Self-Regulating Finite Automata

Alexander Meduna and Tomáš Masopust

Department of Information Systems, Faculty of Information Technology,
Brno University of Technology, Božetěchova 2, Brno 61266, Czech Republic
meduna@fit.vutbr.cz, masopust@fit.vutbr.cz

Abstract: This paper introduces and discusses *self-regulating finite automata*. In essence, these automata regulate the use of their rules by a sequence of rules applied during previous moves. A special attention is paid to *turns* defined as moves during which a self-regulating finite automaton starts a new self-regulating sequence of moves. Based on the number of turns, the present paper establishes two infinite hierarchies of language families resulting from two variants of these automata. In addition, it demonstrates that these hierarchies coincide with the hierarchies resulting from parallel right linear grammars and right linear simple matrix grammars, so the self-regulating finite automata can be viewed as the automaton counterparts to these grammars. Finally, this paper compares both infinite hierarchies. In addition, as an open problem area, it suggests the discussion of self-regulating pushdown automata and points out that they give rise to no infinite hierarchy analogical to the achieved hierarchies resulting from the self-regulating finite automata.

Keywords: regulated automata, self-regulation, infinite hierarchies of language families, parallel right linear grammars, right linear simple matrix grammars

2000 Mathematics Subject Classification Number: 68Q45.

1 Introduction

Over its history, automata theory has modified and restricted classical automata in many ways (see [3, 5, 6, 7, 8, 16, 22, 24, 26]). Recently, regulated automata have been introduced and studied in [17, 18]. In essence, these automata regulate the use of their rules according to which they make moves by control languages. In this paper, we continue with this topic by defining and investigating *self-regulating finite automata*. Instead of prescribed control languages, however, the self-regulating finite automata restrict the selection of a rule according to which the current move is made by a rule according to which a previous move was made.

To give a more precise insight into self-regulating automata, consider a finite automaton, M, with a finite binary relation, R, over M's rules. Furthermore, suppose that M makes a sequence of moves, ρ, that leads to the acceptance of a word, so ρ can be expressed as a concatenation of $n+1$ consecutive subsequences, $\rho = \rho_0\rho_1 \dots \rho_n$, $|\rho_i| = |\rho_j|$, $0 \leq i, j \leq n$, in which r_i^j denote the rule according to which the ith move in ρ_j is made, for all $0 \leq j \leq n$ and $1 \leq i \leq |\rho_j|$ (as usual, $|\rho_j|$ denotes the length of ρ_j). If for all $0 \leq j < n$, $(r_1^j, r_1^{j+1}) \in R$, then M represents an *n-turn first-move self-regulating finite automaton with respect to R*. If for all $0 \leq j < n$ and all $1 \leq i \leq |\rho_i|$, $(r_i^j, r_i^{j+1}) \in R$, then M represents an *n-turn all-move self-regulating finite automaton with respect to R*.

Based on the number of turns, we establish two infinite hierarchies of language families that lie between the families of regular and context-sensitive languages. First, we demonstrate that

n-turn first-move self-regulating finite automata give rise to an infinite hierarchy of language families coinciding with the hierarchy resulting from $(n+1)$-parallel right linear grammars (see [20, 21, 27, 28]). Recall that n-parallel right linear grammars generate a proper language subfamily of the language family generated by $(n+1)$-parallel right linear grammars (see Theorem 5 in [21]). As a result, n-turn first-move self-regulating finite automata accept a proper language subfamily of the language family accepted by $(n + 1)$-turn first-move self-regulating finite automata, for all $n \geq 0$. Similarly, we prove that n-turn all-move self-regulating finite automata give rise to an infinite hierarchy of language families coinciding with the hierarchy resulting from $(n + 1)$-right linear simple matrix grammars (see [4, 10, 28]). As n-right linear simple matrix grammars generate a proper subfamily of the language family generated by $(n+1)$-right linear simple matrix grammars (see Theorem 1.5.4 in [4]), n-turn all-move self-regulating finite automata accept a proper language subfamily of the language family accepted by $(n + 1)$-turn all-move self-regulating finite automata. Furthermore, since the families of right linear simple matrix languages coincide with the language families accepted by multitape nonwriting automata (see [5]) and by finite-turn checking automata (see [24]), the all-move self-regulating finite automata characterize these families, too. Finally, we summarize the results about both infinite hierarchies.

In the conclusion of this paper, as an open problem area, we suggest the discussion of *self-regulating pushdown automata*. Regarding self-regulating all-move pushdown automata, we prove that they do not give rise to any infinite hierarchy analogical to the achieved hierarchies resulting from the self-regulating finite automata. Indeed, zero-turn all-move self-regulating pushdown automata define the family of context-free languages while one-turn all-move self-regulating pushdown automata define the family of recursively enumerable languages. On the other hand, as far as self-regulating first-move pushdown automata are concerned, the question whether they define an infinite hierarchy or not is open.

2 Preliminaries

We assume that the reader is familiar with the theory of automata and formal languages (see [1, 2, 9, 11, 12, 13, 15, 19, 25]). For a set Q, $|Q|$ denotes the cardinality of Q. $\mathbb{N} = \{1, 2, 3, \ldots\}$ denotes the set of all natural numbers. 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 $V^+ = V^* - \{\varepsilon\}$; algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. For $w \in V^*$, $|w|$ denotes the length of w. Let $w \in V^*$; then, $alph(w) = \{a \in V : a \text{ appears in } w\}$. For every $L \subseteq V^*$, $alph(L) = \bigcup_{w \in L} alph(w)$.

A *finite automaton*, M, is a quintuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is a finite input alphabet, δ is a finite set of rules of the form $qw \to p$, $q, p \in Q$, $w \in \Sigma^*$, $q_0 \in Q$ is an initial state, and F is a set of final states. Let Ψ be an alphabet of *rule labels* such that $|\Psi| = |\delta|$, and ψ be a bijection from δ to Ψ. For simplicity, to express that ψ maps a rule $qw \to p \in \delta$ to r, where $r \in \Psi$, we write $r.qw \to p \in \delta$; in other words, $r.qw \to p$ means $\psi(qw \to p) = r$. A *configuration* of M is any word from $Q\Sigma^*$. For any configuration qwy, where $y \in \Sigma^*$, $q \in Q$, and any $r.qw \to p \in \delta$, M makes a move from configuration qwy to configuration py according to r, written as $qwy \Rightarrow py\,[r]$. Let χ be any configuration of M. M makes zero moves from χ to χ according to ε, written as $\chi \Rightarrow^0 \chi\,[\varepsilon]$. Let there exist a sequence of configurations $\chi_0, \chi_1, \ldots, \chi_n$, for some $n \geq 1$, such that $\chi_{i-1} \Rightarrow \chi_i\,[r_i]$, where $r_i \in \Psi$, $i = 1, \ldots, n$. Then, M makes n moves from χ_0 to χ_n according to $r_1, \ldots, r_n$, symbolically written as $\chi_0 \Rightarrow^n \chi_n\,[r_1 \ldots r_n]$. We write $\varphi \Rightarrow^* \kappa\,[\mu]$ if $\varphi \Rightarrow^n \kappa\,[\mu]$ for some $n \geq 0$. If $w \in \Sigma^*$

and $q_0 w \Rightarrow^* f\,[\mu]$, for $f \in F$, then w *is accepted by* M and $q_0 w \Rightarrow^* f\,[\mu]$ is an *acceptance* of w in M. The *language* of M is defined as $\mathcal{L}(M) = \{w \in \Sigma^* : q_0 w \Rightarrow^* f\,[\mu]$ is an acceptance of $w\}$.

For $n \geq 1$, an n*-parallel right linear grammar*, n-PRLG, is an $(n+3)$-tuple $G = (N_1, \ldots, N_n, T, S, P)$, where N_i, $1 \leq i \leq n$, are mutually disjoint nonterminal alphabets, T is a terminal alphabet, $S \notin N$ is an initial symbol, $N = N_1 \cup \ldots \cup N_n$, and P is a finite set of rules that contains three kinds of rules:

1. $S \to X_1 \ldots X_n$, $\quad X_i \in N_i$, $1 \leq i \leq n$;
2. $X \to wY$, $\quad X, Y \in N_i$ for some i, $1 \leq i \leq n$, $w \in T^*$;
3. $X \to w$, $\quad X \in N$, $w \in T^*$.

For $x, y \in (N \cup T \cup \{S\})^*$, $x \Rightarrow y$ if and only if

1. either $x = S$ and $S \to y \in P$,
2. or $x = y_1 X_1 \ldots y_n X_n$, $y = y_1 x_1 \ldots y_n x_n$, where $y_i \in T^*$, $x_i \in T^* N \cup T^*$, $X_i \in N_i$, and $X_i \to x_i \in P$, $1 \leq i \leq n$.

If $x, y \in (N \cup T \cup \{S\})^*$ and $l > 0$, then $x \Rightarrow^l y$ if and only if there exists a sequence $x_0 \Rightarrow x_1 \Rightarrow \ldots \Rightarrow x_l$, $x_0 = x$, $x_l = y$. Then, we say $x \Rightarrow^+ y$ if and only if there exists $l > 0$ such that $x \Rightarrow^l y$, and $x \Rightarrow^* y$ if and only if $x = y$ or $x \Rightarrow^+ y$. The language generated by an n-PRLG, G, is defined as $\mathcal{L}(G) = \{w \in T^* : S \Rightarrow^+ w\}$. Language $L \subseteq T^*$ is an n*-parallel right linear language*, n-PRLL, if there is an n-PRLG, G, such that $L = \mathcal{L}(G)$. The family of n-PRLLs is denoted by R_n.

For $n \geq 1$, an n*-right linear simple matrix grammar*, n-RLSMG, is an $(n+3)$-tuple $G = (N_1, \ldots, N_n, T, S, P)$, where N_i, $1 \leq i \leq n$, are mutually disjoint nonterminal alphabets, T is a terminal alphabet, $S \notin N$ is an initial symbol, $N = N_1 \cup \ldots \cup N_n$, and P is a finite set of matrix rules. A matrix rule can be in one of the following three forms:

1. $[S \to X_1 \ldots X_n]$, $\quad X_i \in N_i$, $1 \leq i \leq n$;
2. $[X_1 \to w_1 Y_1, \ldots, X_n \to w_n Y_n]$, $\quad w_i \in T^*$, $X_i, Y_i \in N_i$, $1 \leq i \leq n$;
3. $[X_1 \to w_1, \ldots, X_n \to w_n]$, $\quad X_i \in N_i$, $w_i \in T^*$, $1 \leq i \leq n$.

Let m be a matrix, then $m[i]$ denotes the ith rule of m. For $x, y \in (N \cup T \cup \{S\})^*$, $x \Rightarrow y$ if and only if

1. either $x = S$ and $[S \to y] \in P$,
2. or $x = y_1 X_1 \ldots y_n X_n$, $y = y_1 x_1 \ldots y_n x_n$, where $y_i \in T^*$, $x_i \in T^* N \cup T^*$, $X_i \in N_i$, $1 \leq i \leq n$, and $[X_1 \to x_1, \ldots, X_n \to x_n] \in P$.

We define $x \Rightarrow^+ y$ and $x \Rightarrow^* y$ as above. The language generated by an n-RLSMG, G, is defined as $\mathcal{L}(G) = \{w \in T^* : S \Rightarrow^+ w\}$. Language $L \subseteq T^*$ is an n*-right linear simple matrix language*, n-RLSML, if there is an n-RLSMG, G, such that $L = \mathcal{L}(G)$. The family of n-RLSMLs is denoted by $R_{[n]}$.

Let $G = (N_1, \ldots, N_n, T, S, P)$ be an n-PRLG, for some $n \geq 1$, and $1 \leq i \leq n$. By the *ith component of* G we understand a 1-PRLG $G = (N_i, T, S', P')$, where P' contains rules of the following forms:

1. $S' \to X_i$ $\quad$ if $S \to X_1 \ldots X_n \in P$, $X_i \in N_i$;
2. $X \to wY$ $\quad$ if $X \to wY \in P$ and $X, Y \in N_i$;
3. $X \to w$ $\quad$ if $X \to w \in P$ and $X \in N_i$.

The ith component of an n-RLSMG is defined analogously.

Finally, let REG, CF, and CS denote the families of regular, context-free, and context-sensitive languages, respectively.

3 Definitions and Examples

In this section, we define and illustrate n-turn first-move self-regulating finite automata and n-turn all-move self-regulating finite automata.

Definition 1 A *self-regulating finite automaton*, SFA, M, is a septuple

$$M = (Q, \Sigma, \delta, q_0, q_t, F, R),$$

where

1. $(Q, \Sigma, \delta, q_0, F)$ is a finite automaton,
2. $q_t \in Q$ is a *turn state*, and
3. $R \subseteq \Psi \times \Psi$ is a finite relation on the alphabet of rule labels.

In this paper, we consider two ways of self-regulation—first-move and all-move. According to these two types of self-regulation, two types of n-turn self-regulating finite automata are defined.

Definition 2 Let $n \geq 0$ and $M = (Q, \Sigma, \delta, q_0, q_t, F, R)$ be a self-regulating finite automaton. M is said to be an *n-turn first-move self-regulating finite automaton*, n-first-SFA, if M accepts w in the following way. There is an acceptance of the form $q_0 w \Rightarrow^* f\,[\mu]$ such that

$$\mu = r_1^0 \dots r_k^0 r_1^1 \dots r_k^1 \dots r_1^n \dots r_k^n,$$

where $k \in \mathbb{N}$, r_k^0 is the first rule of the form $qx \rightarrow q_t$, for some $q \in Q$, $x \in \Sigma^*$, and

$$(r_1^j, r_1^{j+1}) \in R$$

for all $0 \leq j < n$.

The family of languages accepted by n-first-SFAs is denoted by W_n.

Example 1 Consider a 1-turn first-move self-regulating finite automaton, $M = (\{s, t, f\}, \{a, b\}, \delta, s, t, \{f\}, \{(1, 3)\})$, with δ containing rules $1.sa \rightarrow s$, $2.sa \rightarrow t$, $3.tb \rightarrow f$, and $4.fb \rightarrow f$ (see Fig. 1).

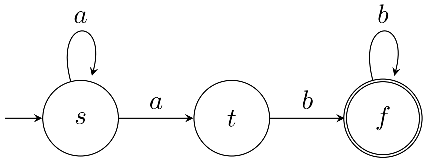

Fig. 1: 1-turn first-move self-regulating finite automaton M.

With $aabb$, M makes

$$saabb \Rightarrow sabb\,[1] \Rightarrow tbb\,[2] \Rightarrow fb\,[3] \Rightarrow f\,[4].$$

In brief, $saabb \Rightarrow^* f\,[1234]$. Observe that $\mathcal{L}(M) = \{a^n b^n : n \geq 1\}$, which belongs to $CF - REG$.

Definition 3 Let $n \geq 0$ and $M = (Q, \Sigma, \delta, q_0, q_t, F, R)$ be a self-regulating finite automaton. M is said to be an *n-turn all-move self-regulating finite automaton*, n-all-SFA, if M accepts w in the following way. There is an acceptance $q_0 w \Rightarrow^* f\,[\mu]$ such that

$$\mu = r_1^0 \ldots r_k^0 r_1^1 \ldots r_k^1 \ldots r_1^n \ldots r_k^n,$$

where $k \in \mathbb{N}$, r_k^0 is the first rule of the form $qx \to q_t$, for some $q \in Q$, $x \in \Sigma^*$, and

$$(r_i^j, r_i^{j+1}) \in R$$

for all $1 \leq i \leq k,\ 0 \leq j < n$.

The family of languages accepted by n-all-SFAs is denoted by S_n.

Example 2 Consider a 1-turn all-move self-regulating finite automaton, $M = (\{s, t, f\}, \{a, b\}, \delta, s, t, \{f\}, \{(1,4), (2,5), (3,6)\})$, with δ containing rules $1.sa \to s$, $2.sb \to s$, $3.s \to t$, $4.ta \to t$, $5.tb \to t$, and $6.t \to f$ (see Fig. 2).

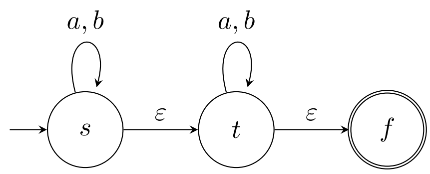

Fig. 2: 1-turn all-move self-regulating finite automaton M.

With $abab$, M makes

$$sabab \Rightarrow sbab\,[1] \Rightarrow sab\,[2] \Rightarrow tab\,[3] \Rightarrow tb\,[4] \Rightarrow t\,[5] \Rightarrow f\,[6].$$

In brief, $sabab \Rightarrow^* f\,[123456]$. Observe that $\mathcal{L}(M) = \{ww : w \in \{a, b\}^*\}$, which belongs to $CS - CF$.

4 Results

We prove that the family of languages accepted by n-first-SFAs coincides with the family of languages generated by $(n+1)$-PRLGs. Furthermore, we demonstrate that the family of languages accepted by n-all-SFAs coincides with the family of languages generated by n-RLSMGs.

4.1 n-Turn First-Move Self-Regulating Finite Automata

Section 4.1 establishes the identity between the family of languages accepted by n-first-SFAs and the family of languages generated by $(n+1)$-PRLGs. To do so, we need the following form of parallel right linear grammars.

Lemma 4 *For every n-PRLG $G = (N_1, \ldots, N_n, T, S, P)$, there is an equivalent n-PRLG $G' = (N_1', \ldots, N_n', T, S, P')$ that satisfies:*

1. *if $S \to X_1 \ldots X_n \in P'$, then X_i does not occur on the right-hand side of any rule, for $1 \leq i \leq n$;*
2. *if $S \to \alpha$, $S \to \beta \in P'$ and $\alpha \neq \beta$, then $alph(\alpha) \cap alph(\beta) = \emptyset$.*

Proof. If G does not satisfy conditions from the lemma, then we will construct a new n-PRLG $G' = (N'_1, \ldots, N'_n, T, S, P')$, where P' contains all rules of the form $X \to \beta \in P$, $X \neq S$, and $N_j \subseteq N'_j$, $1 \leq j \leq n$. For each rule $S \to X_1 \ldots X_n \in P$, we add new nonterminals $Y_j \notin N'_j$ into N'_j, and rules include $S \to Y_1 \ldots Y_n$ and $Y_j \to X_j$ in P', $1 \leq j \leq n$. Clearly,

$$S \Rightarrow_G X_1 \ldots X_n \text{ if and only if } S \Rightarrow_{G'} Y_1 \ldots Y_n \Rightarrow X_1 \ldots X_n.$$

Thus, $\mathcal{L}(G) = \mathcal{L}(G')$. □

Lemma 5 *Let G be an n-PRLG. Then, there is an $(n-1)$-first-SFA, M, such that $\mathcal{L}(G) = \mathcal{L}(M)$.*

Proof. Informally, M is divided into n parts (see Fig. 3). The ith part represents a finite automaton accepting the language of G's ith component, and R also connects the ith part to the $(i+1)$st part as depicted in Fig. 3.

Formally, without loss of generality, we assume $G = (N_1, \ldots, N_n, T, S, P)$ to be in the form from Lemma 4. We construct an $(n-1)$-first-SFA $M = (Q, T, \delta, q_0, q_t, F, R)$, where $Q = \{q_0, \ldots, q_n\} \cup N$, $N = N_1 \cup \ldots \cup N_n$, $\{q_0, q_1, \ldots, q_n\} \cap N = \emptyset$, $F = \{q_n\}$, $\delta = \{q_i \to X_{i+1} : S \to X_1 \ldots X_n \in P, 0 \leq i < n\} \cup \{Xw \to Y : X \to wY \in P\} \cup \{Xw \to q_i : X \to w \in P, w \in T^*, X \in N_i, i \in \{1, \ldots, n\}\}$, $q_t = q_1$, $\Psi = \delta$ with the identity map, and $R = \{(q_i \to X_{i+1}, q_{i+1} \to X_{i+2}) : S \to X_1 \ldots X_n \in P, 0 \leq i \leq n-2\}$.

Next, we prove $\mathcal{L}(G) = \mathcal{L}(M)$. To prove $\mathcal{L}(G) \subseteq \mathcal{L}(M)$, consider a derivation of w in G and construct an acceptance of w in M depicted in Fig. 3. This figure clearly demonstrates the

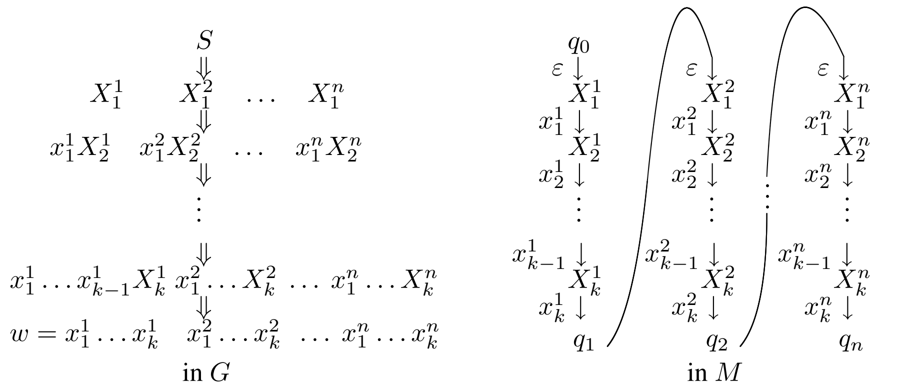

Fig. 3: A derivation of w in G and the corresponding acceptance of w in M.

fundamental idea behind this part of the proof; its complete and rigorous version is lengthy and left to the reader. Thus, for each derivation $S \Rightarrow^* w$, $w \in T^*$, there is an acceptance of w in M.

To prove $\mathcal{L}(M) \subseteq \mathcal{L}(G)$, let $w \in \mathcal{L}(M)$. Consider an acceptance of w in M. Observe that the acceptance is of the form depicted on the right-hand side of Fig. 3. It means that the number of steps M made from q_{i-1} to q_i is the same as from q_i to q_{i+1} since the only rule in the relation with $q_{i-1} \to X_1^i$ is the rule $q_i \to X_1^{i+1}$. Moreover, M can never come back to a state corresponding to a previous component. (By a component of M, we mean the finite automaton $M_i = (Q, \Sigma, \delta, q_{i-1}, \{q_i\})$, for $1 \leq i \leq n$.) Now, construct a derivation of w in G. By Lemma 4, we have $|\{X : (q_i \to X_1^{i+1}, q_{i+1} \to X) \in R\}| = 1$, for all $0 \leq i < n-1$. Thus, $S \to X_1^1 X_1^2 \ldots X_1^n \in P$. Moreover, if $X_j^i x_j^i \to X_{j+1}^i$, we apply $X_j^i \to x_j^i X_{j+1}^i \in P$, and if $X_k^i x_k^i \to q_i$, we apply $X_k^i \to x_k^i \in P$, $1 \leq i \leq n$, $1 \leq j < k$.

Hence, Lemma 5 holds. □

Lemma 6 *Let M be an n-first-SFA. There is an $(n+1)$-PRLG, G, such that $\mathcal{L}(G) = \mathcal{L}(M)$.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, q_t, F, R)$. Consider $G = (N_0, \dots, N_n, \Sigma, S, P)$, where $N_i = (Q\Sigma^l \times Q \times \{i\} \times Q) \cup (Q \times \{i\} \times Q)$, $l = \max\{|w| : qw \to p \in \delta\}$, $0 \le i \le n$, and

$P = \{S \to [q_0x_0, q^0, 0, q_t][q_tx_1, q^1, 1, q_{i_1}][q_{i_1}x_2, q^2, 2, q_{i_2}] \dots [q_{i_{n-1}}x_n, q^n, n, q_{i_n}] : r_0.q_0x_0 \to q^0, r_1.q_tx_1 \to q^1, r_2.q_{i_1}x_2 \to q^2, \dots, r_n.q_{i_{n-1}}x_n \to q^n \in \delta,$
$(r_0, r_1), (r_1, r_2), \dots, (r_{n-1}, r_n) \in R,\ q_{i_n} \in F\} \cup$
$\{[px, q, i, r] \to x[q, i, r]\} \cup$
$\{[q, i, q] \to \varepsilon : q \in Q\} \cup$
$\{[q, i, p] \to w[q', i, p] : qw \to q' \in \delta\}$.

Next, we prove $\mathcal{L}(G) = \mathcal{L}(M)$. To prove $\mathcal{L}(G) \subseteq \mathcal{L}(M)$, observe that we make $n+1$ copies of M and go through them similarly to Fig. 3. Consider a derivation of w in G. Then, in greater detail, this derivation is of the form

$$\begin{aligned} S &\Rightarrow [q_0x_0^0, q_1^0, 0, q_t][q_tx_0^1, q_1^1, 1, q_{i_1}] \dots [q_{i_{n-1}}x_0^n, q_1^n, n, q_{i_n}] \\ &\Rightarrow x_0^0[q_1^0, 0, q_t]x_0^1[q_1^1, 1, q_{i_1}] \dots x_0^n[q_1^n, n, q_{i_n}] \\ &\Rightarrow x_0^0x_1^0[q_2^0, 0, q_t]x_0^1x_1^1[q_2^1, 1, q_{i_1}] \dots x_0^nx_1^n[q_2^n, n, q_{i_n}] \\ &\ \vdots \\ &\Rightarrow x_0^0x_1^0 \dots x_k^0[q_t, 0, q_t]x_0^1x_1^1 \dots x_k^1[q_{i_1}, 1, q_{i_1}] \dots x_0^nx_1^n \dots x_k^n[q_{i_n}, n, q_{i_n}] \\ &\Rightarrow x_0^0x_1^0 \dots x_k^0x_0^1x_1^1 \dots x_k^1 \dots x_0^nx_1^n \dots x_k^n \end{aligned} \tag{1}$$

and $r_0.q_0x_0^0 \to q_1^0, r_1.q_tx_0^1 \to q_1^1, r_2.q_{i_1}x_0^2 \to q_1^2, \dots, r_n.q_{i_{n-1}}x_0^n \to q_1^n \in \delta$, $(r_0, r_1), (r_1, r_2), \dots, (r_{n-1}, r_n) \in R$, and $q_{i_n} \in F$.

Thus, the list of rules used in the acceptance of w in M is

$$\begin{aligned} \mu = &(q_0x_0^0 \to q_1^0)(q_1^0x_1^0 \to q_2^0) \dots (q_k^0x_k^0 \to q_t) \\ &(q_tx_0^1 \to q_1^1)(q_1^1x_1^1 \to q_2^1) \dots (q_k^1x_k^1 \to q_{i_1}) \\ &(q_{i_1}x_0^2 \to q_1^2)(q_1^2x_1^2 \to q_2^2) \dots (q_k^2x_k^2 \to q_{i_2}) \\ &\quad \vdots \\ &(q_{i_{n-1}}x_0^n \to q_1^n)(q_1^nx_1^n \to q_2^n) \dots (q_k^nx_k^n \to q_{i_n}). \end{aligned} \tag{2}$$

Now, we prove $\mathcal{L}(M) \subseteq \mathcal{L}(G)$. Informally, the acceptance is divided into $n+1$ parts of the same length. Grammar G generates the ith part by the ith component and records the state from which the next component starts.

Let μ be a list of rules used in an acceptance of $w = x_0^0x_1^0 \dots x_k^0x_0^1x_1^1 \dots x_k^1 \dots x_0^nx_1^n \dots x_k^n$ in M of the form (2). Then, the derivation of the form (1) is the corresponding derivation of w in G since $[q_j^i, i, p] \to x_j^i[q_{j+1}^i, i, p] \in P$ and $[q, i, q] \to \varepsilon$, for all $0 \le i \le n$, $1 \le j < k$.

Hence, Lemma 6 holds. □

The first main result of this paper follows next.

Theorem 7 *For all $n \ge 0$, $W_n = R_{n+1}$.*

Proof. This proof follows from Lemma 5 and 6. □

Corollary 8 *The following statements hold true.*

1. *$REG = W_0 \subset W_1 \subset W_2 \subset \dots \subset CS$.*

2. *$W_1 \subset CF$.*
3. *$W_2 \not\subseteq CF$.*
4. *$CF \not\subseteq W_n$ for any $n \geq 0$.*
5. *For all $n \geq 0$, W_n is closed under union, finite substitution, homomorphism, intersection with a regular language, and right quotient with a regular language.*
6. *For all $n \geq 1$, W_n is not closed under intersection and complement.*

Proof. Recall the following statements proved in [21]:

– $REG = R_1 \subset R_2 \subset R_3 \subset \ldots \subset CS$.
– $R_2 \subset CF$.
– $CF \not\subseteq R_n, n \geq 1$.
– For all $n \geq 1$, R_n is closed under union, finite substitution, homomorphism, intersection with a regular language, and right quotient with a regular language.
– For all $n \geq 2$, R_n is not closed under intersection and complement.

These statements and Theorem 7 imply statements 1, 2, 4, 5, 6 of Corollary 8. Moreover, observe that $\{a^n b^n c^{2n} : n \geq 0\} \in W_2 - CF$, which proves 3. □

Theorem 9 *For all $n \geq 1$, W_n is not closed under inverse homomorphism.*

Proof. For $n = 1$, let $L = \{a^k b^k : k \geq 1\}$, and let the homomorphism $h : \{a, b, c\}^* \to \{a, b\}^*$ be defined as $h(a) = a$, $h(b) = b$, and $h(c) = \varepsilon$. Then, $L \in W_1$, but

$$L' = h^{-1}(L) \cap c^* a^* b^* = \{c^* a^k b^k : k \geq 1\} \notin W_1.$$

Assume that L' is in W_1. Then, by Theorem 7, there is a 2-PRLG $G = (N_1, N_2, T, S, P)$ such that $\mathcal{L}(G) = L'$. Let $k > |P| \cdot \max\{|w| : X \to wY \in P\}$. Consider a derivation of $c^k a^k b^k \in L'$. The second component can generate only finitely many as; otherwise, it derives $\{a^k b^n : k < n\}$, which is not regular. Analogously, the first component generates only finitely many bs. Therefore, the first component generates any number of as, and the second component generates any number of bs. Moreover, there is a derivation of the form $X \Rightarrow^m X$, for some $X \in N_2$, and $m \geq 1$, used in the derivation in the second component. In the first component, there is a derivation $A \Rightarrow^l a^s A$, for some $A \in N_1$, and $s, l \geq 1$. Then, we can modify the derivation of $c^k a^k b^k$ so that in the first component, we repeat the cycle $A \Rightarrow^l a^s A$ $(m+1)$-times, and in the second component, we repeat the cycle $X \Rightarrow^m X$ $(l + 1)$-times. The derivations of both components have the same length—the added cycles are of length ml, and the rest is of the same length as in the derivation of $c^k a^k b^k$. Therefore, we have derived $c^k a^r b^k$, where $r > k$, which is not in L'—a contradiction.

For $n > 1$, the proof is analogous and left to the reader. □

Corollary 10 *For all $n \geq 1$, W_n is not closed under concatenation. Therefore, it is not closed under Kleene closure either.*

Proof. For $n = 1$, let $L_1 = \{c\}^*$ and $L_2 = \{a^k b^k : k \geq 1\}$. Then, $L_1 L_2 = \{c^* a^k b^k : k \geq 1\}$. Analogously, prove this corollary for $n > 1$. □

4.2 n-Turn All-Move Self-Regulating Finite Automata

This section discusses n-turn all-move self-regulating finite automata. It proves that the family of languages accepted by n-all-SFAs coincides with the family of languages generated by n-RLSMGs.

Lemma 11 *For every n-RLSMG, $G = (N_1, \ldots, N_n, T, S, P)$, there is an equivalent n-RLSMG, G', that satisfies:*

1. *if $[S \to X_1 \ldots X_n]$, then X_i does not occur on the right-hand side of any rule, $1 \leq i \leq n$;*
2. *if $[S \to \alpha]$, $[S \to \beta] \in P$ and $\alpha \neq \beta$, then $alph(\alpha) \cap alph(\beta) = \emptyset$;*
3. *for any two matrices $m_1, m_2 \in P$, if $m_1[i] = m_2[i]$, for some $1 \leq i \leq n$, then $m_1 = m_2$.*

Proof. The first two conditions can be proved analogously to Lemma 4. Suppose that there are matrices m and m' such that $m[i] = m'[i]$, for some $1 \leq i \leq n$. Let $m = [X_1 \to x_1, \ldots, X_n \to x_n]$, $m' = [Y_1 \to y_1, \ldots, Y_n \to y_n]$. Replace these matrices with matrices $m_1 = [X_1 \to X'_1, \ldots, X_n \to X'_n]$, $m_2 = [X'_1 \to x_1, \ldots, X'_n \to x_n]$, and $m'_1 = [Y_1 \to Y''_1, \ldots, Y_n \to Y''_n]$, $m'_2 = [Y''_1 \to y_1, \ldots, Y''_n \to y_n]$, where X'_i, Y''_i are new nonterminals for all i. These new matrices satisfy condition 3. Repeat this replacement until the resulting grammar satisfies the properties of G' given in this lemma. □

Lemma 12 *Let G be an n-RLSMG. There is an $(n-1)$-all-SFA, M, such that $\mathcal{L}(G) = \mathcal{L}(M)$.*

Proof. Without loss of generality, we assume that $G = (N_1, \ldots, N_n, T, S, P)$ is in the form described in Lemma 11. We construct $(n-1)$-all-SFA $M = (Q, T, \delta, q_0, q_t, F, R)$, where $Q = \{q_0, \ldots, q_n\} \cup N$, $N = N_1 \cup \ldots \cup N_n$, $\{q_0, q_1, \ldots, q_n\} \cap N = \emptyset$, $F = \{q_n\}$, $\delta = \{q_i \to X_{i+1} : [S \to X_1 \ldots X_n] \in P, 0 \leq i < n\} \cup \{X_i w_i \to Y_i : [X_1 \to w_1 Y_1, \ldots, X_n \to w_n Y_n] \in P, 1 \leq i \leq n\} \cup \{X_i w_i \to q_i : [X_1 \to w_1, \ldots, X_n \to w_n] \in P, w_i \in T^*, 1 \leq i \leq n\}$, $q_t = q_1$, $\Psi = \delta$ with the identity map, and $R = \{(q_i \to X_{i+1}, q_{i+1} \to X_{i+2}) : [S \to X_1 \ldots X_n] \in P, 0 \leq i \leq n-2\} \cup \{(X_i w_i \to Y_i, X_{i+1} w_{i+1} \to Y_{i+1}) : [X_1 \to w_1 Y_1, \ldots, X_n \to w_n Y_n] \in P, 1 \leq i < n\} \cup \{(X_i w_i \to q_i, X_{i+1} w_{i+1} \to q_{i+1}) : [X_1 \to w_1, \ldots, X_n \to w_n] \in P, w_i \in T^*, 1 \leq i < n\}$.

We next prove $\mathcal{L}(G) = \mathcal{L}(M)$. The proof of $\mathcal{L}(G) \subseteq \mathcal{L}(M)$ is very similar to the proof of the same inclusion of Lemma 5, so it is left to the reader.

To prove $\mathcal{L}(M) \subseteq \mathcal{L}(G)$, consider $w \in \mathcal{L}(M)$ and an acceptance of w in M. As in Lemma 5, the derivation looks like the one depicted on the right-hand side of Fig. 3. Next, we generate w in G as follows. By Lemma 11, there is matrix $[S \to X_1^1 X_1^2 \ldots X_1^n]$ in P. Moreover, if $X_j^i x_j^i \to X_{j+1}^i$, $1 \leq i \leq n$, then $(X_j^i \to x_j^i X_{j+1}^i, X_j^{i+1} \to x_j^{i+1} X_{j+1}^{i+1}) \in R$, for $1 \leq i < n$, $1 \leq j < k$. We apply $[X_j^1 \to x_j^1 X_{j+1}^1, \ldots, X_j^n \to x_j^n X_{j+1}^n]$ from P. If $X_k^i x_k^i \to q_i$, $1 \leq i \leq n$, then $(X_k^i \to x_k^i, X_k^{i+1} \to x_k^{i+1}) \in R$, for $1 \leq i < n$, and we apply $[X_k^1 \to x_k^1, \ldots, X_k^n \to x_k^n] \in P$. Thus, $w \in \mathcal{L}(G)$.

Hence, Lemma 12 holds. □

Lemma 13 *Let M be an n-all-SFA. There is an $(n+1)$-RLSMG, G, such that $\mathcal{L}(G) = \mathcal{L}(M)$.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, q_t, F, R)$. Consider $G = (N_0, \ldots, N_n, \Sigma, S, P)$, where $N_i = (Q\Sigma^l \times Q \times \{i\} \times Q) \cup (Q \times \{i\} \times Q)$, $l = \max\{|w| : qw \to p \in \delta\}$, $0 \leq i \leq n$, and

$$\begin{aligned}
P = \{&[S \to [q_0 x_0, q^0, 0, q_t][q_t x_1, q^1, 1, q_{i_1}] \ldots [q_{i_{n-1}} x_n, q^n, n, q_{i_n}]] : \\
&r_0.q_0 x_0 \to q^0, r_1.q_t x_1 \to q^1, \ldots, r_n.q_{i_{n-1}} x_n \to q^n \in \delta, \\
&(r_0, r_1), \ldots, (r_{n-1}, r_n) \in R, q_{i_n} \in F\} \cup \\
&\{[[p_0 x_0, q_0, 0, r_0] \to x_0[q_0, 0, r_0], \ldots, [p_n x_n, q_n, n, r_n] \to x_n[q_n, n, r_n]]\} \cup
\end{aligned}$$

$\{[[q_0, 0, q_0] \to \varepsilon, \ldots, [q_n, n, q_n] \to \varepsilon] : q_i \in Q,\ 0 \le i \le n\} \cup$
$\{[[q_0, 0, p_0] \to w_0[q_0', 0, p_0], \ldots, [q_n, n, p_n] \to w_n[q_n', n, p_n]] : r_j.q_jw_j \to q_j' \in \delta,\ 0 \le j \le n,\ (r_i, r_{i+1}) \in R,\ 0 \le i < n\}$.

Next, we prove $\mathcal{L}(G) = \mathcal{L}(M)$. To prove $\mathcal{L}(G) \subseteq \mathcal{L}(M)$, consider a derivation of w in G. Then, the derivation is of the form (1) and there are rules $r_0.q_0x_0^0 \to q_1^0, r_1.q_tx_0^1 \to q_1^1, \ldots, r_n.q_{i_{n-1}}x_0^n \to q_1^n$ in δ such that $(r_0, r_1), \ldots, (r_{n-1}, r_n) \in R$. Moreover, $(r_j^l, r_j^{l+1}) \in R$, where $r_j^l.q_j^lx_j^l \to q_{j+1}^l \in \delta$, and $(r_k^l, r_k^{l+1}) \in R$, where $r_k^l.q_k^lx_k^l \to q_{i_l} \in \delta, 0 \le l < n, 1 \le j < k$, q_{i_0} denotes q_t, and $q_{i_n} \in F$. Thus, M accepts w with the list of rules μ of the form (2).

To prove $\mathcal{L}(M) \subseteq \mathcal{L}(G)$, let μ be a list of rules used in an acceptance of

$$w = x_0^0x_1^0 \ldots x_k^0x_0^1x_1^1 \ldots x_k^1 \ldots x_0^nx_1^n \ldots x_k^n$$

in M of the form (2). Then, the derivation is of the form (1) because

$$[[q_j^0, 0, q_t] \to x_j^0[q_{j+1}^0, 0, q_t], \ldots, [q_j^n, n, q_{i_n}] \to x_j^n[q_{j+1}^n, n, q_{i_n}]] \in P,$$

for all $q_j^i \in Q, 1 \le i \le n, 1 \le j < k$, and $[[q_t, 0, q_t] \to \varepsilon, \ldots, [q_{i_n}, n, q_{i_n}] \to \varepsilon] \in P$.

Hence, Lemma 13 holds. □

The second main result of this paper follows next.

Theorem 14 *For all $n \ge 0$, $S_n = R_{[n+1]}$.*

Proof. This proof follows from Lemma 12 and 13. □

Corollary 15 *The following statements hold:*

1. $REG = S_0 \subset S_1 \subset S_2 \subset \ldots \subset CS$.
2. $S_1 \not\subseteq CF$.
3. $CF \not\subseteq S_n$, *for every* $n \ge 0$.
4. *For all $n \ge 0$, S_n is closed under union, concatenation, finite substitution, homomorphism, intersection with a regular language, and right quotient with a regular language.*
5. *For all $n \ge 1$, S_n is not closed under intersection, complement, and Kleene closure.*

Proof. Recall the following statements proved in [28]:

- $REG = R_{[1]} \subset R_{[2]} \subset R_{[3]} \subset \ldots \subset CS$.
- For all $n \ge 1$, $R_{[n]}$ is closed under union, finite substitution, homomorphism, intersection with a regular language, and right quotient with a regular language.
- For all $n \ge 2$, $R_{[n]}$ is not closed under intersection and complement.

Furthermore, recall these statements proved in [23] and [24]:

- For all $n \ge 1$, $R_{[n]}$ is closed under concatenation.
- For all $n \ge 2$, $R_{[n]}$ is not closed under Kleene closure.

These statements and Theorem 14 imply statements 1, 4, and 5 of Corollary 15. Moreover, observe that $\{ww : w \in \{a, b\}^*\} \in S_1 - CF$ (see *Example 2*), which proves 2. Finally, let $L = \{wcw^R : w \in \{a, b\}^*\}$. In [4, Theorem 1.5.2], there is a proof that $L \notin R_{[n]}$, for any $n \ge 1$. Thus, 3 follows from Theorem 14. □

Theorem 16, given next, follows from Theorem 14 and from Corollary 3.3.3 in [24]. However, Corollary 3.3.3 in [24] is not proved effectively. We next prove Theorem 16 effectively.

Theorem 16 *S_n is closed under inverse homomorphism, for all $n \geq 0$.*

Proof. For $n = 1$, let $M = (Q, \Sigma, \delta, q_0, q_t, F, R)$ be a 1-all-SFA, and let $h : \Delta^* \to \Sigma^*$ be a homomorphism. Next, we construct 1-all-SFA $M' = (Q', \Delta, \delta', q'_0, q'_t, \{q'_f\}, R')$ accepting $h^{-1}(\mathcal{L}(M))$ as follows. Denote $k = \max\{|w| : qw \to p \in \delta\} + \max\{|h(a)| : a \in \Delta\}$. Let $Q' = q'_0 \cup \{[x, q, y] : x, y \in \Sigma^*, |x|, |y| \leq k, q \in Q\}$. Initially, set δ' and R' to $\emptyset$. Then, extend δ' and R' by performing 1 through 5:

1. For $y \in \Sigma^*$, $|y| \leq k$, add
   $(q'_0 \to [\varepsilon, q_0, y], q'_t \to [y, q_t, \varepsilon])$ to R';
2. For $A \in Q'$, $q \neq q_t$, add
   $([x, q, y]a \to [xh(a), q, y], A \to A)$ to R';
3. For $A \in Q'$, add
   $(A \to A, [x, q, \varepsilon]a \to [xh(a), q, \varepsilon])$ to R';
4. For $(qx \to p, q'x' \to p') \in R$, $q \neq q_t$, add
   $([xw, q, y] \to [w, p, y], [x'w', q', \varepsilon] \to [w', p', \varepsilon])$ to R';
5. For $q_f \in F$, add
   $([y, q_t, y] \to q'_t, [\varepsilon, q_f, \varepsilon] \to q'_f)$ to R'.

In essence, M' simulates M in the following way. In a state of the form $[x, q, y]$, the three components have the following meaning:

- $x = h(a_1 \dots a_n)$, where $a_1 \dots a_n$ is the input string that M' has already read;
- q is the current state of M;
- y is the suffix remaining as the first component of the state that M' enters during a turn; y is thus obtained when M' reads the last symbol right before the turn occurs in M; M reads y after the turn.

More precisely, $h(w) = w_1 y w_2$, where w is an input string, w_1 is accepted by M before making the turn, i.e. from q_0 to q_t, and yw_2 is accepted by M after making the turn, i.e. from q_t to $q_f \in F$. A rigorous version of this proof is left to the reader.

For $n > 1$, the proof is analogous and left to the reader. □

4.3 Language Families Accepted by n-first-SFAs and n-all-SFAs

In this section, we compare the family of languages accepted by n-first-SFAs with the family of languages accepted by n-all-SFAs.

Theorem 17 *For all $n \geq 1$, $W_n \subset S_n$.*

Proof. In [21] and [28], it is proved that for all $n > 1$, $R_n \subset R_{[n]}$. The proof of Theorem 17 thus follows from Theorem 7 and 14. □

Theorem 18 *$W_n \not\subseteq S_{n-1}$, $n \geq 1$.*

Proof. It is easy to see that $L = \{a_1^k a_2^k \dots a_{n+1}^k : k \geq 1\} \in W_n = R_{n+1}$. However, $L \notin S_{n-1} = R_{[n]}$ (see Lemma 1.5.6 in [4]). □

Lemma 19 *For each regular language, L, language $\{w^n : w \in L\} \in S_{n-1}$.*

Proof. Let $L = \mathcal{L}(M)$, where M is a finite automaton. Make n copies of M. Rename their states so all the sets of states are pairwise disjoint. In this way, also rename the states in the rules of each of these n automata; however, keep the labels of the rules unchanged. For each rule label r, include (r, r) into R. As a result, we obtain an n-turn all-move self-regulating finite automaton that accepts $\{w^n : w \in L\}$. A rigorous version of this proof is left to the reader. □

Theorem 20 *$S_n - W \neq \emptyset$, for all $n \geq 1$, where $W = \bigcup_{m=1}^{\infty} W_m$.*

Proof. By induction on $n \geq 1$, we prove that language $L = \{(cw)^{n+1} : w \in \{a,b\}^*\} \notin W$. From Lemma 19, $L \in S_n$.

Basis: For $n = 1$, let G be an m-PRLG generating L, for some positive integer m. Consider a sufficiently large string $cw_1cw_2 \in L$ such that $w_1 = w_2 = a^{n_1}b^{n_2}$, $n_2 > n_1 > 1$. Then, there is a derivation of the form

$$\begin{array}{c} S \Rightarrow^p \\ x_1A_1x_2A_2 \ldots x_mA_m \Rightarrow^k x_1y_1A_1x_2y_2A_2 \ldots x_my_mA_m \end{array} \quad (3)$$

in G, where cycle (3) generates more than one a in w_1. The derivation continues as

$$\begin{array}{c} x_1y_1A_1x_2y_2A_2 \ldots x_my_mA_m \Rightarrow^r \\ x_1y_1z_1B_1x_2y_2z_2B_2 \ldots x_my_mz_mB_m \Rightarrow^l x_1y_1z_1u_1B_1x_2y_2z_2u_2B_2 \ldots x_my_mz_mu_mB_m \quad (4) \\ \text{(cycle (4) generates no } a\text{s)} \Rightarrow^s cw_1cw_2. \end{array}$$

Next, modify the left derivation, the derivation in components generating cw_1, so that the a-generating cycle (3) is repeated $(l + 1)$-times. Similarly, modify the right derivation, the derivation in the other components, so that the no-a-generating cycle (4) is repeated $(k + 1)$-times. Thus, the modified left derivation is of length $p+k(l+1)+r+l+s = p+k+r+l(k+1)+s$, which is the length of the modified right derivation. Moreover, the modified left derivation generates more as in w_1 than the right derivation in w_2—a contradiction.

Induction step: Suppose that the theorem holds for all $n \leq k$, for some $k \geq 1$. Consider $n + 1$ and let $\{(cw)^{n+1} : w \in \{a,b\}^*\} \in W_l$, for some $l \geq 1$. As W_l is closed under the right quotient with a regular language, and language $\{cw : w \in \{a,b\}^*\}$ is regular, we obtain $\{(cw)^n : w \in \{a,b\}^*\} \in W_l \subseteq W$—a contradiction. □

Fig. 4 summarizes the language families discussed in this paper.

5 Conclusion and Discussion

This paper has discussed self-regulating finite automata. As demonstrated next, we can analogically introduce and discuss self-regulating pushdown automata.

Recall that a *pushdown automaton* (see [15]), M, is a septuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where Q, Σ, $q_0 \in Q$, F are as in a finite automaton, Γ is a finite pushdown alphabet, δ is a finite set of rules of the form $Zqw \to \gamma p$, $q, p \in Q$, $Z \in \Gamma$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$, and Z_0 is an initial pushdown symbol. Again, let ψ denote the bijection from δ to Ψ, and write $r.Zqw \to \gamma p$ instead of $\psi(Zqw \to \gamma p) = r$. A configuration of M is any word from $\Gamma^*Q\Sigma^*$. For any configuration $xAqwy$, where $x \in \Gamma^*$, $y \in \Sigma^*$, $q \in Q$, and any $r.Aqw \to \gamma p \in \delta$, M makes a move from $xAqwy$ to $x\gamma py$ according to r, written as $xAqwy \Rightarrow x\gamma py\,[r]$. As usual, we define closure $\Rightarrow^*$. If $w \in \Sigma^*$ and $Z_0q_0w \Rightarrow^* f\,[\mu]$, $f \in F$, then w is accepted by M and $Z_0q_0w \Rightarrow^* f\,[\mu]$ is an acceptance of w in M. The language of M is defined as $\mathcal{L}(M) = \{w \in \Sigma^* : Z_0q_0w \Rightarrow^* f\,[\mu]$ is an acceptance of $w\}$.

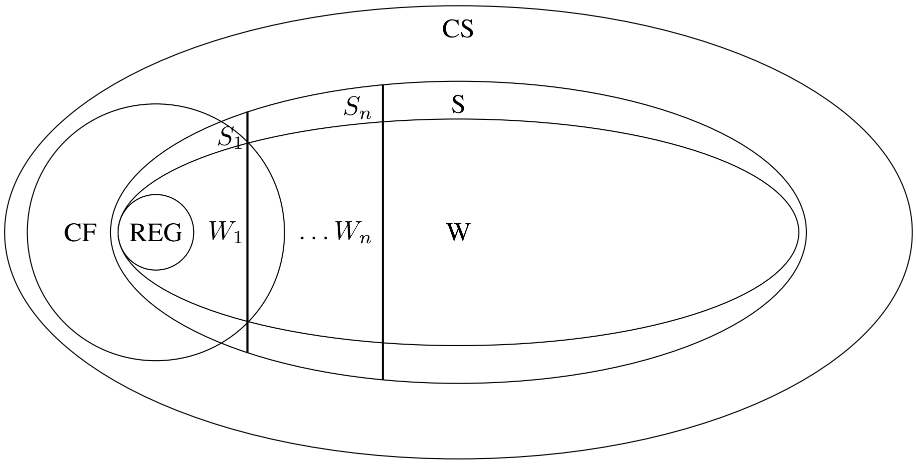

Fig. 4: The hierarchy of languages.

Definition 21 A *self-regulating pushdown automaton*, SPDA, M, is a nonuple

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_t, Z_0, F, R),$$

where

1. $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is a pushdown automaton,
2. $q_t \in Q$ is a *turn state*, and
3. $R \subseteq \Psi \times \Psi$ is a *finite relation*, where Ψ is an alphabet of rule labels.

Definition 22 Let $n \geq 0$ and $M = (Q, \Sigma, \Gamma, \delta, q_0, q_t, Z_0, F, R)$ be a self-regulating pushdown automaton. M is said to be an *n-turn first-move self-regulating pushdown automaton*, n-first-SPDA, if M accepts w in the following way. There is an acceptance $Z_0q_0w \Rightarrow^* f\,[\mu]$ such that

$$\mu = r_1^0 \dots r_k^0 r_1^1 \dots r_k^1 \dots r_1^n \dots r_k^n,$$

where $k \in \mathbb{N}$, r_k^0 is the first rule of the form $Zqx \to \gamma q_t$, for some $Z \in \Gamma$, $q \in Q$, $x \in \Sigma^*$, $\gamma \in \Gamma^*$, and

$$(r_1^j, r_1^{j+1}) \in R$$

for all $0 \leq j < n$.

The family of languages accepted by n-first-SPDAs is denoted by $\mathcal{L}(n\text{-first-SPDA})$.

Definition 23 Let $n \geq 0$ and $M = (Q, \Sigma, \Gamma, \delta, q_0, q_t, Z_0, F, R)$ be a self-regulating pushdown automaton. M is said to be an *n-turn all-move self-regulating pushdown automaton*, n-all-SPDA, if M accepts w in the following way. There is an acceptance $Z_0q_0w \Rightarrow^* f\,[\mu]$ such that

$$\mu = r_1^0 \dots r_k^0 r_1^1 \dots r_k^1 \dots r_1^n \dots r_k^n,$$

where $k \in \mathbb{N}$, r_k^0 is the first rule of the form $Zqx \to \gamma q_t$, for some $Z \in \Gamma$, $q \in Q$, $x \in \Sigma^*$, $\gamma \in \Gamma^*$, and

$$(r_i^j, r_i^{j+1}) \in R$$

for all $1 \leq i \leq k$, $0 \leq j < n$.

The family of languages accepted by n-all-SPDAs is denoted by $\mathcal{L}(n\text{-all-SPDA})$.

5.1 n-Turn All-Move Self-Regulating Pushdown Automata

It is easy to see that an n-turn all-move self-regulating pushdown automaton without any turn state is exactly a common pushdown automaton. Therefore, $\mathcal{L}(0\text{-all-SPDA}) = CF$. Moreover, if we consider 1-turn all-move self-regulating pushdown automata, their power is that of the Turing machines.

Theorem 24 $\mathcal{L}(1\text{-}all\text{-}SPDA) = RE$.

Proof. For any $L \in RE$, $L \subseteq \Delta^*$, there are context-free languages $\mathcal{L}(G)$ and $\mathcal{L}(H)$ and a homomorphism $h : \Sigma^* \to \Delta^*$ such that $L = h(\mathcal{L}(G) \cap \mathcal{L}(H))$ (see Theorem 1.12 in [14]). Suppose that $G = (N_G, \Sigma, P_G, S_G)$, $H = (N_H, \Sigma, P_H, S_H)$ are in the Greibach normal form, i.e. all rules are of the form $A \to a\alpha$, where A is a nonterminal, a is a terminal, and α is a (possibly empty) string of nonterminals. Let us construct 1-all-SPDA $M = (\{q_0, q, q_t, p, f\}, \Delta, \Sigma \cup N_G \cup N_H \cup \{Z\}, \delta, q_0, Z, \{f\}, R)$, $Z \notin \Sigma \cup N_G \cup N_H$, with R made as follows:

1. add $(Zq_0 \to ZS_Gq, Zq_t \to ZS_Hp)$ to R
2. add $(Aq \to B_n \dots B_1aq, Cp \to D_m \dots D_1ap)$ to R if
   $A \to aB_1 \dots B_n \in P_G$ and
   $C \to aD_1 \dots D_m \in P_H$
3. add $(aqh(a) \to q, ap \to p)$ to R
4. add $(Zq \to Zq_t, Zp \to f)$ to R

Moreover, δ contains only the rules from the definition of R.

Now, we prove $w \in h(\mathcal{L}(G) \cap \mathcal{L}(H))$ if and only if $w \in \mathcal{L}(M)$.

Only if Part: Let $w \in h(\mathcal{L}(G) \cap \mathcal{L}(H))$. There are $a_1, a_2, \dots, a_n \in \Sigma$ such that $a_1a_2 \dots a_n \in \mathcal{L}(G) \cap \mathcal{L}(H)$ and $w = h(a_1a_2 \dots a_n)$, for some $n \geq 0$. There are leftmost derivations $S_G \Rightarrow^n a_1a_2 \dots a_n$ and $S_H \Rightarrow^n a_1a_2 \dots a_n$ of length n in G and H, respectively, because in every derivation step exactly one terminal element is derived. Thus, M accepts $h(a_1)h(a_2) \dots h(a_n)$ as

$$Zq_0h(a_1)h(a_2) \dots h(a_n) \Rightarrow ZS_Gqh(a_1)h(a_2) \dots h(a_n), \dots, Za_nqh(a_n) \Rightarrow Zq,\ Zq \Rightarrow Zq_t,$$

$$Zq_t \Rightarrow ZS_Hp, \dots, Za_np \Rightarrow Zp,\ Zp \Rightarrow f.$$

In state q, by using its pushdown, M simulates G's derivation of $a_1 \dots a_n$ but reads $h(a_1) \dots h(a_n)$ as the input. In p, M simulates H's derivation of $a_1a_2 \dots a_n$ but reads no input. As $a_1a_2 \dots a_n$ can be derived in both G and H by making the same number of steps, the automaton can successfully complete the acceptance of w.

If Part: Notice that in one step, M can read only $h(a) \in \Delta^*$, for some $a \in \Sigma$. Let $w \in \mathcal{L}(M)$, then $w = h(a_1)h(a_2) \dots h(a_n)$, for some $a_1, a_2, \dots, a_n \in \Sigma$. Consider M's acceptance of w

$$Zq_0h(a_1)h(a_2) \dots h(a_n) \Rightarrow ZS_Gqh(a_1)h(a_2) \dots h(a_n), \dots, Za_nqh(a_n) \Rightarrow Zq,\ Zq \Rightarrow Zq_t,$$

$$Zq_t \Rightarrow ZS_Hp, \dots, Za_np \Rightarrow Zp,\ Zp \Rightarrow f.$$

As stated above, in q, M simulates G's derivation of $a_1a_2 \dots a_n$, and then in p, M simulates H's derivation of $a_1a_2 \dots a_n$. It successfully completes the acceptance of w only if $a_1a_2 \dots a_n$ can be derived in both G and H. Hence, the if part holds, too. $\square$

5.2 Open Problems

Although the fundamental results about self-regulating automata have been achieved in this paper, there still remain several open problems concerning them. Perhaps most importantly, these open problem areas include 1 through 3 given next:

1. What is the language family accepted by n-turn first-move self-regulating pushdown automata, when $n \geq 1$ (see Definition 22)?
2. By analogy with the standard deterministic finite and pushdown automata (see page 145 and page 437 in [15]), introduce the deterministic versions of self-regulating automata. What is their power?
3. Discuss the closure properties of other language operations, such as the reversal.

Acknowledgements

The authors would like to thank both anonymous referees for their suggestions. This work was supported by the GAČR 201/07/0005, 102/05/H050, and FR762/2007/G1 grants.

References

1. A. V. Aho and J. D. Ullman. *The Theory of Parsing, Translation and Compiling, Volume I: Parsing*. Prentice Hall, Englewood Cliffs, New Jersey, 1972.
2. J. M. Autebert, J. Berstel, and L. Boasson. Context-free languages and pushdown automata. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1, Word Language Grammar, pages 111–174. Springer-Verlag, Berlin, 1997.
3. B. Courcelle. On jump deterministic pushdown automata. *Math. Systems Theory*, 11:87–109, 1977.
4. J. Dassow and Gh. Păun. *Regulated Rewriting in Formal Language Theory*. Springer-Verlag, Berlin, 1989.
5. P. C. Fischer and A. L. Rosenberg. Multitape one-way nonwriting automata. *J. Comput. System Sci.*, 2:38–101, 1968.
6. S. Ginsburg, S. A. Greibach, and M. A. Harrison. One-way stack automata. *J. ACM*, 14:389–418, 1967.
7. S. Ginsburg and E. Spanier. Finite-turn pushdown automata. *SIAM J. Control*, 4:429–453, 1968.
8. S. A. Greibach. Checking automata and one-way stack languages. *J. Comput. System Sci.*, 3:196–217, 1969.
9. M. A. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley, Reading, Massachusetts, 1978.
10. O. H. Ibarra. Simple matrix languages. *Inform. and Control*, 17:359–394, 1970.
11. M. Ito. *Algebraic Theory of Automata and Languages*. World Scientific, Singapore, 2004.
12. H. R. Lewis and C. H. Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall, Englewood Cliffs, 1981.
13. J. C. Martin. *Introduction to Languages and the Theory of Computation*. McGraw-Hill, New York, 1991.
14. A. Mateescu and A. Salomaa. Aspects of classical language theory. In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages*, volume 1, Word Language Grammar, pages 175–251. Springer-Verlag, Berlin, 1997.
15. A. Meduna. *Automata and Languages: Theory and Applications*. Springer, London, 2000.
16. A. Meduna. Simultaneously one-turn two-pushdown automata. *Int. J. Comp. Math.*, 80:679–687, 2003.
17. A. Meduna and D. Kolář. Regulated pushdown automata. *Acta Cybernet.*, 14:653–664, 2000.

18. A. Meduna and D. Kolář. One-turn regulated pushdown automata and their reduction. *Fund. Inform.*, 51:399–405, 2002.
19. G. E. Revesz. *Introduction to Formal Languages*. McGraw-Hill, New York, 1983.
20. R. D. Rosebrugh and D. Wood. A characterization theorem for n-parallel right linear languages. *J. Comput. System Sci.*, 7:579–582, 1973.
21. R. D. Rosebrugh and D. Wood. Restricted parallelism and right linear grammars. *Util. Math.*, 7:151–186, 1975.
22. J. Sakarovitch. Pushdown automata with terminating languages. *Languages and Automata Symposium, RIMS 421, Kyoto University*, pages 15–29, 1981.
23. R. Siromoney. *Studies in the Mathematical Theory of Grammars and its Applications*. PhD thesis, University of Madras, Madras, India, 1969.
24. R. Siromoney. Finite-turn checking automata. *J. Comput. System Sci.*, 5:549–559, 1971.
25. T. A. Sudkamp. *Languages and Machines*. Addison-Wesley, Reading, Massachusetts, 1988.
26. L. Valiant. The equivalence problem for deterministic finite turn pushdown automata. *Inform. and Control*, 81:265–279, 1989.
27. D. Wood. Properties of n-parallel finite state languages. Technical report, McMaster University, 1973.
28. D. Wood. m-parallel n-right linear simple matrix languages. *Util. Math.*, 8:3–28, 1975.