PARALLEL FINITE AUTOMATA SYSTEMS COMMUNICATING BY TRANSITIONS

Patrik Petřík
Doctoral Degree Programme (1), FIT BUT
E-mail: ipetrik@fit.vutbr.cz

Supervised by: Alexander Meduna
E-mail: meduna@fit.vutbr.cz

ABSTRACT

We introduce parallel finite automata systems communicating by transitions. Similar systems were already introduce, but the principle of communication were different. Simple question is, if the possibility of this communication brings similarly results like by the earlier defined systems. In this paper we introduce, that the class of languages defined by parallel finite automata systems communicating by transitions is a subset of the class of languages defined by *n*-right linear simple matrix languages.

1 INTRODUCTION

Various types of grammar systems has been recently investigated in formal language theory (see [5]). These systems consist of various types of grammars which cooperate and the systems generates the language. On the other hand there are missing works about systems, thats combine different types of automata. Works investigating automata systems already exist (see [6, 2]), but the area of automata systems is not so comprehensive like the area of grammar systems. Motivation of this work is that there are missing descriptions of automata systems, which has the same expressing power like some already known grammar systems or grammars.

In this paper, we focus on the theory of parallel (communicating) automata systems, which components are finite state machines. This components communicate by transition, when this approach seems to be more natural then communication by states. And contrary of the systems defined in [6] each component has only part of the input string. We define this systems and prove their properties considering already known grammar systems or other grammar structures. Paper is based on [1].

2 PRELIMINARIES

This Paper assumes that the reader is familiar with the formal language theory (see [7]). For an alphabet V, V^* represents the free monoid generated by V under the operation of concatenation. The identity of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$. For a finite set A we denote by $card(A)$ the cardinality of A. For every $\omega \in V^*$, $|\omega|$ denotes the length of ω.

A *finite state machine* (*FSM*) is a quintuple, $M = (Q, \Sigma, R, s, F)$, where Q is a finite, non-empty set of states. Σ is input alphabet (finite, non-empty set of symbols). R is the finite set of rules of the form $pa \rightarrow q$, where $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$. $s \in Q$ is an initial state and $F \subseteq Q$ is the set of final states.

Right linear grammar is quadruple, $G = (N, T, P, S)$, where N and T are two disjoint alphabets. Symbols in N and T are referred to as nonterminals and terminals, respectively, and $S \in N$ is the start symbol of G. P is a finite set of rules of the form $x \rightarrow y$, where $x \in N$ and $y \in T^* \cup T^*N$. *Strictly right linear grammar* is a *right linear grammar* where P is a finite set of rules of the form $x \rightarrow y$, $x \in N$ and $y \in TN \cup \{\varepsilon\}$.

For $m, n \geq 1$ an *m-parallel n-right linear simple matrix grammar* (*n-Pn-G*) is an $(mn+3)$-tuple $G = (N_{11}, \ldots, N_{1n}, \ldots, N_{m1}, \ldots, N_{mn}, T, S, P)$, where N_{ij}, $1 \leq i \leq m$, $1 \leq j \leq n$ are mutually disjoint nonterminal alphabets, T is a terminal alphabet, S is a sentence symbol, S not in $N_{11} \cup \ldots \cup N_{mn} \cup T$ and P is a finite set of matrix rules. If $m = 1$ then we obtain an *n-right linear simple matrix grammar* and if $n = 1$ then we obtain an *m-parallel right linear grammar*. Definitions and properties of these grammars may be found in [3, 4], where is also definition of *slot grammar* used hereafter.

3 DEFINITIONS

In this section, we define the fundamental notions of this paper. We introduce parallel finite automata systems communicating by transitions. Components of these systems are finite state automata.

Definition 3.1: A parallel finite automata system communicating by transitions ($pc\ as\ fsa(n)$) is an $(n+1)$-tuple $AS = (\Sigma, A_1, A_2, \ldots, A_n)$, where Σ is the input alphabet, $A_i = (Q_i, \Sigma, R_i, s_i, F_i)$, $1 \leq i \leq n$, are finite automata with the set of states Q_i, $s_i \in Q_i$ (the initial state), $F_i \subseteq Q_i$ (the set of final states) and R_i is the transition function of the automaton i defined as follows: $R_i = \{pa \rightarrow q | p, q \in Q_i, a \in \Sigma \cup \{\varepsilon\}\} \cup \{p \xrightarrow{c} q | p, q \in Q_i, c \in \cup_{j=1}^{n} Q_j\}$, where $p \xrightarrow{c} q$ represents communication rule. Note that $\bigcap_{j=1}^{n} Q_j = \emptyset$.

Definition 3.2: Let $AS = (\Sigma, A_1, A_2, \ldots, A_n)$ be a $pc\ as\ fsa(n)$. We define a set of communicating states of i-th component as follows: $C_i = \{q | q \in Q_i \wedge \exists (q \xrightarrow{c} r) \in R_i, c \in \cup_{j=1}^{n} Q_j, r \in Q_i\}$.

Definition 3.3: Let $AS = (\Sigma, A_1, A_2, \ldots, A_n)$ be a $pc\ as\ fsa(n)$. We define a set of query states of i-th component as follows: $I_i = \{c | c \in Q_i \wedge \exists j | (q \xrightarrow{c} r) \in R_j, q, r \in Q_j, j \in \{1, \ldots, n\}\}$.

The set C_i represent states, from them there is a possibility to communicate and the set I_i represent states, that may be a target of communication. The automata $A_1, A_2, \ldots, A_n$ are called the *components* of the system AS. If there exists just one $1 \leq i \leq n$ such that $C_i \neq \emptyset$, then the system is said to be *centralized*, the master of this system being the component i.

Definition 3.4: Let $AS = (\Sigma, A_1, A_2, \ldots, A_n)$ be a $pc\ as\ fsa(n)$. Configuration of AS is a string $\chi = q_1 v_1 q_2 v_2 \cdots q_n v_n$, where $q_i \in Q_i$ is the current state of component i, $v_i \in \Sigma^*$ is the remaining (unread) part of the input word appurtenant to the component i, $i \in \{1, \ldots, n\}$.

Definition 3.5: Let $AS = (\Sigma, A_1, A_2, \ldots, A_n)$ be a $pc\ as\ fsa(n)$. We define relation on the set of configurations of AS in the following way: $q_1 v_1 q_2 v_2 \cdots q_n v_n \vdash p_1 u_1 p_2 u_2 \cdots p_n u_n$, where $q_i, p_i \in Q_i$, $v_i = a_i u_i$, $a_i \in \Sigma \cup \{\varepsilon\}$, $v_i, u_i \in \Sigma^*$, $1 \leq i \leq n$, iff one of the following conditions holds:

1. for all i, $1 \leq i \leq n$ holds, that $q_i \notin C_i$ and exists $r_i = (q_i a_i \rightarrow p_i) \in R_i$,

2. for all j, $1 \le j \le n$, such that $q_j \notin C_j$ exists $r_j = (q_j a_j \to p_j) \in R_j$ and for all the other i, $1 \le i \le n$, such that $q_i \in C_i$ exists r_i such that: if $r_i = (q_i \xrightarrow{c} p_i) \in R_i$ then there exists k, $1 \le k \le n$, such that $c = q_k$, $q_k \notin C_k$ and $v_i = u_i$ else $r_i = (q_i a_i \to p_i) \in R_i$

Definition 3.6: Let $AS = (\Sigma, A_1, A_2, \ldots, A_n)$ be a $pc\ as\ fsa(n)$. The language accepted by parallel finite automata system communicating by transitions AS, $\mathcal{L}(AS)$ is defined as

$$\mathcal{L}(AS) = \{w | s_1 v_1 s_2 v_2 \cdots s_n v_n \vdash^* f_1 f_2 \cdots f_n, w \in \Sigma^*, v_1 v_2 \cdots v_n = w, f_i \in F_i, 1 \le i \le n\},$$

where $\vdash^*$ denotes the reflexive and transitive closure of $\vdash$.

We shall denote by *pc as fsa(n)* a parallel finite automata system (communicating by transitions) of degree n and *PC AS FSA(n)* a class of languages accepted by *pc as fsa(n)*.

4 COMPUTATIONAL POWER

In this section we show basic results related to *parallel finite automata systems (communicating by transition)*, we compare the computational power with the computational power of *m-parallel n-right linear simple matrix grammar*.

Theorem 4.1: For each *n-right linear simple matrix grammar G* there exists *parallel finite automata system AS (communicating by states) of degree n*, such that $\mathcal{L}(G) = \mathcal{L}(AS)$.

Proof. Let $G = (N_1, N_2, \ldots, N_n, T, S, P)$ be a $n-RLSMG$ (see [3]), where N_i, $1 \le i \le n$ are mutually disjoint *nonterminal alphabets*, T is a *terminal alphabet*, S is a *sentence symbol*, S not in $N_1 \cup \cdots \cup N_n \cup T$ and P is a finite set *matrix rules*. A matrix rule can be in one of the following three forms:

i. $[S \to X_1 \cdots X_n]$, $X_i \in N_i$, $1 \le i \le n$,

ii. $[X_1 \to a_1, \ldots, X_n \to a_n]$, $X_i \in N_i$, $a_i \in T^*$, $1 \le i \le n$,

iii. $[X_1 \to a_1 Y_1, \ldots, X_n \to a_n Y_n]$, $X_i, Y_i \in N_i$, $a_i \in T^*$, $1 \le i \le n$.

Proof has two parts. First, we prove, that for each $n-RLSMG\ G$ there exists an equivalent (except for homomorphism h defined hereafter) $n-RLSMG\ G'$ containing rules of specific form. After that we prove, that for this G' there exists $pc\ as\ fsa\ AS$, such that $\mathcal{L}(G') = \mathcal{L}(AS)$. Let G be a $n-RLSMG$ defined above. For this G construct $n-RLSMG\ G' = (N'_1, N'_2, \ldots, N'_n, T', S, P')$, such that $N'_i = N_i \cup \{X^j_{i_k} | 1 \le j \le card(P),$
$1 \le k \le (2n-2)\}$, $1 \le i \le n$, $T' = T \cup \{\overline{A} | A \in \bigcup_{i=1}^{n} N'_i\}$ and P' contains following rules:

i. $r = [S \to X_1 \cdots X_n]$, $X_i \in N_i$, $1 \le i \le n$, iff $r \in P$,

ii. $r_{j_1} = [X_1 \to a_1 X^j_{1_1}, X_2 \to X^j_{2_1}, \ldots, X_n \to X^j_{n_1}]$,
$r_{j_2} = [X^j_{1_1} \to X^j_{1_2}, X^j_{2_1} \to \overline{X^j_{1_1}} X^j_{2_2}, \ldots, X^j_{n_1} \to \overline{X^j_{1_1}} X^j_{n_2}]$,
$r_{j_3} = [X^j_{1_2} \to X^j_{1_3}, X^j_{2_2} \to a_2 X^j_{2_3}, \ldots, X^j_{n_2} \to X^j_{n_3}]$,
$\vdots$
$r_{j_{2n-2}} = [X^j_{1_{2n-3}} \to \overline{X^j_{n-1_{2n-3}}} X^j_{1_{2n-2}}, \ldots, X^j_{n_{2n-3}} \to \overline{X^j_{n-1_{2n-3}}} X^j_{n_{2n-2}}]$,

$r_{j_{2n-1}} = [X^j_{1_{2n-2}} \to \varepsilon, X^j_{2_{2n-2}} \to \varepsilon, \ldots, X^j_{n_{2n-2}} \to a_n]$,
iff exists j, $1 \leq j \leq card(P)$, such that
$r_j = [X_1 \to a_1, X_2 \to a_2, \ldots, X_n \to a_n] \in P$, $X_i \in N_i$, $a_i \in T^*$, $1 \leq i \leq n$,

iii. and similarly for the last form of matrix rules.

Let h be a total function from T'^* to T^* such that $h(uv) = h(u)h(v)$ for every $u, v \in T'^*$. h is a homomorphism, such that $h(a) = a$ for every $a \in T$ and $h(\overline{a}) = \varepsilon$ for every $\overline{a} \in T' \backslash T$.

Claim 4.1: Let $S \Rightarrow^m w_1 w_2 \ldots w_n$ in G, where $m \geq 0$, $w_i = t_i u_i$, $t_i \in T^*$, $u_i \in N_i \cup \{\varepsilon\}$, $1 \leq i \leq n$. Then $S \Rightarrow^{(2n-1)(m-1)+1} v_1 v_2 \ldots v_n$ in G', where $v_i = t'_i u_i$, $h(t'_i) = t_i$, $t'_i \in T'$.

Proof. Claim 4.1 is proved by induction:

Basic: Let $m = 1$. Then $S \Rightarrow^1 X_1 \cdots X_n$ in G. Observe that $S \Rightarrow^1 X_1 \cdots X_n$ in G'.

Induction hypothesis: Assume that Claim 4.1 holds for all m-step derivations, where $m = 0, \ldots, k$ for some $k \geq 0$.

Induction step: Consider $S \Rightarrow^{k+1} y_1 y_2 \cdots y_n$ in G. Then there is sentential form $u_1 A_1 u_2 A_2 \cdots u_n A_n$ in G, where $u_i \in T^*$, $A_i \in N_i$, such that $S \Rightarrow^k u_1 A_1 u_2 A_2 \cdots u_n A_n \Rightarrow u_1 x_1 u_2 x_2 \cdots u_n x_n$, where $u_i x_i = y_i$, for all $i = 1, \ldots, n$.

1. $S \Rightarrow^k u_1 A_1 u_2 A_2 \cdots u_n A_n$ in G implies $S \Rightarrow^{(2n-1)(k-1)+1} u'_1 A_1 u'_2 A_2 \cdots u'_n A_n$ in G', where $u_i = h(u'_i)$, $1 \leq i \leq n$.

2.Let $u_1 A_1 u_2 A_2 \cdots u_n A_n \Rightarrow u_1 x_1 u_2 x_2 \cdots u_n x_n$ in G. Then there holds: $[A_1 \to x_1, \ldots, A_n \to x_n] \in P$, and it implies that $[A_1 \to y_1 A^j_{1_1}, \ldots, A_n \to A^j_{n_1}] \in P'$, $[A^j_{1_1} \to A^j_{1_2}, \ldots, A^j_{n_1} \to \overline{A^j_{1_1}} A^j_{n_2}] \in P'$, $\ldots$, $[A_{1_{2n-2}} \to z_1, \ldots, A^j_{n_{2n-2}} \to x_n] \in P'$, where $x_i = y_i z_i$, $y_i \in T^*$, $z_i \in N_i \cup \{\varepsilon\}$, $1 \leq i \leq n$, for some $1 \leq j \leq card(P)$. So $u'_1 A_1 u'_2 A_2 \cdots u'_n A_n \Rightarrow^{2n-1} u'_1 y_1 \overline{y_1} z_1 u'_2 y_2 \overline{y_2} z_2 \cdots u'_n y_n \overline{y_n} z_n$ in G', where $u_i = h(u'_i)$, $x_i = y_i h(\overline{y_i}) z_i = y_i z_i$.

1. and 2. imply that $S \Rightarrow^{(2n-1)(k-1)+1} u'_1 A_1 u'_2 A_2 \cdots u'_n A_n \Rightarrow^{2n-1} u'_1 y_1 \overline{y_1} z_1 u'_2 y_2 \overline{y_2} z_2 \cdots u'_n y_n \overline{y_n} z_n$ thus $S \Rightarrow^{(2n-1)(k+1-1)+1} u'_1 y_1 \overline{y_1} z_1 u'_2 y_2 \overline{y_2} z_2 \cdots u'_n y_n \overline{y_n} z_n$, where $h(u'_1 y_1 \overline{y_1}) z_1 h(u'_2 y_2 \overline{y_2}) z_2 \cdots h(u'_n y_n \overline{y_n}) z_n = u_1 x_1 \ldots u_n x_n$.

Claim 4.2: For every $n-RLSMG$ G' defined above there exists pc as $fsa(n)$ AS, $AS = (\Sigma, A_1, \ldots, A_n)$, $A_i = (Q_i, \Sigma, R_i, S_i, F_i)$, $1 \leq i \leq n$, such that if $S \Rightarrow^m w_1 w_2 \cdots w_n$ in G', where $w_i = u'_i q'_i$, $u'_i \in T'^*$ $q'_i \in N'_i \cup \{\varepsilon\}$, $1 \leq i \leq n$, then $S_1 u_1 S_2 u_2 \ldots S_n u_n \vdash^m q_1 q_2 \cdots q_n$ in AS, where $u_i = h(u'_i) \in \Sigma^*$, $q_i \in Q_i$, for all $1 \leq i \leq n$.

Proof. This Claim is proved by induction. For the sake of simplicity consider that G' is *n-strictly right linear simple matrix grammar*. Let $G' = (N'_1, N'_2, \ldots, N'_n, T', S, P')$ (as defined above), then G' consists of n slot grammars (see [3]), of the form $G_i = (N'_i \cup \{S_i\}, T', S_i, P'_i)$, where $S_i = S$, $S_i \notin N'_i$, $P'_i = \{S_i \to X_i | [S \to X_1 \cdots X_i \cdots X_n] \in P'\} \cup \{X_i \to x_i | [X_1 \to x_1, \ldots X_i \to x_i, \ldots, X_n \to x_n] \in P'\}$. For each component of AS $A_i = (Q_i, \Sigma, R_i, S_i, F_i)$, let $Q_i = N'_i \cup \{S_i\}$, $\Sigma = T$, $R_i = \{pa \to q | (p \to aq) \in P'_i\} \cup \{p \xrightarrow{c} q | (p \to \overline{c}q) \in P'_i\}$, $F_i = \{q | (q \to \varepsilon) \in P'_i\}$. We just used an generally known algorithm for converting *strictly right linear grammar* to a *non-deterministic finite state machine*.

Basic: Let $m = 0$. Then $S \Rightarrow^0 S$ in G'. Observe that $S_1 S_2 \cdots S_n \vdash^0 S_1 S_2 \cdots S_n$ in AS.

Induction hypothesis: Assume that Claim 4.2 holds for all m-step derivations, where $m = 0, \ldots, k$, for some $k \geq 0$.

Induction step: Consider $S \Rightarrow^{k+1} x'_1 u'_1 q_1 \cdots x'_n u'_n q_n$ in G', then there exists sentential form $x'_1 p_1 \cdots x'_n p_n$ in G', where $p_i \in N'_i$ such that $S \Rightarrow^k x'_1 p_1 \cdots x'_n p_n \Rightarrow x'_1 u'_1 q_1 \cdots x'_n u'_n q_n$ in G'.

1. $S \Rightarrow^k x'_1 p_1 \cdots x'_n p_n$ in G' implies $S_1 x_1 S_2 x_2 \ldots S_n x_n \vdash^k p_1 p_2 \cdots p_n$ in AS, $x_i = h(x'_i)$, for all $i = 1, \ldots, n$ by the induction hypothesis.

2. Let $x'_1 p_1 \cdots x'_n p_n \Rightarrow x'_1 u'_1 q_1 \cdots x'_n u'_n q_n$ in G'. Then, there holds $(p_i \to u'_i q_i) \in P'_i$. Algorithm used by definition of AS implies that $(p_i u_i \vdash q_i) \in R_i$, $u_i \in \Sigma$ if $u_i = h(u'_i)$, or $(p_i \xrightarrow{c} q_i)$, $u'_i = \overline{c}$ if $\varepsilon = h(u'_i)$ for all $i = 0, \ldots, n$. So $p_1 u_1 \cdots p_n u_n \vdash q_1 \cdots q_n$ in AS.

1. and 2. imply that $S_1 x_1 u_1 S_2 x_2 u_2 \ldots S_n x_n u_n \vdash^{k+1} q_1 q_2 \cdots q_n$, where $x_i u_i = h(x'_i u'_i)$, $1 \leq i \leq n$.

Consider Claim 4.1 with $m \geq 0$, $w_i \in T^*$, for all $1 \leq i \leq n$. At this point, if $S \Rightarrow^m w_1 \cdots w_n$ in G, then $S \Rightarrow^{(2n-1)(m-1)+1} v_1 \cdots v_n$ in G', where $h(v_i) = w_i$, for all $1 \leq i \leq n$. Hence $\mathcal{L}(G) \subseteq h(\mathcal{L}(G'))$. Consider Claim 4.2 with $m \geq 0$, $w_1 \in T'^*$, for all $1 \leq i \leq n$. At this point, if $S \Rightarrow^m w_1 \cdots w_n$ in G', then $S_1 u_1 \cdots S_n u_n \vdash^m q_1 \cdots q_n$ in AS, where $u_i = h(w_i)$, $q_i \in F_i$, for all $1 \leq i \leq n$. Hence $h(\mathcal{L}(G')) \subseteq \mathcal{L}(AS)$. $\mathcal{L}(G) \subseteq h(\mathcal{L}(G'))$ and $h(\mathcal{L}(G')) \subseteq \mathcal{L}(AS)$ imply $\mathcal{L}(G) \subseteq \mathcal{L}(AS)$.

□

Hence $R_{[n]} \subseteq PS\ AS\ FSA(n)$, where $R_{[n]}$ denotes the family (class) of $n - RLSMG$.

5 CONCLUSION

We introduced parallel finite systems communicating by transitions and we have showed in Theorem 4.1, which has been proven, that the class of languages accepted by n-parallel finite automata systems is a subset of class of languages generated by n-right linear simple matrix languages. And from [1] we already know, that these classes are equivalent and because classes of languages generated by $n - RLSMG$ forms a hierarchy, where $R_{[n]} \subset R_{[n+1]}$, then for the classes of languages accepted by $pc\ as\ fsa(n)$ holds that $PS\ AS\ FSA(n) \subset PS\ AS\ FSA(n+1)$.

Acknowledgement: This work was partially supported by the BUT FIT grant FIT-S-10-2 and the research plan MSM0021630528.

REFERENCES

[1] Petřík, P: Automata systems, Brno, 2007, Bachelor thesis, FIT BUT

[2] Krithivasan, K., Balan, S., Harsha, P.: Distributed Processing in Automata. International Journal of Foundations of Computer Science, 10(4):443-463, December 1999

[3] Wood, D.: m-parallel n-right linear simple matrix languages. Util. Math., 8:3-28, 1975

[4] Ibarra, O. H.: Simple Matrix Languages. Information and Control, 17(4):359-394, 1970

[5] Dassow, J., Păun, G., Rozenberg, G.: Grammar systems. Handbook of formal languages, pages 155-213, 1997

[6] Martín-Vide, C., Mateescu, A., Mitrana, V.: Parallel Finite Automata Systems Communicating by States. Int. J. Found. Comput. Sci., 13(5):733-749, 2002

[7] Meduna, A.: Automata and Languages: Theory and Applications. Springer Verlag, 2005