Jumping Finite Automata: New Results
Part Two: New Models

Radim Kocman and Zbyněk Křivka

Faculty of Information Technology
Brno University of Technology
Božetěchova 2, Brno Czech Republic

{ikocman,krivka}@fit.vutbr.cz

LTA 2016 (December 5, 2016)
Table of Contents

- Motivation
- n-Parallel Jumping Finite Automata
- Double-Jumping Finite Automata
- One-Way Jumping Finite Automata
Motivation
Why study other models?
Possible Advantages

- completely discontinuous reading
- can accept some CF and CS languages
Motivation – Jumping Finite Automata

Possible Advantages
- completely discontinuous reading
- can accept some CF and CS languages

Possible Disadvantages
- cannot guarantee any specific reading order
- therefore it cannot accept languages like $a^*b^*$
- heavily nondeterministic behavior
Definition

A GJFA makes a right jump from $wpyxz$ to $wxqz$ by $py \rightarrow q$:

$$wpyxz \overset{r}{\sim} wxqz$$

where $w, x, y, z \in \Sigma^*$. 

A GJFA makes a left jump from $wxpyz$ to $wqxz$ by $py \rightarrow q$:

$$wxpyz \overset{l}{\sim} wqxz$$

where $w, x, y, z \in \Sigma^*$. 

Definition

A GJFA makes a **right jump** from \( wpyz \) to \( wxqz \) by \( py \rightarrow q \):

\[
wpyz \; r \ni \; wxqz
\]

where \( w, x, y, z \in \Sigma^* \).

Definition

A GJFA makes a **left jump** from \( wxpyz \) to \( wqxz \) by \( py \rightarrow q \):

\[
wxpyz \; l \ni \; wqxz
\]

where \( w, x, y, z \in \Sigma^* \).
Motivation – Right and Left Jumps

Properties of Right Jumps

- consider the configuration \( u \rho v \), where \( \rho \in Q \), \( u, v \in \Sigma^* \)
- the automaton will get stuck for any \( |u| > 0 \)
- result: the same power as FAs
Properties of Right Jumps

- consider the configuration $u p v$, where $p \in Q$, $u$, $v \in \Sigma^*$
- the automaton will get stuck for any $|u| > 0$
- result: the same power as FAs

Properties of Left Jumps

- open problem
- can define some non-regular languages
Motivation for New Models

- partially discontinuous reading
- partially continuous reading
- explore new possibilities
- more deterministic behavior
$n$-Parallel Jumping Finite Automata

Based on
Radim Kocman and Alexander Meduna
On Parallel Versions of Jumping Finite Automata
Proceedings of SDOT 2015
• heavily used in formal grammars
  
  \( (n\text{-parallel grammars, simple matrix grammars, ...}) \)
- heavily used in formal grammars
  \((n\text{-parallel grammars, simple matrix grammars, ...})\)

### Example Derivations

\[ S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow aaAbbbBccC \Rightarrow aaabbbccc \]
• heavily used in formal grammars
  \((n\text{-parallel grammars, simple matrix grammars, }\ldots)\)

Example Derivations

\[
S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow aaAAbbBccC \Rightarrow aaabbbcccc
\]

• rarely used in classical automata
  \((\text{multiple tapes, more heads reading the same input, }\ldots)\)
- Parallel JFAs – Parallelism

- heavily used in formal grammars
  \((n\)-parallel grammars, simple matrix grammars, \ldots\)

<table>
<thead>
<tr>
<th>Example Derivations</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow aaAbbBccC \Rightarrow aaabbbcccc )</td>
</tr>
</tbody>
</table>

- rarely used in classical automata
  \((\text{multiple tapes, more heads reading the same input, } \ldots\) \)

- What if the parallelism is combined with the jumping?
**Definition**

An \( n \)-parallel general jumping finite automaton (\( n \)-PGJFA) is a quintuple

\[
M = (Q, \Sigma, R, S, F)
\]

where

- \( Q \) is a finite set of states;
- \( \Sigma \) is an input alphabet, \( Q \cap \Sigma = \emptyset \);
- \( R \) is a finite set of rules: \( py \rightarrow q \), where \( p, q \in Q, y \in \Sigma^* \);
- \( S \) is a set of start state strings, \( S \subseteq Q^n \);
- \( F \) is a set of final states.
• arbitrary splits the input into $n$ parts
• steps of all heads are synchronized
• different types of the jumping:
  • unrestricted jumps – each part is processed as in JFA
  • right jumps – each part is processed as in FA
Consider the 3-PGJFA

\[ M = (\{s, r, p\}, \Sigma, R, \{srp\}, \{s, r, p\}) \],

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

\[ sa \rightarrow s, \quad rb \rightarrow r, \quad pc \rightarrow p. \]

\[ L(M, 3-r) = \{a^n b^n c^n \mid n \geq 0\} \]
Example

Consider the 3-PGJFA

\[ M = (\{s, r, p\}, \Sigma, R, \{srp\}, \{s, r, p\}) \],

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

\[ sa \rightarrow s, \quad rb \rightarrow r, \quad pc \rightarrow p. \]

\[ L(M, 3-r) = \{a^n b^n c^n | n \geq 0\} \]

Example Steps (for \( n = 3 \) with only right jumps)

\[ |aaa|bbb|ccc \sim |aa|bb|cc \sim |a|b|c \sim |\]
Theorem

For every \( n \)-PRLG \( G = (N_1, \ldots, N_n, T, S_1, P) \), there is an \( n \)-PGJFA using only right \( n \)-jumps \( M = (Q, \Sigma, R, S_2, F) \), such that \( L(M, n-r) = L(G) \).

Theorem

For every \( n \)-PGJFA using only right \( n \)-jumps \( M = (Q, \Sigma, R, S_2, F) \), there is an \( n \)-PRLG \( G = (N_1, \ldots, N_n, T, S_1, P) \), such that \( L(G) = L(M, n-r) \).
n-Parallel JFAs – Characterization

Theorem
\[ r_1 \text{-PGJFA} = r \text{GJFA} = \text{REG}. \]

Theorem
\[ r_2 \text{-PGJFA} \subset \text{CF}. \]

Theorem
\[ r_n \text{-PGJFA} \subset \text{CS} \] and there exist non-context-free languages in \( r_n \text{-PGJFA} \) for all \( n > 2 \).

Theorem
For all \( n \in \mathbb{N} \), \( r_n \text{-PGJFA} \subset r(n + 1) \text{-PGJFA} \).
Double-Jumping Finite Automata

Based on

Radim Kocman, Zbyněk Křivka and Alexander Meduna
On Double-Jumping Finite Automata
Proceedings of NCMA 2016
A general jumping finite automaton (GJFA) is a quintuple

\[ M = (Q, \Sigma, R, s, F) \]

where

- \( Q \) is a finite set of states;
- \( \Sigma \) is the input alphabet;
- \( R \) is a finite set of rules of the form
  
  \[ py \rightarrow q \quad (p, q \in Q, y \in \Sigma^*) \]

- \( s \) is the start state;
- \( F \) is a set of final states.
### Double-JFAs – Modes

#### Used Symbols
- ▶ – Right Jump
- ◀ – Left Jump
- ♦ – Both Directions

#### Studied Modes
- ♦♦↷ – Unrestricted 2-Jumps
- ▶◀↷ – Right-Left 2-Jumps
- ▶▶↷ – Right-Right 2-Jumps
- ◀▶↷ – Left-Right 2-Jumps
- ◀◀↷ – Left-Left 2-Jumps

Example:

$L(M)^{♭}$

$$L(M)^{♭} = \{uvw | u, v, w \in \Sigma^*, usvsw^{♭} | f, f \in F\}.$$
Double-JFAs – Modes

Used Symbols

▶ – Right Jump
◀ – Left Jump
♦ – Both Directions

Studied Modes

♦♦↷ – Unrestricted 2-Jumps
▶◀↷ – Right-Left 2-Jumps
▶▶↷ – Right-Right 2-Jumps
◀▶↷ – Left-Right 2-Jumps
◀◀↷ – Left-Left 2-Jumps

Example

\[ L(M_{\leftrightarrow\circ}) = \{uvw | u, v, w \in \Sigma^*, usvsw \leftrightarrow\circ* ff, f \in F \}. \]
Conditions for 2-Jumps

- both jumps follow the same rule
- the jumps cannot ever cross each other
Conditions for 2-Jumps

- both jumps follow the same rule
- the jumps cannot ever cross each other

Example with ◀▶↷

- configuration: $uu'a p v p a w' w$, where $a, u, u', v, w, w' \in \Sigma^*, p \in Q$
- rule: $(p, a, q)$
- 2-jump: $uu'a p v p a w' w \leftrightarrow \sim uqu'v w' q w$
Properties

- required initial configuration: \( sxs \), where \( x \in \Sigma^* \)
- cannot jump over any symbols
- every \( x \in L(M_{\uparrow\downarrow\leftarrow}) \) can be written as
  \( x = u_1u_2 \ldots u_nu_n \ldots u_2u_1 \), where \( n \in \mathbb{N} \), and \( u_i \in \Sigma^*, 1 \leq i \leq n \)
- accept string palindromes of even length
Properties

- required initial configuration: $sxs$, where $x \in \Sigma^*$
- cannot jump over any symbols
- every $x \in L(M_{\leftarrow\rightarrow})$ can be written as $x = u_1u_2\ldots u_nu_n\ldots u_2u_1$, where $n \in \mathbb{N}$, and $u_i \in \Sigma^*$, $1 \leq i \leq n$
- accept string palindromes of even length

Language Family ($L_{\leftarrow\rightarrow}$)

- a subfamily of the family of linear languages
- the same as $L_{\leftarrow\leftarrow\rightarrow}$
Double-JFAs – Comparison

- identity
- proper inclusion
- incomparability

Diagram:

- REG — LIN — SEL
- REG — L — SEL
- REG — FIN — L
- REG — L — FIN
- REG — L — L

Jumping Finite Automata: New Results – Part Two: New Models
Properties

- the first jump should not skip symbols
- the second jump can skip symbols
Properties

- the first jump should not skip symbols
- the second jump can skip symbols

Example Behavior

- $u_1 u'_1 u_2 u'_2 \ldots u_n u'_n$, where $n \in \mathbb{N}$, $u_i, u'_i \in \Sigma$, $u_i = u'_i$, $1 \leq i \leq n$
- red symbols can be also shifted to the right over blue symbols
Properties

- the first jump should not skip symbols
- the second jump can skip symbols

Example Behavior

- \( u_1 u'_1 u_2 u'_2 \ldots u_n u'_n \), where \( n \in \mathbb{N} \), \( u_i, u'_i \in \Sigma \), \( u_i = u'_i \), \( 1 \leq i \leq n \)
- red symbols can be also shifted to the right over blue symbols

Language Family \( (L_{\Rightarrow\Rightarrow}) \)

- a subfamily of the family of context-sensitive languages
- not the same as \( L_{\Rightarrow\Leftarrow\Rightarrow} \)
Double-JFAs – Comparison

identity

proper inclusion

incomparability

CS

CF

L

REG

FIN

L

Jumping Finite Automata: New Results – Part Two: New Models
<table>
<thead>
<tr>
<th>Operation</th>
<th>$L_{\rightarrow\leftarrow}$</th>
<th>$L_{\rightarrow\rightarrow}$</th>
<th>$L_{\rightarrow\leftarrow}$</th>
<th>$L_{\leftarrow\rightarrow}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>endmarking (both sides)</td>
<td>$-$</td>
<td>$(+)$</td>
<td>$-(−)$</td>
<td>$-(−)$</td>
</tr>
<tr>
<td>concatenation</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>square ($L^2$)</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>shuffle</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>union</td>
<td>$+$</td>
<td>$+$</td>
<td>$+$</td>
<td>$+$</td>
</tr>
<tr>
<td>complement</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>intersection</td>
<td>$+$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>int. with regular languages</td>
<td>$+$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>mirror image</td>
<td>$+$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>finite substitution</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>homomorphism</td>
<td>$+$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>$\varepsilon$-free homomorphism</td>
<td>$+$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td>inverse homomorphism</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
</tbody>
</table>
One-Way Jumping Finite Automata

Based on

Hiroyuki Chigahara, Szilárd Zsolt Fazekas and Akihiro Yamamura
One-way Jumping Finite Automata

Szilárd Zsolt Fazekas and Akihiro Yamamura
On Regular Languages accepted by One-Way Jumping Finite Automata
Short Papers of NCMA 2016
### Definition

A **right one-way jumping finite automaton** (ROWJFA) is a quintuple $M = (Q, \Sigma, R, s, F)$, where $Q$, $\Sigma$, $R$, $s$ and $F$ are defined as in a DFA.
A right one-way jumping finite automaton (ROWJFA) is a quintuple $M = (Q, \Sigma, R, s, F)$, where $Q$, $\Sigma$, $R$, $s$ and $F$ are defined as in a DFA.

The right one-way jumping relation, symbolically denoted by $\rightarrow$, over $Q\Sigma^*$, is defined as follows. Suppose that $x$ and $y$ belong to $\Sigma^*$, $a$ belongs to $\Sigma$, $p$ and $q$ are states in $Q$ and $pa \rightarrow q \in R$. Then the ROWJFA $M$ makes a jump from the configuration $pxay$ to the configuration $qyx$, written as

$$pxay \rightarrow qyx$$

if $x$ belongs to $\{\Sigma \setminus \Sigma_p\}^*$ where

$\Sigma_p = \{b \in \Sigma \mid (p, b, q) \in R \text{ for some } q \in Q\}$. 
The language accepted by $M$, denoted by $L(M)$, is defined as

$$L(M) = \{ w \in \Sigma^* | sw \circ^* f, f \in F \}$$
One-Way JFAs – Definitions

Definition

The language accepted by $M$, denoted by $L(M)$, is defined as

$$L(M) = \{ w \in \sum^* \mid sw \underset{\circ} \rightarrow^* f, f \in F \}$$

- fully deterministic behavior

- There is also a similar definition for the left one-way jumping finite automaton.
Example 1

Let $M_1$ be a ROWJFA given by

$$M_1 = (\{q_0, q_1, q_2\}, \Sigma, R, q_0, \{q_0\}),$$

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

$$q_0a \rightarrow q_1, \quad q_1b \rightarrow q_2, \quad q_2c \rightarrow q_0.$$

$L(M_1) = \{w \in \Sigma \mid |w|_a = |w|_b = |w|_c\}$
Example 1

Let $M_1$ be a ROWJFA given by

$$M_1 = (\{q_0, q_1, q_2\}, \Sigma, R, q_0, \{q_0\})$$

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

$$q_0 a \rightarrow q_1, \quad q_1 b \rightarrow q_2, \quad q_2 c \rightarrow q_0.$$

$L(M_1) = \{w \in \Sigma \mid |w|_a = |w|_b = |w|_c\}$

Example 2

Let $M_2$ be a ROWJFA given by

$$M_2 = (\{q_0, q_1\}, \Sigma, R, q_0, \{q_0, q_1\})$$

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

$$q_0 a \rightarrow q_0, \quad q_0 b \rightarrow q_1, \quad q_1 b \rightarrow q_1.$$ 

$L(M_2) = a^*b^*$
Theorem
ROWJ properly includes REG.

Theorem
ROWJ and LOWJ are incomparable.

Theorem
ROWJ $\not\subset$ JFA.

Theorem
CF and ROWJ are incomparable.
Theorem

The class \textbf{ROWJ} is not closed under

- intersection,
- concatenation,
- reversal,
- intersection with regular languages,
- concatenation with regular languages,
- substitution,
- Kleene star,
- Kleene plus.

Theorem

Let $M$ be a ROWJFA. If there exists a constant $k$, such that for any word $w \in L(M)$ the number of sweeps needed by $M$ to process $w$ is at most $k$, then the language $L(M)$ is regular.
Overall Conclusion

\(n\)-Parallel Jumping Finite Automata
– combination of the parallel and jumping behavior

Double-Jumping Finite Automata
– parallel combination of different jumping modes

One-Way Jumping Finite Automata
– fully deterministic behavior
Thank you for your attention!