1 \ldots k_{i-1} \boldsymbol{k}_i} = l_{k_1 \ldots k_{i-1} \boldsymbol{k}_i} \quad \text{for } i = 1, \ldots, m$$

$$l_{k_1 \ldots k_i 1} = l_{k_1 \ldots k_i 1} \quad \text{for } i = 1, \ldots, m-1$$

and

$$l_{k_1 \ldots k_i} = l_{k_1 \ldots k_i 1} \downarrow l_{k_1 \ldots k_i 2} \rightarrow l_{k_1 \ldots k_i 3} \quad \text{for } i = 1, \ldots, m-1 .$$

Finally according to forests introduced in steps I and II the forest $\bar{l}$ is defined by the equality

$$\bar{l} = l_1 \downarrow l_2 \rightarrow l_3$$

Definition 4. Let $l, l \in (S \cup \Sigma)^F$. A forest l *generates* a forest l in F-grammar $FG = \langle S_1^F, \Sigma_1^F, P^F, \underset{\downarrow}{s_0} \rangle$ (symbolically $l \overset{*}{\underset{FG}{\Rightarrow}} l$) if and only if there exist forests

$$l_1, \ldots, l_n \in (S \cup \Sigma)^F$$

such that

$$l_1 = l, \; l_n = l$$

and for $i = 1, \ldots, n-1$

$$l_i \underset{FG}{\Rightarrow} l_{i+1} .$$

This sequence of forests i.e. $l_1, \ldots, l_n$ is called F-derivation in F-grammar FG of the forest l for l.

Definition 5. *F-language generated* by F-grammar $FG = \langle S_1^F, \Sigma_1^F, P^F, \underset{\downarrow}{s_0} \rangle$ is the following set of forests

$$L(FG) = \{ l \in \Sigma^F : \underset{\downarrow}{s_0} \overset{*}{\underset{FG}{\Rightarrow}} l \} .$$

Definition 6. Two F-grammars FG_1 and FG_2 are called *equivalent* if and only if

$$L(FG_1) = L(FG_2) .$$

Now two types of F-grammars are introduced.

Definition 7. A F-grammar $FG = \langle S_1^F, \Sigma_1^F, P^F, \underset{\downarrow}{s_0} \rangle$ is called *context-free* if and only if a set P^F consists of productions of the form:

$$\underset{\downarrow}{s} \Rightarrow \underset{\downarrow}{s_1} \downarrow \underset{\downarrow}{s_2} \rightarrow \underset{\downarrow}{s_3} ,$$

$$\underset{\downarrow}{s} \Rightarrow \underset{\downarrow}{\sigma} ,$$

$$\underset{\downarrow}{s} \Rightarrow e, \quad \underset{\downarrow}{s} \Rightarrow \bar{e} ,$$

where $\underset{\downarrow}{s}, \underset{\downarrow}{s_1}, \underset{\downarrow}{s_2}, \underset{\downarrow}{s_3} \in S_1^F$ and $\underset{\downarrow}{\sigma} \in \Sigma_1^F$.

Definition 8. A F-grammar $FG = \langle S_1^F, \Sigma_1^F, P^F, \underset{\downarrow}{s_0} \rangle$ is called *finite state* if and only if a set P^F consists of productions of the form:

$$\underset{\downarrow}{s} \Rightarrow \underset{\downarrow}{\sigma} \downarrow \underset{\downarrow}{s_1} \rightarrow \underset{\downarrow}{s_2} ,$$

$$\underset{\downarrow}{s} \Rightarrow e, \quad \underset{\downarrow}{s} \Rightarrow \bar{e} ,$$

where $\underset{\downarrow}{s}, \underset{\downarrow}{s_1}, \underset{\downarrow}{s_2} \in S_1^F$ and $\underset{\downarrow}{\sigma} \in \Sigma_1^F$.

On the basis of the framework thus introduced for F-languages the properties of families of F-languages generated by two types of F-grammars were tested. Now we present the obtained results.

The family of F-languages generated by finite state F-grammars is closed under the sum, intersection, F-concatenation and F-juxtaposition, and also under the homomorphism. Analogical results except the closure property for intersection hold for context-free family of F-languages.

Now we present the theorem connecting the notions of F-grammar and string grammar. By Σ_s^F we denote the following set of trees:

$$\Sigma_s^F = \{l \in \Sigma^F\colon\ l = (X, \Gamma, v),\ \Gamma | X \subset \hat{X} \times \hat{X}\hat{X},\ L(\Lambda) \in \hat{X},\ \Gamma(\Phi) \in \hat{X}\}.$$

Any forest of the above defined set is a tree without any branching.

THEOREM 1. *The families of finite state (context-free) string grammars and finite state (context-free) F-grammars generating F-languages contained in Σ_s^F are isomorphically equivalent.*

Proof. For any alphabet Σ we define a mapping $f_s\colon \Sigma^* \to \Sigma_s^F$ in the following way:

1° $f_s(\lambda) = e$, $f_s(\sigma) = \underset{\downarrow}{\sigma}$ for any $\sigma \in \Sigma$,

2° $f_s(w) = f_s(\sigma w') = f_s(\sigma) \downarrow f_s(w')$

for any $w \in \Sigma^+$ such that $w = \sigma w'$, $\sigma \in \Sigma$, $w' \in \Sigma^*$.

Of course f_s is an isomorphism.

Let $G = \langle S, \Sigma, P, s_0\rangle$ be any string grammar and let $FG = \langle \bar{S}_1^F, \Sigma_1^F, P^F, \underset{\downarrow}{\bar{s}_0}\rangle$ be any F-grammar. We prove the theorem for finite state grammars. The argumentation for context-free grammars is the same.

We connect the grammars G and FG in the following way:

$$\begin{aligned}
&\bar{S} = S \cup \{\bar{s}\} \quad \text{where } \bar{s} \notin S,\\
&s \Rightarrow \sigma s' \in P \quad \text{iff } \underset{\downarrow}{s} \Rightarrow \underset{\downarrow}{\sigma} \downarrow \underset{\downarrow}{s'} \to \underset{\downarrow}{\bar{s}},\ \underset{\downarrow}{\bar{s}} \Rightarrow \bar{e} \in P^F,\\
&s \Rightarrow \lambda \in P \quad \text{iff } \underset{\downarrow}{s} \Rightarrow e \in P^F,\\
&\underset{\downarrow}{\bar{s}_0} = \underset{\downarrow}{s_0}.
\end{aligned}$$

It is clear that for such connected grammars holds

$$f_s(L(G)) = L(FG).$$

Hence the theorem is proved.

The above theorem implies two corollaries.

COROLLARY 1. *The family of context-free F-languages is not closed under the intersection and complementation.*

COROLLARY 2. *The family of context-free F-languages is essentially larger than the finite state one.*

These facts are the consequence of the context-free string languages family properties.

Now we compare the notions of F-grammar and F-automata.

THEOREM 2. *The family of F-languages recognized by F-automata with context-free set of final words contains the family of finite state F-languages.*

Proof (outline). Let L will be any F-language generated by finite state F-grammar $FG = \langle S_1^F, \Sigma_1^F, P^F, s_0 \rangle$. We assume that F-grammar FG contains exactly one production $s^1 \Rightarrow e$ and one $s^2 \Rightarrow \bar{e}$, where $s^1, s^2 \in S_1^F - \{s_0\}$. We also assume that the above productions are the only productions with states s^1, s^2 on the left side, and that there are no productions of the type $s \Rightarrow \sigma \downarrow s' \rightarrow s^1$ for any $s, s' \in S_1^F$ and $\sigma \in \Sigma_1^F$. It is not difficult to show that for any finite state F-grammar there exists an equivalent one with such properties.

Now we construct an appropriate F-automaton. Let

$$A = \langle S' \cup \{\omega\}, \Sigma, \delta, s_0, D \rangle$$

where: $S' = S \cup \{\bar{s}\}$, $\bar{s} \notin S$,

S — alphabet S_1^F in grammar FG is defined under,

Σ — alphabet Σ_1^F in grammar FG is defined under, for any $s \in S'$, $\sigma \in \Sigma$

$$\delta(s, \sigma) = \begin{cases} \{(s', s'')\} & \text{if } s \Rightarrow \sigma \downarrow s' \rightarrow s'' \in P^F \\ \{(\bar{s}, \bar{s})\} & \text{otherwise}, \end{cases}$$

$$D = \{s^2, \omega\}^* \cap L(G) \cup \{s_0 \in S\colon s_0 \Rightarrow \bar{e} \in P^F\}$$

where $G = \langle \bar{S}, S \cup \{\omega\}, P, s_0 \rangle$ is the string grammar associated with F-grammar FG by the conditions:

S — alphabet S_1^F in grammar FG is defined under,

$\bar{S} = \{\bar{s}\cdot\ s \in S\}$,

$P = \{\bar{s} \Rightarrow \bar{s}'\bar{s}''\colon \exists \sigma \in \Sigma, s \Rightarrow \sigma \downarrow s \rightarrow s \in P^F\} \cup \{\bar{s} \Rightarrow s\colon \bar{s} \in \bar{S}\} \cup \{\bar{s}^1 \Rightarrow \omega\}$.

We add to the set D the state ω if $s_0 \Rightarrow e \in P^F$.

For the F-automaton thus defined it is obvious that the following equality holds

$$L(A) = L(FG).$$

Hence the proof is finished.

The following fact is also obtained which will be cited without a proof.

THEOREM 3. *The intersection of a finite state and context-free F-languages is a context-free F-language.*

Now we present the generalization of the so called Yield theorem. So far this theorem was proved for free languages only.

THEOREM 4. *A frontier of any finite state F-language is a context-free string language.*

Proof (outline). Let $FG = \langle S_1^F, \Sigma_1^F, P^F, s_0 \rangle$ will be any finite state F-grammar. We define a string grammar $G = \langle S, \Sigma, P, s_0 \rangle$ in the following way:

S, Σ — alphabets S_1^F, Σ_1^F are defined in FG under,

s_0 — the state for which $\underset{\downarrow}{s_0}$ is the initial forest in FG,

P contains the following productions:

if $\underset{\downarrow}{s} \Rightarrow \underset{\downarrow}{\sigma} \downarrow \underset{\downarrow}{s'} \rightarrow \underset{\downarrow}{s''} \in P^F$ then

$$s \Rightarrow \sigma s'' \in P \quad \text{if } (\underset{\downarrow}{s} \Rightarrow e \in P^F \text{ or } \underset{\downarrow}{s'} \Rightarrow \bar{e} \in P^F) \text{ and } \underset{\downarrow}{s''} \Rightarrow \bar{e} \in P^F,$$

$$s \Rightarrow s's'' \in P \quad \text{if } \underset{\downarrow}{s} \Rightarrow \underset{\downarrow}{\sigma'} \downarrow \underset{\downarrow}{\bar{s}} \rightarrow \underset{\downarrow}{\bar{s}'}, \underset{\downarrow}{s''} \Rightarrow \underset{\downarrow}{\sigma''} \downarrow \underset{\downarrow}{\bar{\bar{s}}} \rightarrow \underset{\downarrow}{\bar{\bar{s}}'} \in P^F,$$

$$s \Rightarrow s' \in P \quad \text{if } \underset{\downarrow}{s} \Rightarrow \underset{\downarrow}{\sigma'} \downarrow \underset{\downarrow}{\bar{s}} \rightarrow \underset{\downarrow}{\bar{s}'}, \underset{\downarrow}{s''} \Rightarrow e \in P^F,$$

$$s \Rightarrow \sigma \in P \quad \text{if } (\underset{\downarrow}{s'} \Rightarrow e \in P^F \text{ or } \underset{\downarrow}{s''} \Rightarrow e \in P^F) \text{ and } \underset{\downarrow}{s''} \Rightarrow \bar{e} \in P^F.$$

We add to P the production $s_0 \Rightarrow \lambda$ if $\underset{\downarrow}{s_0} \Rightarrow e \in P^F$ or $\underset{\downarrow}{s_0} \Rightarrow \bar{e} \in P^F$.

For grammar G thus defined the following proposition is proved:

For any finite state F-grammar FG and associated by the above conditions string grammar G holds

$$l \in L(FG) \quad \text{iff } \mathrm{fr}(l) \in L(G) \quad \text{for any } l \in \Sigma^F.$$

The proof of this proposition is given by induction taking into consideration the number of nodes of the forest l.

Basing on this fact the theorem for grammar G is proved. It directly results that G is a context-free string grammar. Hence the proof is complete.

COROLLARY 3. *Let FG will be any finite state F-grammar. The emptiness problem of the set L(FG) has the effective solution.*

It follows immediately from Theorem 4 and the corresponding theorem for string context-free languages — the theorem of Bar-Hillel, Perles and Shamir.

References

[1] W. S. Brainerd, *Tree Generating Regular Systems*, Information and Control 14 (1969).

[2] J. Doner, *Tree Acceptors and Some of Their Applications*, J. Comp. Syst. Sc. 4 (1970).

[3] M. Foryś, *O F-gramatykach generujących F-języki*, rozprawa doktorska, Kraków 1976.

[4] W. Foryś, *F-languages and F-automata*, Zeszyty Naukowe UJ Prace Matematyczne 22, 1980.

[5] R. Knast, *Net Languages*, Univ. Waterloo, reprint, 1974.

[6] W. C. Rounds, *Mappings and Grammars on Trees*, Math. Syst. Th. 4 (1970).

UNIWERSYTET JAGIELLOŃSKI
INSTYTUT INFORMATYKI
KRAKÓW (POLAND)