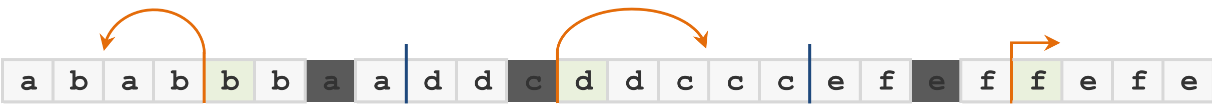
Excel@FIT 2015
http://excel.fit.vutbr.cz

n-Parallel Jumping Finite Automata

Radim Kocman*



Abstract
The present paper proposes an idea for a new investigation area in automata theory — *n-parallel jumping finite automata*. These automata are a combination of recently presented jumping finite automata and more settled n-parallel grammars. They read input words discontinuously as general jumping finite automata; however, they use multiple heads to do so, which is quite similar to the principle of the multiple nonterminals in n-parallel right linear grammars. This paper establishes definitions for such automata, outlines expected results, and suggests future investigation areas.

Keywords: jumping finite automata; n-parallel languages; discontinuous tape reading

Supplementary Material: *N/A*

*ikocman@fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

In previous century, most formal models were designed for continuous information processing. This, however, does not often reflects the requirements of modern information methods. Therefore, there is currently active research around the formal models that process information in a discontinuous way. Most notably, there are newly invented *jumping finite automata* that are completely focused on discontinuous reading. These automata go so far that they cannot even define some quite simple languages, e.g. a^*b^*, because they cannot guarantee any specific reading order between jumps.

The present paper proposes an idea for a modification of these automata — *n-parallel jumping finite automata*. They take inspiration from n-parallel grammars and divide input into n parts; each part is then processed with a separate head. This way, each part is read discontinuously but the overall order between parts is preserved. Consequently, such automata can define richer language families than standard jumping finite automata. This paper establishes definitions for such automata and outlines expected results.

2. Preliminaries

This paper assumes that the reader is familiar with theory of automata and formal languages (see [1, 2]). Let $\mathbb{N}$ denote the set of all positive integers. For a set Q, $card(Q)$ denotes the cardinality of Q. For an alphabet (finite nonempty set) V, V^* represents the free monoid generated by V under operation of concatenation. The unit of V^* is denoted by ε. For $x \in V^*$, $|x|$ denotes the length of x, and $alph(x)$ denotes the set of all symbols occurring in x; for instance, $alph(0010) = \{0,1\}$. For $a \in V$, $|x|_a$ denotes the number of occurrences of a in x. Let $x = a_1a_2 \dots a_n$, where $a_i \in V$, for some $n \geq 0$ ($x = \varepsilon$ if and only if $n = 0$).

A *general jumping finite automaton* (see [3]), a GJFA for short, is a quintuple, $M = (Q, \Sigma, R, s, F)$, where Q is a finite set of states, Σ is an input alphabet, $Q \cap \Sigma = \emptyset$, $R \subseteq Q \times \Sigma^* \times Q$ is finite, $s \in Q$ is the start state, and F is a set of final states. Members of R are referred as rules of M and instead of $(p, y, q) \in R$, we write $py \to q \in R$. A configuration of M is any string in $\Sigma^* Q \Sigma^*$. The binary jumping relation, symbolically denoted by $\curvearrowright$, over $\Sigma^* Q \Sigma^*$, is defined as follows. Let

$x, z, x', z' \in \Sigma^*$ such that $xz = x'z'$ and $py \to q \in R$; then, M makes a jump from $xpyz$ to $x'qz'$, symbolically written as $xpyz \curvearrowright x'qz'$. In standard manner, we extend $\curvearrowright$ to $\curvearrowright^m$, where $m \geq 0$. Let $\curvearrowright^+$ and $\curvearrowright^*$ denote the transitive closure of $\curvearrowright$ and transitive-reflexive closure of $\curvearrowright$, respectively. The language accepted by M, denoted by $L(M)$, is defined as $L(M) = \{uv \mid u, v \in \Sigma^*,\ usv \curvearrowright^* f,\ f \in F\}$. We also define two special cases of jumping relation. Let $w, x, y, z \in \Sigma^*$; then, (1) M makes a left jump from $wxpyz$ to $wqxz$, symbolically written as $wxpyz\ {}_l\!\curvearrowright\ wqxz$, and (2) M makes a right jump from $wpyxz$ to $wxqz$, written as $wpyxz\ {}_r\!\curvearrowright\ wxqz$. We denote language accepted by M as ${}_lL(M)$, and ${}_rL(M)$ when M uses only left jumps, and M uses only right jumps, respectively.

For $n > 0$, an *n-parallel right linear grammar* (see [4, 5, 6, 7, 8]), an n-PRLG for short, is an $(n+3)$-tuple $G = (N_1, \dots, N_n, T, S, P)$ where N_i, $1 \leq i \leq n$, are mutually disjoint nonterminal alphabets, T is a terminal alphabet, S is sentence symbol, S not in $N_1 \cup \dots \cup N_n \cup T$, and P is a finite set of pairs. Members of P are referred as rules of G and instead of $(X, x) \in P$, we write $X \to x \in P$. Each rule in P has one of the following forms: (1) $S \to X_1 \dots X_n$, $X_i \in N_i$, $1 \leq i \leq n$, (2) $X_i \to a_i$, $X_i \in N_i$, $a_i \in T^*$, $1 \leq i \leq n$, and (3) $X_i \to a_i Y_i$, $X_i, Y_i \in N_i$, $a_i \in T^*$, $1 \leq i \leq n$. The binary yield operation, symbolically denoted by $\Rightarrow$, is defined as follows. Let $x, y \in (N_1 \cup \dots \cup N_n \cup \{S\} \cup T)^*$ then $x \Rightarrow y$ iff either $x = S$ and $S \to y \in P$ or $x = a_1 X_1 \dots a_n X_n$, $y = a_1 x_1 \dots a_n x_n$ and $a_i \in T^*$, $X_i \in N_i$, $X_i \to x_i \in P$, $1 \leq i \leq n$. In standard manner, we extend $\Rightarrow$ to $\Rightarrow^m$, where $m \geq 0$. Let $\Rightarrow^+$ and $\Rightarrow^*$ denote the transitive closure of $\Rightarrow$ and transitive-reflexive closure of $\Rightarrow$, respectively. The language accepted by G, denoted by $L(G)$, is defined as $L(G) = \{x \mid S \Rightarrow^* x,\ x \in T^*\}$.

3. Definitions

In this section, we define n-parallel general jumping finite automaton together with multiple modified variants of its jumping relations. Each variant of jumping relation significantly changes behavior of such automaton. Thus, in fact, we introduce 12 distinct automata.

Definition 1. Let $n \in \mathbb{N}$. An *n-parallel general jumping finite automaton*, an n-PGJFA for short, is a quintuple

$$M = (Q, \Sigma, R, S, F)$$

where Q is a finite set of states, Σ is an input alphabet, $Q \cap \Sigma = \emptyset$, $R \subseteq Q \times \Sigma^* \times Q$ is finite, $S \subseteq Q^n$ is a set of start state strings, and F is a set of final states. Members of R are referred as rules of M and instead of $(p, y, q) \in R$, we write $py \to q \in R$.

A *configuration* of M is any string in $\Sigma^* Q \Sigma^*$. Let X denote the set of all configurations over M. The binary *jumping relation*, symbolically denoted by $\curvearrowright$, over X, is defined as follows. Let $x, z, x', z' \in \Sigma^*$ such that $xz = x'z'$ and $py \to q \in R$; then, M makes a jump from $xpyz$ to $x'qz'$, symbolically written as

$$xpyz \curvearrowright x'qz'.$$

Let \$ be a special symbol, $\$ \notin Q \cup \Sigma$. An *n-configuration* of M is any string in $(X\{\$\})^n$. Let ${}_nX$ denote the set of all n-configurations over M. The binary *first n-jumping relation*, symbolically denoted by ${}_n\!\curvearrowright_1$, over ${}_nX$, is defined as follows. Let $\zeta_1\$ \dots \zeta_n\$, \vartheta_1\$ \dots \vartheta_n\$ \in {}_nX$, so $\zeta_i, \vartheta_i \in X$, $1 \leq i \leq n$; then, M makes an n-jump from $\zeta_1\$ \dots \zeta_n\$$ to $\vartheta_1\$ \dots \vartheta_n\$$, symbolically written as

$$\zeta_1\$ \dots \zeta_n\$ \ {}_n\!\curvearrowright_1\ \vartheta_1\$ \dots \vartheta_n\$$$

iff $\zeta_i \curvearrowright \vartheta_i$ for all $1 \leq i \leq n$. In standard manner we extend ${}_n\!\curvearrowright_1$ to ${}_n\!\curvearrowright_1^m$, where $m \geq 0$. Let ${}_n\!\curvearrowright_1^+$ and ${}_n\!\curvearrowright_1^*$ denote the transitive closure of ${}_n\!\curvearrowright_1$ and transitive-reflexive closure of ${}_n\!\curvearrowright_1$, respectively.

The language accepted by M, denoted by $L(M, n, 1)$, is defined as $L(M, n, 1) = \{u_1 v_1 \dots u_n v_n \mid s_1 \dots s_n \in S,\ u_i, v_i \in \Sigma^*,\ u_1 s_1 v_1\$ \dots u_n s_n v_n\$ \ {}_n\!\curvearrowright_1^*\ f_1\$ \dots f_n\$,\ f_i \in F,\ 1 \leq i \leq n\}$. Let $w \in \Sigma^*$. We say that M accepts w if and only if $w \in L(M, n, 1)$. M rejects w if and only if $w \in \Sigma^* - L(M, n, 1)$.

Next, we define two special cases of jumping relations that corresponds with jumps in GJFAs.

Definition 2. Let $M = (Q, \Sigma, R, S, F)$ be an n-PGJFA. Let $w, x, y, z \in \Sigma^*$, and $py \to q \in R$; then, (1) M makes a left jump from $wxpyz$ to $wqxz$, symbolically written as

$$wxpyz\ {}_l\!\curvearrowright\ wqxz$$

and (2) M makes a right jump from $wpyxz$ to $wxqz$, written as

$$wpyxz\ {}_r\!\curvearrowright\ wxqz.$$

Let X denote the set of all configurations over M, ${}_nX$ denotes the set of all n-configurations over M, and $\zeta_1\$ \dots \zeta_n\$, \vartheta_1\$ \dots \vartheta_n\$ \in {}_nX$, so $\zeta_i, \vartheta_i \in X$, $1 \leq i \leq n$; then, (1) M makes a *left first n-jump* from $\zeta_1\$ \dots \zeta_n\$$ to $\vartheta_1\$ \dots \vartheta_n\$$, symbolically written as

$$\zeta_1\$ \dots \zeta_n\$ \ {}_{n-l}\!\curvearrowright_1\ \vartheta_1\$ \dots \vartheta_n\$$$

iff $\zeta_i\ {}_l\!\curvearrowright\ \vartheta_i$ for all $1 \leq i \leq n$; and (2) M makes a *right first n-jump* from $\zeta_1\$ \dots \zeta_n\$$ to $\vartheta_1\$ \dots \vartheta_n\$$, written as

$$\zeta_1\$ \dots \zeta_n\$ \ {}_{n-r}\!\curvearrowright_1\ \vartheta_1\$ \dots \vartheta_n\$$$

iff $\zeta_i\ {}_r\!\curvearrowright\ \vartheta_i$ for all $1 \leq i \leq n$.

Extend ${}_{n-l}\curvearrowright_1$ and ${}_{n-r}\curvearrowright_1$ to ${}_{n-l}\curvearrowright_1^m$, ${}_{n-l}\curvearrowright_1^+$, ${}_{n-l}\curvearrowright_1^*$, ${}_{n-r}\curvearrowright_1^m$, ${}_{n-r}\curvearrowright_1^+$, and ${}_{n-r}\curvearrowright_1^*$, where $m \geq 0$, by analogy with extending the corresponding notations for ${}_n\curvearrowright_1$. Let $L(M, n-l, 1)$, and $L(M, n-r, 1)$ denote language accepted by M using only left first n-jumps, and M using only right first n-jumps, respectively.

Now, we also define three different cases of n-jumping relation.

Definition 3. Let $M = (Q, \Sigma, R, S, F)$ be an n-PGJFA. Let X denote the set of all configurations over M, ${}_nX$ denotes the set of all n-configurations over M, and $\zeta_1\$\ldots\zeta_n\$, \vartheta_1\$\ldots\vartheta_n\$ \in {}_nX$, so $\zeta_i, \vartheta_i \in X$, $1 \leq i \leq n$; then, (1) M makes a *second n-jump* from $\zeta_1\$\ldots\zeta_n\$$ to $\vartheta_1\$\ldots\vartheta_n\$$, symbolically written as

$$\zeta_1\$\ldots\zeta_n\$ \ {}_n\curvearrowright_2 \ \vartheta_1\$\ldots\vartheta_n\$$$

iff $card(alph(\zeta_1\$\ldots\zeta_n\$) \cap Q) = 1$, and $\zeta_i \curvearrowright \vartheta_i$ for all $1 \leq i \leq n$; (2) M makes a *third n-jump* from $\zeta_1\$\ldots\zeta_n\$$ to $\vartheta_1\$\ldots\vartheta_n\$$, written as

$$\zeta_1\$\ldots\zeta_n\$ \ {}_n\curvearrowright_3 \ \vartheta_1\$\ldots\vartheta_n\$$$

iff either $\zeta_i \curvearrowright \vartheta_i$ or $\zeta_i = \vartheta_i$, $1 \leq i \leq n$; and (3) M makes a *fourth n-jump* from $\zeta_1\$\ldots\zeta_n\$$ to $\vartheta_1\$\ldots\vartheta_n\$$, written as

$$\zeta_1\$\ldots\zeta_n\$ \ {}_n\curvearrowright_4 \ \vartheta_1\$\ldots\vartheta_n\$$$

iff $\zeta_1\$\ldots\zeta_n\$ \ {}_n\curvearrowright_3 \ \vartheta_1\$\ldots\vartheta_n\$$ and $\zeta_i \neq \vartheta_i$ for no more than one i, $1 \leq i \leq n$.

Extend ${}_n\curvearrowright_2$, ${}_n\curvearrowright_3$, and ${}_n\curvearrowright_4$ to ${}_n\curvearrowright_2^m$, ${}_n\curvearrowright_2^+$, ${}_n\curvearrowright_2^*$, ${}_n\curvearrowright_3^m$, ${}_n\curvearrowright_3^+$, ${}_n\curvearrowright_3^*$, ${}_n\curvearrowright_4^m$, ${}_n\curvearrowright_4^+$, and ${}_n\curvearrowright_4^*$, where $m \geq 0$, by analogy with extending the corresponding notations for ${}_n\curvearrowright_1$. Let $L(M, n, 2)$, $L(M, n, 3)$, and $L(M, n, 4)$ denote language accepted by M using only second n-jumps, M using only third n-jumps, and M using only fourth n-jumps, respectively.

If we combine previous definitions together, we can also get $L(M, n-l, 2)$, $L(M, n-r, 2)$, $L(M, n-l, 3)$, $L(M, n-r, 3)$, $L(M, n-l, 4)$, and $L(M, n-r, 4)$.

4. Examples

This section illustrates previous definitions on three simple examples. Each example corresponds to the definition with the same ordinal number. Every example defines specific n-PGJFA and shows the language accepted with such automaton and corresponding jumps.

Example 1. Consider the 2-PGJFA

$$M = (\{s, r, p, q\}, \Sigma, R, \{sr\}, \{s, r\}),$$

where $\Sigma = \{a, b, c, d\}$ and R consists of the rules

$$sa \to p, \ \ pb \to s, \ \ rc \to q, \ \ qd \to r.$$

Starting from sr, M has to read some a, and some b with the first head and some c, and some d with the second head, entering again the start (and also final) states sr. Therefore, the accepted language is

$$L(M, 2, 1) = \{uv \mid u \in \{a, b\}^*, \ v \in \{c, d\}^*, \\ |w|_a = |w|_b = |w|_c = |w|_d\}.$$

Such language cannot be neither accepted by any GJFA, or generated by any n-PRLG.

Example 2. Consider the 2-PGJFA

$$M = (\{s, r, t\}, \Sigma, R, \{ss\}, \{s\}),$$

where $\Sigma = \{a, b, c\}$ and R consists of the rules

$$sa \to r, \ \ rb \to t, \ \ tc \to s.$$

Starting from ss, M has to read some a, some b, and some c with both heads. If we work with unbound jumps or left jumps, each head can read a, b, and c in arbitrary order. However, if we work only with right jumps, each head must read input symbols in the original order. Therefore, the accepted languages are

$$L(M, 2, 1) = \{ww \mid w \in \{a, b, c\}^*, |w|_a = |w|_b = |w|_c\},$$

$$L(M, 2-l, 1) = L(M, 2, 1),$$

$$L(M, 2-r, 1) = \{ww \mid w \in \{abc\}^*\}.$$

It is not yet clear how precisely do left jumps affect the acceptance power of GJFAs, and n-PGJFAs. Language $L(M, 2-r, 1)$ can be generated by an n-PRLG.

Example 3. Consider the 2-PGJFA

$$M = (\{s, r, t\}, \Sigma, R, \{ss, sr\}, \{s, r\}),$$

where $\Sigma = \{a, b\}$ and R consists of the rules

$$sa \to s, \ \ rb \to t, \ \ tb \to r.$$

If we start from ss, both heads of M have to read some a; if we start from sr, M has to read two times some a with the first head, and two times some b with the second head. If we use second n-jumps, then we can start only from ss; if we use third n-jumps or fourth n-jumps, then we desynchronize the heads. Therefore, the accepted languages are

$$L(M, 2, 1) = \{w \mid w \in \{a\}^{2m}, \ m \geq 0\} \\ \cup \{w \mid w \in \{a\}^{2m}\{b\}^{2m}, \ m \geq 0\},$$

$$L(M,2,2) = \{w \mid w \in \{a\}^{2m},\ m \geq 0\},$$

$$L(M,2,3) = \{w \mid w \in \{a\}^{m},\ m \geq 0\} \cup \{w \mid w \in \{a\}^{m}\{b\}^{2l},\ m,l \geq 0\},$$

$$L(M,2,4) = L(M,2,3).$$

Observed behavior of these different n-jumping relations will be discussed in the next section.

5. Expected results

In general, we predict that every variant of n-PGJFAs, with its increasing n, will define an infinite hierarchy of language families. This is well-known and common property for parallel grammars, such as n-parallel grammars and simple matrix grammars (see [9]).

Let $n = 1$; then, all variants of 1-PGJFAs should accept same language families as GJFAs. Consequently 1-PGJFAs with right jumps should accept same language family as classical finite automata. Further, for $n = 1$, all n-jumping relations should work in the same way, because there are no multiple heads that could be synchronized in any way.

Let $n \geq 1$; then, we are almost sure that n-PGJFAs with right jumps define exactly the same infinite hierarchy of language families as n-PRLGs. The power of other variants is currently unknown and is subject for further research. It is, however, possible that third n-jumps and fourth n-jumps define the same infinite hierarchy of language families.

It is also possible that we do not find any counterpart in grammars for unrestricted version of n-PGJFAs. That was the same for GJFAs, until jumping grammars (see [10]) were recently introduced. Therefore, there is the area for further research. In the same manner, we could also try to investigate jumping automata that would use properties from simple matrix grammars, which are generally more powerful than n-parallel grammars.

6. Conclusion

This paper suggested the study of n-parallel jumping finite automata as a new investigation area in automata theory. Within the previous sections, we established definitions for such automata together with multiple variants of their jumping relations. Subsequently, several examples briefly showed differences between these jumping relations.

We believe that these automata will create more general model of jumping finite automata; in the same way as n-parallel right linear grammars create more general model of right linear grammars. Such approach would combine positive effects of jumping and parallelism, and therefore enabled define language families that cannot be defined by neither of these previous models alone. Furthermore, with restricted variants of jumping relations, we could also cover all previously defined language families.

As a following step in the research, we need to prepare theorems that will mathematically prove our claims about the expected results.

Acknowledgments

The author wishes to thank Prof. RNDr. Alexander Meduna, CSc. for his help in the preparation of this paper.

This work was supported by the BUT grant FIT-S-14-2299.

References

[1] Alexander Meduna. *Automata and Languages: Theory and Applications*. Springer, London, 2000.

[2] Derick Wood. *Theory of Computation: A Primer*. Addison-Wesley, Boston, 1987.

[3] Alexander Meduna and Petr Zemek. Jumping finite automata. *International Journal of Foundations of Computer Science*, 23(7):1555–1578, 2012.

[4] Robert D. Rosebrugh and Derick Wood. Restricted parallelism and right linear grammars. *Utilitas Mathematica*, 7:151–186, 1975.

[5] Derick Wood. n-linear simple matrix languages and n-parallel linear languages. *Rev. Roum. de Math. Pures et Appl.*, pages 408–412, 1977.

[6] Derick Wood. Properties of n-parallel finite state languages. *Utilitas Mathematica*, 4:103–113, 1973.

[7] Robert D. Rosebrugh. A characterization theorem for n-parallel right linear languages. *Journal of Computer and System Sciences*, 7:579–582, 1973.

[8] Robert D. Rosebrugh and Derick Wood. Image theorem for simple matrix languages and n-parallel languages. *Mathematical Systems Theory*, 8(2), 1974.

[9] Oscar H. Ibarra. Simple matrix languages. *Information and control*, 17:359–394, 1970.

[10] Alexander Meduna and Krivka Zbynek. Jumping grammars. *International Journal of Foundations of Computer Science*. (in press).