LINEAR N-PARALLEL AUTOMATA

Rudolf SCHÖNECKER, Master Degree Programme (5)
Dept. of Information Systems, FIT, BUT
E-mail: xschon01@stud.fit.vutbr.cz

Supervised by: Doc. RNDr. Alexander Meduna, CSc.

ABSTRACT

A linear n-parallel automata represent a simple formal model of parallel abstract machines computing. These automata accept instances of languages with respect to the analogy of restricted parallelism in the linear n-parallel grammars. Under their generative power they constitute an infinite hierarchy of languages which exceeds from bounds of the context free languages.

1 INTRODUCTION

The first mention about the restricted parallelism in grammars is dated to early 1970s. R. D. Rosebrugh and D. Wood‘s article [1] summarizes the basic aspects of the parallel computing, necessary steps by the modification of the current system which is represented by the classical sequential grammars. They have designed a model of the n-parallel linear grammars (n-PLG) which comes out of the simple principle – an application of the n-rewriting grammar rules to the n-available non-terminals at the same time.

The partial language linearity is the only condition which ensures uniqueness by the application of the rewriting rules. That article takes in studies about properties of the n-parallel linear languages (n-PLL) – their closure or structural properties and generative power of the n-PLLs family. It was proved that family n-PLLs constitute an infinite hierarchy of languages bounded by family of context sensitive languages.

In this paper, we complete current n-PLL study with new formal models of the regular and linear n-parallel automaton. We prove, for every n-PLG, it is possible to construct a linear n-parallel automaton, and also, for every linear n-parallel automaton, there exists a n-PLG. We show the class of the linear n-parallel automata accepts n-PLLs family.

Models of n-parallel automata can be used in these situations where we need to verify some context dependent sequences of data. The only limitation consists in nondeterministic specification of the start positions for n-reading headers. However, this could be resolved by using key words or other auxiliary programming techniques.

2 PRELIMINARIES

A n-parallel right linear grammar (n-PRLG) is a quintuple $G_S = (N, T, P, S, n)$, where N is a set of nonterminals, T is a set of terminal symbols, $S \in N$ is the start symbol and P is a set of rewriting rules of the form: $S \to x$, for $x \in T^*$, or $S \to X_1 \dots X_n$, for $X_i \in (N \setminus \{S\})$ where $1 \le i \le n$, $X \to y$, for $y \in T^*(N \setminus \{S\})$ and $X \in N \setminus \{S\}$, and $X \to x$, for $x \in T^+$, $X \in N$. Number $n \in Z$, $n > 0$, is used in the meaning of the strange of parallelism.

For $x, y \in (N \cup T)^*$ the relation of derivation $x \Rightarrow y$ is defined with these conditions: $x = S$ and $S \to y \in P$, or $x = x_1X_1 \dots x_nX_n$, $y = x_1y_1 \dots x_ny_n$, for $x_i \in T^*$, $1 \le i \le n$, $X_i \to y_i \in P$ and $y_i \to T^*(N \setminus \{S\})$, $1 \le i \le n$, or $x = x_1X_1 \dots x_nX_n$, $y = x_1y_1 \dots x_ny_n$, $x_i \in (N \cup T)^*$, $1 \le i \le n$, or $X_i \to y_i \in P$ and $y_i \to T^*$, $1 \le i \le n$.

For every n-PRLG $G_S = (N, T, P, S, n)$ there exists a $\overline{G_S} = (\overline{N}, T, \overline{P}, S, n)$ so that $L(G) = L(\overline{G})$, where $\overline{N} = \{S\} \cup N_1 \cup \dots \cup N_n$, $S \notin T \cup N_1 \cup \dots \cup N_n$ and N_i are mutually pair wise disjunctive. If $S \to X_1 \dots X_n \in \overline{P}$ then $X_i \in N_i$ where $1 \le i \le n$. If $X_i \to yY_j \in \overline{P}$ and $X_i \in N_i$, $Y_j \in N_j$, then $i = j$. Grammar $\overline{G}$ is known as a grammar in *a normal form* and we should write $\overline{G} = (N_1, \dots, N_n, T, S, \overline{P}, n)$.

3 DEFINITIONS

A nondeterministic regular n-parallel automaton (n-RA) is an one way initial automaton with n-headers for reading, specified as a quintuple $M = (Q, T, \delta, \delta_P, F, n)$, where $Q = (Q_1, \dots, Q_n)$ is an n-tuple of finite sets of internal states, T is a finite set of input symbols, δ is a transition function in the form $Q_i \times (T \cup \{\lambda\}) \to 2^{Q_i}$ for $\forall i \in \{1, \dots, n\}$, $\delta_P = \{x | x \in (Q_1 \times \dots \times Q_n)\}$ is a set of initial n-tuples of states for parallel computation and $F = (F_1, \dots, F_n)$ is an n-tuple of end states sets, where $F_i \subseteq Q_i$ for $\forall i \in \{1, \dots, n\}$.

For this automaton, we define a configuration $(q, w) \in (Q_{E1}, \dots, Q_{En}) \times T^*$, where $Q_{Ei} = Q_i \cup \{\lambda\}$ for $\forall i \in \{1, \dots, n\}$. The pair $((\lambda, \dots, \lambda), w)$ represents the start configuration, $((q_1, \dots, q_n), \lambda)$ for $q_i \in F_i$ the final configuration.

On the set of configurations, we define *a simple move*: $((\lambda, \dots, \lambda), w) \Rightarrow ((S_1, \dots, S_n), w)$, where $\exists (S_1, \dots, S_n) \in \delta_P$, in the starting case, else $((q_1, \dots, q_n), \underline{a_1}v_1\underline{a_2}v_2 \dots \underline{a_n}v_n) \Rightarrow ((p_1, \dots, p_n), v_1v_2 \dots v_n)$, for $p_i, q_i \in Q_i$, $a_i \in T \cup \{\lambda\}$ and $v_i \in T^*$, only when $p_i \in \delta(q_i, a)$ for $1 \le in \le$. Underline specifies the symbols where are actually placed reading heads. We define *a nondeterministic linear n-parallel automaton* (n-LA) with the extension of the transition function to $Q_i \times (T^* \cup \{\lambda\}) \to 2^{Q_i}$ and the a_i's domain to $a_i \in T^*$.

Relation $\Rightarrow$ can be extended to its transitive closure $\Rightarrow^+$ and transitive and reflexive closure $\Rightarrow^*$.

The language accepted by an n-RA $M = (Q, T, \delta, \delta_P, F, n)$, for $w \in T^*$, $q = (q_1, \dots, q_n) \in F$, is $L(M) = \{w | ((\lambda, \dots, \lambda), w) \Rightarrow^* ((q_1, \dots, q_n), \lambda)\}$.

Example 1. The n-LA $M_1 = (\{X_1, \dots, X_4, F_1, \dots, F_4, X_3'\}, \{a, b\}, \delta, \delta_P, \{F_1, \dots, F_4\}, 4)$, where $\delta(X_1, a) = \{X_1, F_1\}$, $\delta(X_2, b) = \{X_2, F_2\}$, $\delta(X_3, aa) = \{X_3, X_3'\}$, $\delta(X_4, b) = \{X_4, F_4\}$, $\delta(X_3', a) = \{F_3\}$, $\delta_P = \{(X_1, \dots, X_4), (F_1, F_2, X_3', F_4)\}$, accepts the language $L_1 = a^nb^na^{2n+1}b^n$.

When M accepts the sentence “aabbaaaaabb”, it goes through this configurations: $((\lambda, \dots, \lambda), \underline{a}a\underline{b}b\underline{a}aaaa\underline{b}b)$, $((X_1, \dots, X_4), \underline{a}a\underline{b}b\underline{a}aaaa\underline{b}b)$, $((X_1, \dots, X_4), \underline{a}\,\underline{b}\,\underline{a}aa\underline{b})$, $((F_1, F_2, X_3', F_4), \underline{\lambda}\underline{\lambda}\underline{a}\underline{\lambda})$, $((F_1, F_2, F_3, F_4), \underline{\lambda}\underline{\lambda}\underline{\lambda}\underline{\lambda})$.

4 RESULTS

Theorem 1. *Let $G = (N_1, \dots, N_n, T, S, P, n)$ is an n-PRLG in a normal form. For every n-PRLG G, there exists an equivalent n-RA M so that $L(G) = L(M)$.*

At first, we must prove that for every n-PRLG G, defined above, there exists an automaton M so that $L(G) = L(M)$, and in to opposite, for every n-RA automaton M, there exists an n-PRLG G so that $L(M) = L(G)$. For interest, we sketch the first part of the proof:

Proof 1a.(Sketch) The proof will be done by construction an n-RA automaton for a general n-PRLG $G = (N_1, \dots, N_n, T, S, P, n)$ in a normal form. Let $M = (Q, T, \delta, \delta_P, F, n)$, where $Q = (Q_1, \dots, Q_n)$, $F = (F_1, \dots, F_n)$, $\{F_i\} \cup N_i \in Q_i$, $F_i \notin N_1 \cup \dots \cup N_n$ for $\forall i \in \{1, \dots, n\}$. The sets δ and δ_P are defined in this way: For $S \to x \in P$ where $x = x_1 x_2 \dots x_P$, $x_i \in T$, it will be $(S, F_2, \dots, F_n) \in \delta_P$, $S_i \in Q_1$ for $\forall i \in \{1, \dots, n\}$ and for $\forall i \in \{1, \dots, p-1\}$: $F_i \in \delta(S_p, x_p)$. If $x = \lambda$ then must $(F_1, \dots, F_n) \in \delta_P$. For $S \to X_1 \dots X_n \in P$ where $X_i \in N_i$, $1 \le i \le n$, it will be $(X_1, \dots, X_n) \in \delta_P$. For $X \to x \in P$, $x = x_1 x_2 \dots x_p$, $X \in N_j$, it will be $Z_i \in Q_j$ for $\forall i \in \{1, \dots, p-1\}$, $p \ge 1$. If $p = 1$ then $F_j \in \delta(X, x_1)$, else if $p > 1$ then $Z_1 \in \delta(X, x_1)$, $F_j \in \delta(Z_{p-1}, x_p)$ and for $\forall i \in \{1, \dots, p-2\}$: $Z_{i+1} \in \delta(Z_i, x_{i+1})$. For $X \to y \in P$, where $y = y_1 y_2 \dots y_p Y$, $X, Y \in N_j$, there will be $Z_i \in Q_j$ for $\forall i \in \{1, \dots, p-1\}$. For $p = 0$: $Y \in \delta(X, \lambda)$. If $p = 1$ then $Y \in \delta(X, y_1)$ and if $p > 1$ then $Z_1 \in \delta(X, y_1)$, $Y \in \delta(Z_{p-1}, y_p)$, and for $\forall i \in \{1, \dots, p-2\}$: $Z_{i+1} \in \delta(Z_i, y_{i+1})$. Proof 1a continues with proving $L(G) \subseteq L(M)$ and $L(M) \subseteq L(G)$.

Proof 1b.(Sketch) For an n-RA $M = (Q, T, \delta, \delta_P, F, n)$ we construct an n-PRLG $G = (N_1, \dots, N_n, T, S, P, n)$ in a normal form: If $Q = (Q_1, \dots, Q_n)$ then $N_i = Q_i$ for $\forall i \in \{1, \dots, n\}$ If there exists a sequence of moves $((\lambda, \dots, \lambda), x) \Rightarrow ((S_1, F_2, \dots, F_n), x) \Rightarrow^m ((F_1, \dots, F_n), \lambda)$, for a string $x \in T^*$ of the length m, we add $S \longrightarrow x$ into P. For all n-tuples in the form $(X_1, \dots, X_n) \in \delta_P$ we add $S \to X_1, \dots, X_n$ into P. For every nonterminal X from G and every terminal symbol $a \in T$ we add $X \to aY$ into P when $Y \in \delta(X, a)$ and $Y \notin F_1 \cup \dots \cup F_n$, respective $X \to a$ when $Y \in \delta(X, a)$ and $Y \in F_1 \cup \dots \cup F_n$. It remains to prove $L(G) \subseteq L(M)$ and $L(M) \subseteq L(G)$ again.

Theorem 2. *For every nondeterministic regular n-parallel automaton there exists a nondeterministic linear n-parallel automaton.*

This fact arises out of regular and linear languages have the same generative capacity so that for every regular grammar exists an ekvivalent linear grammar.

REFERENCES

[1] Rosebrugh, R.D., Wood, D.: Restricted parallelism and right linear grammars, Computer Science, 1972

[2] Molnár, L'., Češka, M., Melichar, M.: Gramatiky a jazyky, Alfa SNTL Bratislava 1987

[3] Meduna, A.: Automata and Languages: Theory and Applications, London, GB, Springer, 2000, s. 892, ISBN 1-85233-074-0