MULTILANGUAGES AND MULTIACCEPTING AUTOMATA SYSTEM

Martin Čermák
Doctoral Degree Programme (2), FIT BUT
E-mail: xcerma16@stud.fit.vutbr.cz

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

ABSTRACT

This paper introduces a new area of modern theoretical computer science and deals about multilanguages processing by parallel automata system based on push-down automata.

1 INTRODUCTION

Theoretical computer science contains many formal models which describe different families of languages. This paper introduces a new model of automata system processes multilanguages instead of languages. This automata system may be useful in biomedicine or parallel compilers.

2 PRELIMINARIES

In this paper, we assume that the reader is familiar with formal language theory (see [3]).

For any integer n, $I(n) = \{1, 2, \ldots, n\}$. For a set, Q, $|Q|$ denotes the cardinality of Q. For an alphabet, V, V^* represents the free monoid generated by V. The identity of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$; algebraically, V^+ is thus the free semigroup generated by V. For $w \in V^*$, $|w|$ denotes the length of w.

A *pushdown automaton* is a septuple $M = (Q, \Sigma, \Omega, \delta, q_0, Z_0, F)$, where Q is a finite set of states, Σ is an alphabet, $q_0 \in Q$ is the initial state, Ω is a pushdown alphabet, δ is a finite set of rules of the form $Zqa \to \gamma p$, where $p, q \in Q$, $Z \in \Omega$, $a \in \Sigma \cup \{\varepsilon\}$, $\gamma \in \Omega^*$, $F \subseteq Q$ is a set of final states, and $Z_0 \in \Omega$ is the initial pushdown symbol. A configuration of M is any word from $\Omega^* Q \Sigma^*$. For any configuration $xAqay$, where $x \in \Omega^*$, $y \in \Sigma^*$, $q \in Q$, and any $Aqa \to \gamma p \in \delta$, M makes a move from configuration $xAqay$ to configuration $x\gamma py$ according to $Aqa \to \gamma p$, written as $xAqay \Rightarrow x\gamma py\,[Aqa \to \gamma p]$, or, simply, $xAqay \Rightarrow x\gamma py$. If $x, y \in \Omega^* Q \Sigma^*$ and $m > 0$, then $x \Rightarrow^m y$ if there exists a sequence $x_0 \Rightarrow x_1 \Rightarrow \cdots \Rightarrow x_m$, where $x_0 = x$ and $x_m = y$. Then, we say $x \Rightarrow^+ y$ if there exists $m > 0$ such that $x \Rightarrow^m y$, and $x \Rightarrow^* y$ if $x = y$ or $x \Rightarrow^+ y$. If $w \in \Sigma^*$ and $Z_0 q_0 w \Rightarrow^* f$, where $f \in F$, then w is accepted by M, and $Z_0 q_0 w \Rightarrow^* f$ is an acceptance of w in M. The language of M is defined as $\mathcal{L}(M) = \{w \in \Sigma^* : Z_0 q_0 w \Rightarrow^* f \text{ is an acceptance of } w\}$. Family of languages defined by pushdown automata is denoted by **CF**.

A *context-free grammar* is quadruple $G = (N, T, S, P)$, where N and T are disjoint alphabets of nonterminal and terminal symbols, respectively, $S \in T$ is the start symbol of G and P is finite set of rules of the form $A \to \alpha$, where $A \in N$ and $\alpha \in (N \cup T)^*$. Let $u, v \in (N \cup T)^*$, for all $A \to \alpha$, write $uAv \Rightarrow u\alpha v$. Let $\Rightarrow^*$ denote transitive and reflexive closure of $\Rightarrow$. The language of G is defined as $L(G) = \{\omega : S \Rightarrow^* \omega, \omega \in T^*\}$.

A *n–Multigenerative nonterminal–synchronized grammar system* (denoted by n–KGN) is (n+1)–tuple $\Gamma = (G_1, G_2, \ldots, G_n, Q)$, where $G_i = (N_i, T_i, P_i, S_i)$ is a context-free grammar for each $i = 1, \ldots, n$ and Q is finite set of n–tuples of the form $(A_1, \ldots, A_n)$, where $A_i \in N_i$ for all $i = 1, \ldots, n$. A sentential n–form of n–KGN is an n–tuple of the form $\chi = (x_1, \ldots, x_n)$, where $x_i \in (N \cup T)^*$ for all $i = 1, \ldots, n$. Let n–forms $\chi = (u_1A_1v_1, \ldots, u_nA_nv_n)$ and $\chi' = (u_1x_1v_1, \ldots, u_nx_nv_n)$ be two sentential forms, where $A_i \in N_i$, $u_i \in T^*$ and $v_i, x_i \in (N \cup T)^*$ for all $i = 1, \ldots, n$. Let $A_i \to x_i$ for all $i = 1, \ldots, n$ and $(A_1, \ldots, A_n) \in Q$. Then $\chi \Rightarrow \chi'$ and $\Rightarrow^*$ and $\Rightarrow^+$ are it's transitive-reflexive and transitive closure, respectively. The n-language of Γ is defined as $n\text{-}L(\Gamma) = \{(w_1, \ldots, w_n) : (S_1, \ldots, S_n) \Rightarrow^* (w_1, \ldots, w_n), w_i \in T_i^* \ \forall i = 1, 2, \ldots, n\}$. The language generated by Γ in the union mode is defined as $L_{union}(\Gamma) = \{w : (w_1, \ldots, w_n) \in n\text{-}L(\Gamma), w \in \{w_1, \ldots, w_n\}\}$, the language generated by Γ in the concatenation mode is defined as $L_{conc}(\Gamma) = \{w : (w_1, \ldots, w_n) \in n\text{-}L(\Gamma), w = w_1, \ldots, w_n\}$, the language generated by Γ in the first mode is defined as $L_{first}(\Gamma) = \{w_1 : (w_1, \ldots, w_n) \in n\text{-}L(\Gamma)\}$. For more information see [1] and [2]. Family of languages generated by n–Multigenerative nonterminal–synchronized grammar system is denoted by **RE**.

3 DEFINITIONS

3.1 *N*–ACCEPTING STATE–SYNCHRONIZING AUTOMATA SYSTEM

Let $I = I(n)$ for some $n \geq 1$. Let $\forall i \in I$, $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, s_i, z_{i,0}, F_i)$ is push–down automaton. Then n–Accepting, State–Synchronizing Automata System is defined as $\vartheta = (M_1, \ldots, M_n, \Psi, S)$ where Ψ is finite set of switch rules of the form $(q_1, \ldots, q_n) \to (\hbar_1, \ldots, \hbar_n)$, where $\forall i \in I$, $q_i \in Q_i$, $\hbar_i \in \{e, d\}$, e denote enable component of the automata system, d denote disable component of the automata system, S is n-tuple $(\hbar_1^0, \ldots, \hbar_n^0)$ and denotes starting activities of components in n-MAS.

3.2 *N*–CONFIGURATION OF *N*–MAS

Let $I = I(n)$ for some $n \geq 1$ and $\vartheta = (M_1, \ldots, M_n, \Psi, S)$ and $\forall i \in I$, $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, s_i, z_{i,0}, F_i)$ is n-MAS. Then n–configuration is defined as n-tuple $\chi = (x_1^{\hbar_1}, \ldots, x_n^{\hbar_n})$, where $\forall i \in I$: $x_i = (q_i z_i \omega_i) \in Q_i \Gamma_i^* \Sigma^*$, $\hbar_i \in \{d, e\}$, where index d_i and e_i denotes configuration of disabled and enabled component M_i in n-MAS, respectively, $\omega_i \in \Sigma^*$ denotes unreaded input string.

3.3 COMPUTING STEP IN *N*–MAS

Let $I = I(n)$ for some $n \geq 1$ and $\vartheta = (M_1, \ldots, M_n, \Psi, S)$ is n–MAS, $\forall i \in I$, $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, s_i, z_{i,0}, F_i)$. Let $\chi = ((q_1\gamma_1z_1a_1\omega_1)^{\hbar_1}, \ldots, (q_n\gamma_nz_na_n\omega_n)^{\hbar_n})$, $\chi' = ((q_1'\gamma_1'z_1'\omega_1')^{\hbar_1'}, \ldots, (q_n'\gamma_n'z_n'\omega_n')^{\hbar_n'})$, are two n-configurations, where $\forall i \in I$, $q_i, q_i' \in Q_i$; $\gamma_i', z_i, z_i' \in \Gamma_i^*$; $\gamma_i \in \Gamma \cup \{\varepsilon\}$; $\hbar_i, \hbar_i' \in \{e, d\}$, $\omega_i, \omega_i' \in \Sigma^*$, $a_i \in \Sigma \cup \{\varepsilon\}$, for i such that $\hbar_i = e$, $\gamma_i q_i a_i \to \gamma_i' q_i' \in \delta_i$, ϑ moves from n–configuration χ to χ', denoted $\chi \vdash \chi'$, where $\forall j \in I$, where $\hbar_j = d$, $q_j' = q_j$ and $\omega_j' = a_j\omega_j$, $\forall j \in I$, where $\hbar_j = e$, $q_j' \in Q_j$ a $\omega_j' = \omega_j$. If $(q_1', \ldots, q_n') \to (g_1, \ldots, g_n) \in \Psi$, where $g_k \in \{e, d\}$ for all $k \in I$, then $\hbar_k' = g_k$, If $\forall (g_1, \ldots, g_n) \in \overbrace{\{e, d\} \times \ldots \times \{e, d\}}^{n\times} : (q_1', \ldots, q_n') \to (g_1, \ldots, g_n) \notin \Psi$, then for all

$k \in I : h'_k = h_k$.

In the standard way, $\vdash^*$ and $\vdash^+$ denote transitive-reflexive and transitive closure of $\vdash$, respectively.

3.4 N–MAS N-LANGUAGES

Let $I = I(n)$ for some $n \geq 1$ and $\vartheta = (M_1, \ldots, M_n, \Psi, S)$ is n-MAS, where $\forall i \in I$, $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, s_i, z_{i,0}, F_i)$ is push-down automaton. $\chi_0 = ((q_1 z_1 \omega_1)^{h_1}, \ldots, (q_n z_n \omega_n)^{h_n})$ is the start and $\chi_f = ((q'_1 z'_1 \varepsilon)^{h'_1}, \ldots, (q'_n z'_n \varepsilon)^{h'_n})$ is a finish n–configuration of n-MAS. n–language of n-MAS is defined as n–$L(\vartheta) = \{(\omega_1, \ldots, \omega_n) | \chi_0 \vdash^* \chi_f; q'_j \in F_j \text{ for all } j \in I\}$. Futhermore, every $a \in n$–$L(\vartheta)$ is a multistring.

3.5 ALGORITHM OF N–KGN TO N–MAS CONVERTION

Let n–KGN $\widehat{\Gamma} = (G_1, \ldots, G_n, \widehat{Q})$, where $\forall i \in I(n)$, $G_i = (\widehat{N}_i, \widehat{T}_i, \widehat{P}_i, \widehat{S}_i)$ is context–free grammar and n–MAS $\vartheta = (M_1, \ldots, M_n, \Psi, S)$, where $\forall i \in I(n)$, $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, s_0^i, z_0^i, F_i)$ is push-down automaton accepting by final state and empty stack and n–$L(\vartheta) = n$-$L(\widehat{\Gamma})$, where $S = (l_1, \ldots, l_n) : \forall i \in I(n)$, $l_i = e$. Then,

- $\forall i \in I(n)$, G_i and $M_i = (Q_i, \Sigma, \Gamma_i, \delta_i, s_i, z_{i,0}, F_i)$:
  - set $Q_i = \{\langle A \rangle : A \in \widehat{N}_i\} \cup \{r_i, f_i\}$, $\Sigma = \bigcup_{i=1}^n \widehat{T}_i$, $\Gamma_i = \Sigma \cup \widehat{N}_i \cup \{A' : A \in \widehat{N}_i\} \cup \{\Delta, \Delta'\}$, $F_i = \{f_i\}$, $s_0^i = \langle \widehat{S}_i \rangle$, $z_0^i = \Delta'$,
  - δ_i contains rules of the form: 1) $\Delta' \langle \widehat{S}_i \rangle \to \Delta \widehat{S}_i \widehat{S}_i{}' \langle \widehat{S}_i \rangle$, 2) $a \langle A \rangle \to a r_i$: $a \in \widehat{T}_i$ and $A \in \widehat{N}_i$, 3) $A' \langle B \rangle \to A' r_i$: $A, B \in \widehat{N}_i$, 4) $a r_i a \to r_i$: $a \in \widehat{T}_i$, 5) $A' r_i \to \langle A \rangle$: $A \in \widehat{N}_i$, 6) $\Delta \langle A \rangle \to f_i$: $A \in \widehat{N}_i$, 7) $\Delta r_i \to f_i$, 8) $\forall (A \to \alpha) \in \widehat{P}_i$, $A \langle A \rangle \to \theta(\alpha) \langle A \rangle \in \delta_i$: θ is projection from $(\widehat{N}_i \cup \widehat{T}_i)^*$ to $(\{A' : A \in \widehat{N}_i \widehat{N}_i\} \cup \widehat{T}_i)^*$, such that $\theta(\omega) = \omega'$, where ω' is made from ω^R by replacing every $A \in \widehat{N}_i$ in ω^R by $A'A$.
- $\Psi : \forall (A_1, \ldots, A_n) \in \widehat{Q}$, $(\langle A_1 \rangle, \ldots, \langle A_n \rangle) \to (e, \ldots, e) \in \Psi$ $(f_1, \ldots, f_n) \to (e, \ldots, e) \in \Psi$ $\forall (q_1, \ldots, q_n) \in Q_1 \times Q_2 \times \ldots \times Q_n$, where $\{r_1, \ldots, r_n\} \cap \{q_1, \ldots, q_n\} \neq \emptyset$, $(q_1, \ldots, q_n) \to (l_1, \ldots, l_n) \in \Psi$ and $\forall o \in I(n) : q_o \in \{r_o\} \Leftrightarrow l_o = e$. For other $(q_1, \ldots, g_n) \in Q_1 \times Q_2 \times \ldots \times Q_n$, $(q_1, \ldots, q_n) \to (d, \ldots, d) \in \Psi$

3.6 THEOREM

Family of n-languages of n–KGN and family of n-languages of n–MAS are equivalent.

Proof of the Theorem 3.6
First, we prove that algorithm 3.5 is correct by following claims.
Claim A
Let $(\widehat{S}_1, \ldots, \widehat{S}_n) \Rightarrow^* (u_1 A_1 v_1, \ldots, u_n A_n v_n)$ in Γ, where $A_i \in \widehat{N}_i$, $v_i \in (\widehat{N}_i \cup \widehat{T}_i)^*$, $u_i \omega_i \in \widehat{T}_i^*$ $\forall i \in I(n)$ and $(u_1 A_1 v_1, \ldots, u_n A_n v_n) \Rightarrow^* (u_1 \omega_1, \ldots, u_n \omega_n)$. Then $((\Delta' \langle \widehat{S}_1 \rangle u_1 \omega_1)^e, \ldots, (\Delta' \langle \widehat{S}_n \rangle u_n \omega_n)^e) \vdash^* ((\Delta \theta(v_1) A_1 \langle \widehat{A_1} \rangle \omega_1)^{h_1}, \ldots, (\Delta \theta(v_n) A_n \langle \widehat{A_n} \rangle \omega_n)^{h_n})$ in ϑ.
Proof of the Claim A By induction on length of derivation.
Basis:
Let $(\widehat{S}_1, \ldots, \widehat{S}_n) \Rightarrow^0 (\widehat{S}_1, \ldots, \widehat{S}_n)$, where $(\widehat{S}_1, \ldots, \widehat{S}_n) \Rightarrow^* (\omega_1, \ldots, \omega_n)$ and $\forall i \in I(n)$ $\omega_i \in \widehat{T}_i^*$.

Then, $((\Delta'\langle\widehat{S_1}\rangle\omega_1)^e,\ldots,(\Delta'\langle\widehat{S_n}\rangle\omega_n)^e) \vdash ((\Delta\widehat{S_1}\widehat{S_1}'\langle\widehat{S_1}\rangle\omega_1)^e, \ldots,((\Delta\widehat{S_n}\widehat{S_n}'\langle\widehat{S_n}\rangle\omega_n)^e)$[by 1 of δ_i] $\vdash ((\Delta\widehat{S_1}\widehat{S_1}'r_1\omega_1)^e, \ldots,((\Delta\widehat{S_n}\widehat{S_n}'r_n\omega_n)^e)$[by 3 of δ_i] $\vdash ((\Delta\widehat{S_1}\langle\widehat{S_1}\rangle\omega_1)^e, \ldots,((\Delta\widehat{S_n}\langle\widehat{S_n}\rangle\omega_n)^e)$[by 5 of δ_i].

Induction Hypothesis:

Suppose that Claim A holds for j and fewer derivations steps.

Induction Step:

Consider any derivation of the form $(\widehat{S_1},\ldots,\widehat{S_n}) \Rightarrow^{j+1} (u_1x_1v_1,\ldots, u_nx_nv_n)$, where $\forall i \in I(n)$, $u_ix_iv_i = u_iu_i'B_iv_i'$, $u_i,u_i' \in \widehat{T_i}^*$, $B_i \in \widehat{N_i}$, $v_i,v_i',x_i \in (\widehat{N_i}\cup\widehat{T_i})^*$, for $u_i\omega_i \in \widehat{T_i}^*$ $(u_1u_1'B_1v_1'v_1, \ldots, u_nu_n'B_nv_n'v_n) \Rightarrow^* (u_1\omega_1,\ldots, u_n\omega_n)$. This derivation can be expressed as $(\widehat{S_1},\ldots,\widehat{S_n}) \Rightarrow^j (u_1A_1v_1, \ldots, u_nA_nv_n) \Rightarrow (u_1x_1v_1,\ldots, u_nx_nv_n)$, where $\forall i \in I(n)$ $A_i \in \widehat{N_i}$. By induction hypothesis $((\Delta'\langle\widehat{S_1}\rangle u_1\omega_1)^e,\ldots,(\Delta'\langle\widehat{S_n}\rangle u_n\omega_n)^e) \vdash^* ((\Delta\theta(v_1)\ A_1\langle A_1\rangle\omega_1)^{\hbar_1}, \ldots,((\Delta\theta(v_n)A_n\langle A_n\rangle\omega_n)^{\hbar_n})$. From definition of n–KGN, $(A_1,\ldots,A_n) \in \widehat{Q}$ and from algorithm it is obvious that $(\langle A_1\rangle,\ldots,\langle A_n\rangle) \to (e,\ldots,e) \in \Psi$ and for every $A_i \to x_i \in \widehat{P_i}$ there is $A_i\langle A_i\rangle \to \theta(x_i)$. Therefore $\hbar_1,\ldots,\hbar_n = e$ and $((\Delta\theta(v_1)A_1\langle A_1\rangle\omega_1)^e, \ldots,((\Delta\theta(v_n)A_n\langle A_n\rangle\omega_n)^e) \vdash ((\Delta\theta(v_1)(x_1)^R\langle A_1\rangle\omega_1)^e, \ldots, ((\Delta\theta(v_1)(x_n)\langle A_n\rangle\omega_n)^e) = ((\Delta\theta(v_1')(u_1'B_1'B_1)^R\langle A_1\rangle\omega_1)^e, \ldots,((\Delta\theta\ (v_n')(u_n'B_n'B_n)^R\langle A_n\rangle\omega_n)^e)$. There is $a_i \in \Gamma_i - (\widehat{N_i}\cup\{\Delta,\Delta'\})$ on the top of the stack in each automaton. Hence, from rules of the form 2 and 3 in δ_i and from Ψ, $((\Delta\theta(v_1')(u_1'B_1'\ B_1)^R\langle A_1\rangle\omega_1)^e, \ldots,((\Delta\theta(v_n')(u_n'B_n'B_n)^R\langle A_n\rangle\omega_n)^e) \vdash ((\Delta\theta(v_1')(u_1'B_1'B_1)^Rr_1\omega_1)^e, \ldots,((\Delta\theta(v_n')(u_n'\ B_n'B_n)^Rr_n\omega_n)^e)$. Because $u_iu_i'B_iv_i' \Rightarrow^* u_i\omega_i$, $u_i\omega_i = u_iu_i'\omega_i'$. For every component M_i of ϑ, there is a sequence of moves $\Delta\theta(v_i')B_iB_i'(u_i')^Rr_iu_i'\omega_i' \vdash^* \Delta\theta(v_i')B_iB_i'r_i\omega_i'$ (by 4 of δ_i) and $\Delta\theta(v_i')B_iB_i'r_i\omega_i' \vdash \Delta\theta(v_i')B_i\langle B_i\rangle u_i'\omega_i'$ (by 5 of δ_i). From Ψ, it is obvious, that each automaton M_i which is in state $\langle B_i\rangle$ is blocked until any other automaton M_j is in the state r_j. Hence, $((\Delta\theta(v_1')(u_1'B_1'B_1)^Rr_1u_1'\omega_1')^e, \ldots,((\Delta\theta(v_n')(u_n'B_n'B_n)^Rr_nu_n'\omega_n')^e) \vdash^* ((\Delta\theta(v_1')B_1\langle B_1\rangle\omega_1')^{f_1}, \ldots, ((\Delta\theta(v_n')B_n\langle B_n\rangle\omega_n')^{f_n})$. The Claim A holds. □

Claim B

If $(\widehat{S_1},\ldots,\widehat{S_n}) \Rightarrow^* (\omega_1,\ldots,\omega_n)$ in $\widehat{\Gamma}$, where $\forall i \in I(n)$, $\omega_i \in \widehat{T_i}^*$, there is a sequence of movies $(\Delta'\langle\widehat{S_1}\rangle\omega_1)^e,\ldots,(\Delta'\langle\widehat{S_n}\rangle\omega_n)^e) \vdash^* ((f_1)^e,\ldots,(f_n)^e)$ in ϑ.

Proof of the Claim B

Consider any successful derivation $(\widehat{S_1},\ldots,\widehat{S_n}) \Rightarrow^* (\omega_1,\ldots,\omega_n)$. There must be multiform $(u_1A_1v_1,\ldots,u_nA_nv_n)$ such that $(\widehat{S_1},\ldots,\widehat{S_n}) \Rightarrow^* (u_1A_1v_1, \ldots, u_nA_nv_n) \Rightarrow (u_1x_1v_1, \ldots, u_nx_nv_n) = (\omega_1,\ldots,\omega_n)$. By the claim (A), $(\Delta'\langle\widehat{S_1}\rangle\omega_i)^e,\ldots,(\Delta'\langle\widehat{S_n}\rangle\omega_n)^e) \vdash^* (\Delta\theta(v_1)A_1\langle A_1\rangle\omega_1')^{\hbar_1},\ldots, (\Delta\theta(v_n)A_n\langle A_n\rangle\omega_n')^{\hbar_n}$ and $\forall i \in I(n)$, $\omega_i = u_i\omega_i'$. Because $(A_1,\ldots,A_n) \in \widehat{Q}$, $(\langle A_1\rangle,\ldots,\langle A_n\rangle) \to (e,\ldots, e) \in \Psi$, $\forall i \in I(n)$ $\hbar_i = e$ and ϑ moves to multiconfiguration $(\Delta\theta(x_1v_1)\langle A_1\rangle\omega_1')^e,\ldots,(\Delta\theta(x_nv_n)\langle A_n\rangle\omega_n')^e$. It is obvious that $\forall i \in I(n)$, $\omega_i' = x_iv_i$, because $\omega_i = u_ix_iv_i = u_i\omega_i'$. Therefore, there are two possibilities of the top symbol on the stack in each automaton of ϑ. Either, top symbol is Δ, than $x_iv_i = \varepsilon$ and $\omega_i = \varepsilon$, that is M_i moves to f_i. Furthermore, if any other automaton M_j is in the state r_j, automaton M_i is blocked. Or the top symbol is $a_i \in \widehat{T_i}$, than $\Delta(x_iv_i)^R\langle A_i\rangle x_iv_i \vdash \Delta(x_iv_i)^Rr_ix_iv_i$ (by rule of the form 2 of δ_i) and $\Delta(x_iv_i)^Rr_ix_iv_i \vdash^* \Delta r_i$ (by rules type 4 of δ_i), $\Delta r_i \vdash f_i$ (by 7) and automaton M_i is blocked until any other automaton M_j of ϑ is in the state r_j. Then ϑ is in the mutliconfiguration $((f_1)^e,\ldots,(f_n)^e)$ and the Claim (B) holds. □

Claim C

If $(\Delta'\langle\widehat{S_1}\rangle\omega_1)^e,\ldots,(\Delta'\langle\widehat{S_n}\rangle\omega_n)^e) \vdash^* ((f_1)^e,\ldots,(f_n)^e)$ in ϑ, there is a sequence of derivations $(\widehat{S_1},\ldots,\widehat{S_n}) \Rightarrow^* (\omega_1,\ldots,\omega_n)$.

Proof of the Claim C

Consider any successful acceptance (I) of the form $(\Delta'\langle\widehat{S_1}\rangle\omega_1)^e,\ldots,(\Delta'\langle\widehat{S_n}\rangle\omega_n)^e) \vdash^* ((f_1)^e, \ldots, (f_n)^e)$ in ϑ. From algorithm (by rules 1 of δ_i) it is obvious, that first step of ϑ must be

$(\Delta'\langle\widehat{S_1}\rangle\omega_1)^e,\ldots,(\Delta'\langle\widehat{S_n}\rangle\omega_n)^e) \vdash ((\Delta\widehat{S_1}\widehat{S_1}'\langle\widehat{S_1}\rangle\omega_1)^{l_1},\ \ldots,((\Delta\widehat{S_n}\widehat{S_n}'\langle\widehat{S_n}\rangle\omega_n)^{l_n})$ and from construction of Ψ there are two possibilities of activities of automata in ϑ: $(\langle\widehat{S_1}\rangle,\ldots,\langle\widehat{S_n}\rangle) \rightarrow (d,\ldots,d) \in \Psi$, but $\forall i \in I(n)$, $l_i = d$ and ϑ is not successful for any input. Therefore, $(\langle\widehat{S_1}\rangle,\ldots,\langle\widehat{S_n}\rangle) \rightarrow (e,\ldots,e) \in \Psi$ and $\forall i \in I(n)$, $l_i = e$. From rules of the form 3 and 5 in δ_i is clear that ϑ must do fallowing two steps: $((\Delta\widehat{S_1}\widehat{S_1}'\langle\widehat{S_1}\rangle\omega_1)^e,\ \ldots,((\Delta\widehat{S_n}\widehat{S_n}'\langle\widehat{S_n}\rangle\omega_n)^e) \vdash ((\Delta\widehat{S_1}\widehat{S_1}'r_1\omega_1)^e,\ \ldots,((\Delta\widehat{S_n}\widehat{S_n}' r_n\omega_n)^e) \vdash ((\Delta\widehat{S_1}\langle\widehat{S_1}\rangle\omega_1)^e,\ \ldots,((\Delta\widehat{S_n}\langle\widehat{S_n}\rangle\omega_n)^e)$ Consider any multiconfiguration $((\Delta A_1\langle A_1\rangle\omega_1')^e$, $\ldots$, $((\Delta A_n\langle A_n\rangle\omega_n')^e)$, where $((\Delta A_1\langle A_1\rangle\omega_1')^e,\ \ldots,((\Delta A_n\langle A_n\rangle\omega_n')^e) \vdash^* ((f_1)^e,\ldots,(f_n)^e)$. ϑ must do these steps: $((\Delta A_1\langle A_1\rangle\omega_1')^e,\ \ldots,((\Delta A_n\langle A_n\rangle\omega_n')^e) \vdash ((\Delta\gamma_1\langle A_1\rangle\omega_1')^e,\ \ldots,((\Delta\gamma_n\langle A_n\rangle\omega_n')^e) \vdash ((\gamma_1' q_1\omega_1')^{l_1'},\ \ldots,((\gamma_n' q_n\omega_n')^{l_n'})$, and (from 2, 3 and 6 of δ_i and from Ψ) for each $i \in I(n)$ holds: If $\gamma_i = \varepsilon$ then $\gamma_i' = \omega_i' = \varepsilon$, $q_i = f_i$ and $l_i' = \sigma$, where $\sigma \in \{e,d\}$, else $\gamma_i' = \Delta\gamma_i$, $q_i = r_i$ and $l_i' = e$. Furthermore, if $\gamma_1,\ldots,\gamma_n = \varepsilon$ (i.e. $\forall j \in I(n):\ q_j = f_j$) then $\sigma = e$. Otherwise $\sigma = d$. Now, there are only three applicable types of moves under each automaton M_i which is in the state r_i: $ar_ia \rightarrow r_i$: $a \in \widehat{T_i}$ and M_i still active in the next computation step, or $B_i' r_i \rightarrow \langle B_i\rangle$: $B_i \in \widehat{N_i}$, than next top symbol on the stack must be B_i and M_i is blocked until any automaton M_j of ϑ is in the state r_j, or $\Delta r_i \rightarrow f_i$ and M_i is blocked until any automaton M_j of ϑ is in the state r_j. These three types of steps are repeatedly applied on active components of ϑ until any automaton M_j is in the state r_j. Hence and from construction of Ψ, fallowing configuration of ϑ must be either $((f_1)^e,\ldots,(f_n)^e)$ (multistring is accepted), or $((\Delta\gamma_1' B_1\langle B_1\rangle\omega_1'')^e,\ \ldots,\ ((\Delta\gamma_n' B_n\langle B_n\rangle\omega_n'')^e)$ where $\forall i \in I(n)$: $B_i \in \widehat{N_i}$ and $(\langle B_1\rangle,\ldots,\langle B_n\rangle) \rightarrow (e,\ldots,e) \in \Psi$. For any other n-tuple of states, ϑ blocked all automata (by Ψ) and multistring is not accepted. It is obvious that we can express (I) as $(\Delta'\langle\widehat{S_1}\rangle\omega_1)^e,\ldots,(\Delta'\langle\widehat{S_n}\rangle\omega_n)^e) \vdash^3 ((\Delta\widehat{S_1}\langle\widehat{S_1}\rangle\omega_1)^e,\ \ldots,((\Delta\widehat{S_n}\langle\widehat{S_n}\rangle\omega_n)^e) \vdash^{m_1} ((\Delta\gamma_1^{(1)}A_1^{(1)}\langle A_1^{(1)}\rangle\omega_1^{(1)})^e,\ \ldots,((\Delta\gamma_n^{(1)}A_n^{(1)}\langle A_n^{(1)}\rangle\omega_n^{(1)})^e) \ldots \vdash^{m_k} ((\Delta\gamma_1^{(k)}A_1^{(k)}\langle A_1^{(k)}\rangle\omega_1^{(k)})^e,\ \ldots,((\Delta\gamma_n^{(k)}A_n^{(k)}\langle A_n^{(k)}\rangle\omega_n^{(k)})^e) \vdash^{m_{k+1}} ((f_1)^e,\ldots,(f_n)^e)$ for all $i \in I(k+1)$, $m_i \geq 1$. This computation of ϑ we can simulate by $\widehat{\Gamma}$ as $(\widehat{S_1},\ldots,\widehat{S_n}) \Rightarrow (u_1^{(1)}A_1^{(1)}v_1^{(1)},\ldots,$ $u_n^{(1)}A_n^{(1)}v_n^{(1)}) \Rightarrow \ldots \Rightarrow (u_1^{(k+1)},\ldots,u_n^{(k+1)}) = (\omega_1,\ldots,\omega_n)$ where $\forall i \in I(n), \forall j \in I(k+1)$: $u_i^{(j)} \in \widehat{T_i}^*$, $A_i^{(j)} \in \widehat{N_i}$, $v_i^{(j)} \in (\widehat{T_i} \cup \widehat{N_i})^*$ and $u_i^{(j)}\omega_i^{(j)} = \omega_i$. Hence, the Claim (C) holds. □

It is well known family of 2-KGN languges, generated by operations union, concatenation and the first under multilanguages (see [1, 2]), are equivalent with family of **RE** languages. Hence, from claims (A), (B) and (C), it is clear that $n\text{-}L(\vartheta) = n\text{-}L(\widehat{\Gamma})$ and the Theorem 3.6 holds. □

ACKNOWLEDGEMENTS

This work was partially supported by the BUT FIT grant FIT-S-10-2, the FRVŠ grant MŠMT 2010: FR2581/2010/G1 and the research plan MSM0021630528.

REFERENCES

[1] Meduna, A., Lukáš, R.: Multigenerative Grammar Systems, In: Schedae Informaticae, roč. 2006, č. 15, Krakov, PL, s. 175-188, ISSN 0860-0295

[2] Lukáš, R.: Power of Multigenerative Grammar Systems, In: Proceedings of the 12th Conference, Brno, CZ, 2006, s. 380-384, ISBN 80-214-3163-6

[3] Meduna, A.: Automata and languages :theory and applications /London :Springer,2000. xv, 916 s. ISBN 1-85233-074-0