ZESZYTY NAUKOWE UNIWERSYTETU JAGIELLOŃSKIEGO
DLXXX PRACE MATEMATYCZNE, ZESZYT 22 1981

F-Languages and F-Automata

by Wit FORYŚ

The theory of tree languages was created in the sixties of 20'th century by J. W. Thatcher, J. B. Wright and J. E. Doner. It revealed that the employed definition apparatus leads to essential limitations of tree languages family. Therefore occasionaly attempts to formulate a new version of this theory are taken up.

In this paper the new approach to the theory of tree languages was presented. We extend our consideration to languages made up of finite sequences of trees-forests. According to N. Takahaski [6] these languages are called *F-languages*. We present a new definition apparatus and characterize the family of F-languages accepted by introduced F-automata.

We assume familiarity with notations used in set and languages theories. For any finite set $X = \{x_1, \ldots, x_k\}$ we put $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$. An alphabet Σ is a finite non-empty set.

Definition 1. [4]. *A Σ-valued forest* is a system $l = (\hat{X}, \Gamma, v)$, where X is a finite non-empty set (set of nodes), $\hat{X} = X \cup \{\Lambda, \Phi\}$, $\Lambda, \Phi \notin X$ (special nodes), $v\colon X \to \Sigma$ is the so called valued function and the function $\Gamma\colon X \to \hat{X}^* \hat{X}^*$ fulfill the conditions:

1° for every $x \in X$ $\Gamma(x) \in (X \cup \{\Lambda\})(X^* \cup \{\Phi\})$, $\Gamma(\Lambda) \in X^*$, $\Gamma(\Phi) \in \mathbf{X}^*$,

2° for every $x, x' \in X$ if $x \neq \Phi$ and $\Gamma(x) = \mathbf{x}'w$, $w \in X^* \cup \{\Phi\}$ or $x = \Phi$ and $\Gamma(x) = w_1 \mathbf{x}' w_2$, $w_1, w_2 \in \mathbf{X}^*$ then $N_{\mathbf{x} \in \Gamma(x')} = 1$,

3° for every $\mathbf{x} \in \hat{X}$ $N_{\mathbf{x} \in \Gamma(\Phi)} \leqslant 1$,

4° for every $x, x' \in X$ if $x \in \tilde{\Gamma}(x')$ then $x' \notin \tilde{\Gamma}^*(x)$.

Where the symbols used above have the following meanings:

$N_{\mathbf{x} \in \Gamma(x')}$ is the number of occurrences x in word $\Gamma(x')$,

$$\tilde{\Gamma}(x) = \{y \in \hat{X}\colon \Gamma(x) = w_1 y w_2,\ w_1, w_2 \in (X^* \cup \{\Lambda\})(X^* \cup \{\Phi\})\},$$

$$\tilde{\Gamma}^*(x) = \{y \in \hat{X}\colon y \in \tilde{\Gamma}(x) \text{ or } \exists x_1, \ldots, x_k \in \hat{X}, x_1 \in \tilde{\Gamma}(x), \ldots, x_k \in \tilde{\Gamma}(x_{k-1}), y \in \tilde{\Gamma}(x_k)\}.$$

The following notations will be also used:

$$\tilde{\mathbf{\Gamma}}(x) = \{y \in \hat{X}\colon \Gamma(x) = w_1 \mathbf{y} w_2,\ w_1, w_2 \in (X^* \cup \{\Lambda\})(X^* \cup \{\Phi\})\},$$

$$\tilde{\mathbf{\Gamma}}^*(x) = \{y \in \hat{X}\colon y \in \tilde{\mathbf{\Gamma}}(x) \text{ or } \exists x_1, \ldots, x_k \in \hat{X}, x_1 \in \tilde{\mathbf{\Gamma}}(x), \ldots, x_k \in \tilde{\mathbf{\Gamma}}(x_{k-1}), y \in \tilde{\mathbf{\Gamma}}(x_k)\}.$$

We denote by $\underset{\cdot}{a}$ and $\underset{\downarrow}{a}$ one node forest $l = (\{x\}, \Gamma, v)$ if $v(x) = a$ and $\Gamma(x) = \Lambda$ or $\Gamma(x) = \Lambda\Phi$ respectively. The set of all forests over an alphabet Σ is denoted by Σ_0^F.

Two operations on forests are introduced. These are F-concatenation and F-juxtaposition.

Definition 2. Let $l_1, l_2 \in \Sigma_0^F$, $l_i = (X_i, \Gamma_i, v_i)$ for $i = 1, 2$ and $l_2^1, \ldots, l_2^n$ be the sequence of forests isomorphic to l_2, $l_2^i = (X_2^i, \Gamma_2^i, v_2^i)$ for $i = 1, \ldots, n$, $n = |\tilde{\Gamma}_1(\Phi)|$, $X_1, X_2^1, \ldots, X_2^n$ mutually disjoint. *F-concatenation* is an operation of the form $\downarrow : \Sigma_0^F \times \Sigma_0^F \to \Sigma_0^F$ which for any forests l_1, l_2 gives as the result the forest l defined below:

$$l = l_1 \downarrow l_2, \quad l = (X, \Gamma, v), \quad X = X_1 \cup \bigcup_{i=1}^{n} X_2^i,$$

$v = v_1 \cup \bigcup_{i=1}^{n} v_2^i$ and Γ is equal respectively

$$\begin{array}{ll} \Gamma_1(\Lambda) & \text{for } x = \Lambda, \\ \Gamma_1(x) & \text{for } x \in X_1 - \tilde{\Gamma}_1(\Phi), \\ \Gamma_1(x) | \Phi \mapsto \Gamma_2^i(\Lambda) & \text{for } x \in \{x \in X_1 : x = \varepsilon_i(\Gamma_1(\Phi))\}, \\ \Gamma_2^i(x) | \Lambda \mapsto \varepsilon_i(\Gamma_1(\Phi)) & \text{for } x \in \bigcup_{i=1}^{n} \tilde{\Gamma}_2^i(\Lambda), \\ \Gamma_2^i(x) & \text{for } x \in \bigcup_{i=1}^{n} (X_2^i - \tilde{\Gamma}_2^i(\Lambda)), \\ \Gamma_2^1(\Phi) \ldots \Gamma_2^n(\Phi) & \text{for } x = \Phi. \end{array}$$

Where $x | a \mapsto y$ means that in the word x the letter a is changed by the word y; ε_i means i-projection. For any set X the symbol $|X|$ denotes the number of its elements (cardinality). The notion of isomorphic forests is used in the graph sense.

Definition 3. Let $l_1, l_2 \in \Sigma_0^F$, $l_i = (X_i, \Gamma_i, v_i)$ for $i = 1, 2$ and $X_1 \cap X_2 = \emptyset$. *F-juxtaposition* is an operation of the form $\to : \Sigma_0^F \times \Sigma_0^F \to \Sigma_0^F$ which for any forests l_1, l_2 gives as the result the forest l defined below:

$$l = l_1 \to l_2, \; l = (X, \Gamma, v), \; X = X_1 \cup X_2, \; v = v_1 \cup v_2$$

and Γ is equal respectively

$$\begin{array}{ll} \Gamma_1(\Lambda)\Gamma_2(\Lambda) & \text{for } x = \Lambda, \\ \Gamma_1(x) & \text{for } x \in X_1, \\ \Gamma_2(x) & \text{for } x \in X_2, \\ \Gamma_1(\Phi)\Gamma_2(\Phi) & \text{for } x = \Phi. \end{array}$$

The set Σ_0^F with F-concatenation and F-juxtaposition separately after the neutral elements e and $\bar{e}$ respectively are added, forms a monoid.

Let Σ^F denote the set $\Sigma_0^F \cup \{e, \bar{e}\}$. Any subset of Σ^F is called a F-language.

Based on the definitions 2 and 3 F-concatenation and F-juxtaposition of F-languages are defined. First we assume that for any $l \in \Sigma_0^F$, $l = (X, \Gamma, v)$ we have

$$e \to l = l \to e = e, \quad \bar{e} \downarrow l = \bar{e}, \quad l \downarrow \bar{e} = l',$$

where $l' = (X, \Gamma', v)$ and Γ' is equal respectively

$$\begin{array}{ll} \Gamma(x) & \text{for } x \in \hat{X} - (\Gamma(\Phi) \cup \{\Phi\}), \\ \Gamma(x) | \Phi \mapsto \lambda & \text{for } x \in \tilde{\Gamma}(\Phi), \\ \lambda & \text{for } x = \Phi . \end{array}$$

Now we extend F-concatenation of the forests.

Definition 4. Let $l_1, l_{21}, \ldots, l_{2n} \in \Sigma^F$, $n = |\tilde{\Gamma}_1(\Phi)|$,

$$l_1 = (X_1, \Gamma_1, v_1), \quad l_{2i} = (X_{2i}, \Gamma_{2i}, v_{2i}) \quad \text{for } i = 1, \ldots, n .$$

F-concatenation of forest l_1 and the sequence of forests $l_{21}, \ldots, l_{2n}$ gives as the result the forest l defined below:

$$l = l_1 \downarrow (l_{21}, \ldots, \ldots, l_{2n}), \quad l = (X, \Gamma, v), \quad X = X_1 \cup \bigcup_{i=1}^{n} X_{2i},$$

$v = v_1 \cup \bigcup_{i=1}^{n} v_{2i}$ and Γ is equal respectively

$$\begin{array}{ll} \Gamma_1(\Lambda) & \text{for } x = \Lambda \\ \Gamma_1(x) & \text{for } x \in X_1 - \tilde{\Gamma}_1(\Phi) \\ \Gamma_1(x) | \Phi \mapsto \Gamma_{2i}(\Lambda) & \text{for } x \in \{x \in X_1 \colon x = \varepsilon_i(\Gamma_1(\Phi))\} \\ \Gamma_{2i}(x) | \Lambda \mapsto \varepsilon_i(\Gamma_1(\Lambda)) & \text{for } x \in \bigcup_{i=1}^{n} \tilde{\Gamma}_{2i}(\Lambda) \\ \Gamma_{2i}(x) & \text{for } x \in \bigcup_{i=1}^{n} (X_{2i} - \tilde{\Gamma}_{2i}(\Lambda)) \\ \Gamma_{2i}(\Phi) \ldots \Gamma_{2n}(\Phi) & \text{for } x = \Phi \end{array}$$

Definition 5. Let $L_1, L_2 \subset \Sigma^F$ be any F-languages. *F-concatenation of F-languages L_1 and L_2* is defined as follows

$$L_1 \downarrow L_2 = \{l_1 \downarrow (l_{21}, \ldots, l_{2n}) \colon l_1 = (X_1, \Gamma_1, v_1) \in L_1, l_{21}, \ldots, l_{2n} \in L_2, n = |\tilde{\Gamma}_1(\Phi)|\} .$$

Definition 6. Let $L_1, L_2 \subset \Sigma^F$ be any F-languages. *F-juxtaposition of F-languages L_1 and L_2* is defined as follows

$$L_1 \to L_2 = \{l_1 \to l_2 \colon l_1 \in L_1, l_2 \in L_2\} .$$

Now we introduce the notion of standard partition.

Definition 7. Let $l \in \Sigma^F$, $l = (X, \Gamma, v)$. *The standard partition* of any forest l is composed of three forests $l_1, l_2, l_3 \in \Sigma^F$ defined as follows

$$l_i = (X_i, \Gamma_i, v_i) \quad \text{for } i = 1, 2, 3$$

and

$$X_1 = \{\varepsilon_1(\Gamma(\Lambda))\} = \{x_1\}, \quad v_1 = v | X_1$$

and Γ_1 is equal respectively

$$\begin{array}{ll} x_1 & \text{for } x = \Lambda\,, \\ \Lambda\Phi & \text{for } x = x_1\,, \\ x_1 & \text{for } x = \Phi\,, \end{array}$$

$$X_2 = \tilde{\Gamma}^*(x_1) - \{\Phi\}, \quad v_2 = v | X_2$$

and Γ_2 is equal respectively

$$\begin{array}{rl} \Gamma(x_1)|\Lambda \mapsto \lambda & \text{for } x = \Lambda\,, \\ \Gamma(x)|x_1 \mapsto \Lambda & \text{for } x \in \tilde{\Gamma}(x_1) - \{\Phi\}\,, \\ \Gamma(x) & \text{for } x \in X_2 - \tilde{\Gamma}_2(\Lambda)\,, \\ (\varepsilon_1, \ldots, \varepsilon_k)(\Gamma(\Phi))|x_1 \to \Lambda & \text{for } x = \Phi, \quad k = \max\{i\colon \varepsilon_i(\Gamma(\Phi)) \in \tilde{\Gamma}^*(x_1)\}\,, \end{array}$$

$$X_3 = X - (X_1 \cup X_2), \quad v_3 = v | X_3$$

and Γ_3 is equal respectively

$$\begin{array}{ll} \Gamma(\Lambda)|x_1 \mapsto \lambda & \text{for } x = \Lambda\,, \\ \Gamma(x) & \text{for } x \in X_3\,, \\ (\varepsilon_{k+1}, \ldots, \varepsilon_{k+m})(\Gamma(\Phi)) & \text{for } x = \Phi,\ k \text{ as above, } k+m = |\tilde{\Gamma}(\Phi)|\,. \end{array}$$

The standard partition is denoted by the symbol $[l_1; l_2; l_3]$. It is easy to see that for any forest l and its standard partition it holds that

$$l = l_1 \downarrow l_2 \to l_3\,.$$

It will be used the notation $l = [l_1; l_2; l_3]$.

The basic notion of F-languages theory that is the notion of F-automaton is defined now. Some suggestions for this notion are due to R. Knast.

Definition 8. *F-automaton* is a system $A = \langle S \cup \{\omega\}, \Sigma, \delta, s_0, D\rangle$, where S is a finite non-empty set of states, ω distinguished state, $\omega \notin S$, Σ alphabet, $\delta\colon S \times \Sigma \to S \times S$ is a transition relation, s_0 the initial state, $s_0 \in S$ and $D \subset (S \cup \{\omega\})^+$ is a set of final words.

It is assumed that $\delta(s, e) = \{\omega\}$ and $\delta(s, \bar{e}) = \{s\}$. Now the relation δ is generalized by induction to the set Σ^F, the number of the forest nodes being taken into account. For any $s \in S$ we have:

for $l = \underset{\bullet}{\sigma}$

$$\delta(s, l) = \delta(s, \sigma)\,,$$

for $l = \underset{\downarrow}{\sigma}$

$$\delta(s, l) = \delta(\varepsilon_1(\delta(s, \sigma)), e)\varepsilon_2(\delta(s, \sigma))\,,$$

for $l = (X, \Gamma, v)$ such that $|X| > 1$

$$\delta(s, l) = \bigcup_{\bar{s} \in T} \delta(s', l_2)\delta(s'', l_3)$$

where $l = [l_1; l_2; l_3]$, $l_1 = \underset{\downarrow}{\sigma_1}$, $\bar{s} = (s', s'')$, $T = \delta(s, \sigma_1)$.

Definition 9. A forest $l \in \Sigma^F$ is *recognized* (*accepted*) by F-automaton

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

if and only if

$$\delta(s_0, l) \cap D \neq \emptyset .$$

The set of all forests recognized by F-automaton A is called F-language recognized by this F-automaton and denoted by the symbol $L(A)$.

By the above introduced definitions the properties of F-languages family recognized by the class of F-automata were examined. It is assumed that the set D of final words is regular in Chomsky sense. Now the obtained results are presented.

Within the family of F-languages recognized by F-automata with regular set D the closure properties under F-concatenation, F-juxtaposition, sum, intersection, homomorphism and its inverse image are obtained. If the consideration are limited to the deterministic F-automata only (i.e. δ is a function) closure under complementation holds.

Now we compare families of all F-languages and F-languages recognized by F-automata.

THEOREM 1. *The family of recognized F-languages is the essential subfamily of the all F-languages family.*

Proof. We shall prove that the F-language $\{\underset{\cdot}{a}^p\} = \{\underbrace{\underset{\cdot}{a} \dots \underset{\cdot}{a}}_{p}\}_{p \in N_p}$ where N_p is the set of all prime numbers is not recognized by any F-automaton with a regular set of final words.

Let $A = \langle S \cup \{\omega\}, \Sigma, \delta, s_0, D\rangle$ be F-automaton such that $L(A) = \{\underset{\cdot}{a}^p\}$. We define a string grammar $G = \langle \mathbf{S}, S, \mathbf{s}_0, P\rangle$ as follows: S as in F-automaton A, $\mathbf{S} = \{\mathbf{s}\colon s \in S\}$, P contains productions $\mathbf{s} \Rightarrow s's''$ if there exists $\sigma \in \Sigma$ such that $\delta(s, \sigma) = (s', s'')$, and $\mathbf{s} \Rightarrow s$ for any $\mathbf{s} \in \mathbf{S}$. The above defined grammar G is regular (Chomsky 3'th type).

Observe that $L(G) = \delta(s_0, \{\underset{\cdot}{a}^n\})$ where $\{\underset{\cdot}{a}^n\} = \{\underbrace{\underset{\cdot}{a} \dots \underset{\cdot}{a}}_{n}\}_{n \in N}$ (N — the set of all positive integers).

It follows by the form of F-language $\{\underset{\cdot}{a}^p\}$ that

$$\delta(s_0, \{\underset{\cdot}{a}^p\}) = L(G) \cap D .$$

The set $L(G) \cap D$ is a regular one and by G_1 is denoted the grammar which generates it. For the prime numbers $p_1 \neq p_2$ the words $\delta(s_0, \underset{\cdot}{a}^{p_1})$, $\delta(s_0, a^{p_2})$ are of a different length which is equal to p_1+1 and p_2+1 respectively. Thus the language $L(G)$ contains words of the length $p+1$, where p is any prime number. After replacing the last letter in each of the word from $L(G)$ by an empty word we obtain a regular set $L(G_2)$ (G_2 — the regular grammar of which this set is generated). Substituting for each of the letter the same string language formed from one letter — e.g. $\{a\}$ we finally obtain a string language $\{a^p\}$ where p is any prime number.

Hence according to the regularity of $L(G_2)$ language $\{a^p\}$ is a regular one. But from the string language theory we know that this statement is false. So we came to a contradiction and the proof is finished.

Theorem 1 gives as a conclusion a nontriviality of all the problems considered in our paper.

Now the theorem concerning the problem of effective method is presented.

THEOREM 2. *Let* $A = \langle S \cup \{\omega\}, \Sigma, \delta, s_0, D\rangle$ *be any F-automaton. There exists an effective method for solving the emptiness problem of the set* $L(A)$.

Proof. We define a string grammar $G = \langle \boldsymbol{S}, S \cup \{\omega\}, \boldsymbol{s}_0, P\rangle$ where $\boldsymbol{S}$ is defined as in the proof of Theorem 1 and P is the set of the following productions:

for $s, s', s'' \in S$, $\sigma \in \Sigma$ such that $(s', s'') \in \delta(s, \sigma)$

$$\boldsymbol{s} \Rightarrow \boldsymbol{s}'\boldsymbol{s}'', \quad \boldsymbol{s} \Rightarrow \omega s,$$

for every $\boldsymbol{s} \in \boldsymbol{S}$

$$\boldsymbol{s} \Rightarrow s, \quad \text{and} \quad \boldsymbol{s}_0 \Rightarrow \omega .$$

Grammar G is a context-free one and it is easy to see that the words generated by it in the alphabet $S \cup \{\omega\}$ are identical as the result of recognizing forests from Σ^F by F-automaton A. Of course there exists a regular grammar G_2 such that $L(G_2) = D$. The following equivalences are true:

$$L(A) = \emptyset \Leftrightarrow \forall l \in \Sigma^F \delta(s_0, l) \cap D = \emptyset \Leftrightarrow \delta(s_0, \Sigma^F) \cap D = \emptyset \Leftrightarrow L(G_1) \cap L(G_2) = \emptyset .$$

According to the corresponding theorem [1] a string language $L(G_1) \cap L(G_2)$ is a context-free one. So there exists a context-free grammar G such that

$$L(G) = L(G_1) \cap L(G_2) .$$

As it is shown [1] there is an effective solution to the emptiness problem of the language $L(G)$ for any context-free grammar G. Hence the problem proves to be solvable.

References

[1] Y. Bar-Hillel, J. Perles, E. Shamir, *On Formal Properties of Simple Phrase-Structure Grammars*, Z. Phonetic. Sprach. Kommunik. 14 (1961).

[2] J. R. Büchi, *Weak Second-Order Arithmetic and Finite Automata*, Z. Math. Logik Grund. Math. 6 (1960).

[3] W. Foryś, *O F-językach rozpoznawanych przez F-automaty*, rozprawa doktorska, Kraków 1976.

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

[5] M. O. Rabin, *Decidability of Second-Order Theories and Automata on Infinite Trees*, Trans. Amer. Math. Soc. 141 (1969).

[6] N. Takahaski, *Generalization of Regular Sets...*, Information and Control, 27 (1975).

UNIWERSYTET JAGIELLOŃSKI
INSTYTUT INFORMATYKI
KRAKÓW (POLAND)