On Proof Techniques in Jumping Models

Radim Kocman

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

ikocman@fit.vutbr.cz

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

LTA 2019 (December 18, 2019)

Table of Contents

Motivation

- In formal language theory, it is a common task to prove that a certain language can or cannot be accepted by the model in question.
- Student courses (IFJ, etc.) show only basic, well-known techniques (pumping lemmas, etc.).
- This talk shows new techniques used in current research.
- Motivation for further study.

Finite Automata

Finite Automata – Definition (1/2)

Lazy Finite Automaton (LFA)

$$\textit{quintuple} \ \ M = (Q, \Sigma, R, s, F)$$

Q is a finite set of states

Σ is an input alphabet, $Q \cap \Sigma = \emptyset$

R is a finite set of rules: (p, y, q), where $p, q \in Q$, $y \in \Sigma^*$

s is the start state

F is a set of final states

FIT

Lazy Finite Automaton (LFA)

quintuple $M = (Q, \Sigma, R, s, F)$

Q is a finite set of states

Σ is an input alphabet, $Q \cap \Sigma = \emptyset$

R is a finite set of rules: (p, y, q), where $p, q \in Q$, $y \in \Sigma^*$

s is the start state

F is a set of final states

Finite Automaton (FA)

If $(p, y, q) \in R$ implies that $|y| \leq 1$.

Finite Automata – Definition (2/2)

Configuration

$$pw$$

p is the state

w is an unprocessed input

Configuration

$$pw$$

p is the state

w is an unprocessed input

Step/Move

$$pyx \Rightarrow qx$$

if $(p, y, q) \in R$ and $x, y \in \Sigma^*$.
In the standard manner, define $\Rightarrow^+$ and $\Rightarrow^*$.

Configuration

$$pw$$

p is the state

w is an unprocessed input

Step/Move

$$pyx \Rightarrow qx$$

if $(p, y, q) \in R$ and $x, y \in \Sigma^*$.
In the standard manner, define $\Rightarrow^+$ and $\Rightarrow^*$.

Accepted language

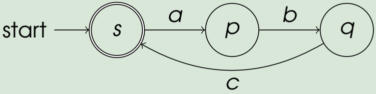
$L(M) = \{w \in \Sigma^* : sw \Rightarrow^* f, \ f \in F\}$

Example automaton

$$M = (\{s, p, q\}, \{a, b, c\}, R, s, \{s\})$$

where R:

(s, a, p)
(p, b, q)
(q, c, s)



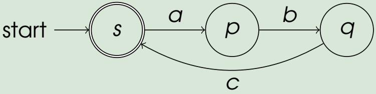
FA – Accepted Languages

Example automaton

$$M = (\{s, p, q\}, \{a, b, c\}, R, s, \{s\})$$

where R:

(s, a, p)
(p, b, q)
(q, c, s)



Example input: *abcabc*

$sabcabc \Rightarrow pbcabc \Rightarrow qcabc \Rightarrow sabc \Rightarrow pbc \Rightarrow qc \Rightarrow s$

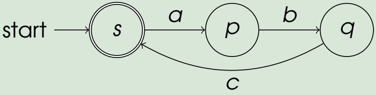
FA – Accepted Languages

Example automaton

$$M = (\{s, p, q\}, \{a, b, c\}, R, s, \{s\})$$

where R:

(s, a, p)
(p, b, q)
(q, c, s)



Example input: *abcabc*

$sabcabc \Rightarrow pbcabc \Rightarrow qcabc \Rightarrow sabc \Rightarrow pbc \Rightarrow qc \Rightarrow s$

Resulting language

- FA: $L(M) = \{abc\}^*$

- What about $L = \{a^n b^n : n \geq 0\}$?
- Can we construct an FA that accepts L?
- How to rigorously prove that it is not possible?

- What about $L = \{a^n b^n : n \geq 0\}$?
- Can we construct an FA that accepts L?
- How to rigorously prove that it is not possible?

Pumping lemma for regular languages

Let L be a regular language over Σ. Then there is a constant k, depending on L, such that for each $w \in L$ with $|w| \geq k$ there exist $x, y, z \in \Sigma^*$ such that $w = xyz$ and

1. $|xy| \leq k$,
2. $|y| > 0$,
3. $xy^iz \in L$ for all $i \geq 0$.

- This lemma is necessary but not sufficient.
- There are sufficient lemmas but they are more complicated.

Theorem

There is no FA M such that $L(M) = \{a^n b^n : n \geq 0\}$.

Proof.

By contradiction. Assume that there is a FA M such that $L(M) = \{a^n b^n : n \geq 0\}$. Then, $L(M)$ is a regular language.

Choose $w = a^k b^k$ in $L(M)$. Clearly, $|w| \geq k$.

By the pumping lemma, $w = xyz$ for some $x, y, z \in \Sigma^*$ such that (1) $|xy| \leq k$, (2) $|y| > 0$, and (3) $xy^i z \in L(M)$ for all $i \geq 0$.

By (1) and (2), we have $y = a^m$, $1 \leq m \leq k$.
But $xy^0 z = xz = a^{k-m} b^k \notin L(M)$. Thus, (3) does not hold.
Therefore, there is no FA M such that $L(M) = \{a^n b^n : n \geq 0\}$. $\square$

Jumping Finite Automata

Based on

- Alexander Meduna and Petr Zemek
  Jumping Finite Automata
  Int. J. Found. Comput. Sci. 23(7):1555–1578 (2012)

- Alexander Meduna and Petr Zemek
  Regulated Grammars and Automata
  Springer (2014)

General Jumping Finite Automaton (GJFA)

$$\textit{quintuple} \;\; M = (Q, \Sigma, R, s, F)$$

Q, Σ, R, s, F are defined as in LFA.

If $(p, y, q) \in R$ implies that $|y| \leq 1$, then M is a jumping finite automaton (JFA).

General Jumping Finite Automaton (GJFA)

$$\textit{quintuple} \;\; M = (Q, \Sigma, R, s, F)$$

Q, Σ, R, s, F are defined as in LFA.

If $(p, y, q) \in R$ implies that $|y| \leq 1$, then M is a jumping finite automaton (JFA).

Configuration

upv where $u, v \in \Sigma^*$ and $p \in Q$.

Jumping Finite Automata – Definition

General Jumping Finite Automaton (GJFA)

$$\textit{quintuple} \quad M = (Q, \Sigma, R, s, F)$$

Q, Σ, R, s, F are defined as in LFA.

If $(p, y, q) \in R$ implies that $|y| \leq 1$, then M is a jumping finite automaton (JFA).

Configuration

upv where $u, v \in \Sigma^*$ and $p \in Q$.

Jump

$$xpyz \curvearrowright x'qz'$$

if $x, z, x', z' \in \Sigma^*$ such that $xz = x'z'$ and $(p, y, q) \in R$; $\curvearrowright^+$, $\curvearrowright^*$.

Jumping Finite Automata – Definition

General Jumping Finite Automaton (GJFA)

$$\textit{quintuple} \quad M = (Q, \Sigma, R, s, F)$$

Q, Σ, R, s, F are defined as in LFA.

If $(p, y, q) \in R$ implies that $|y| \leq 1$, then M is a jumping finite automaton (JFA).

Configuration

upv where $u, v \in \Sigma^*$ and $p \in Q$.

Jump

$$xpyz \curvearrowright x'qz'$$

if $x, z, x', z' \in \Sigma^*$ such that $xz = x'z'$ and $(p, y, q) \in R$; $\curvearrowright^+$, $\curvearrowright^*$.

Accepted language

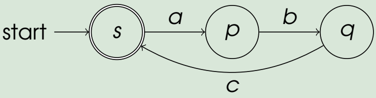
$L(M) = \{uv : u, v \in \Sigma^*, \; usv \curvearrowright^* f, \; f \in F\}$

Example automaton

$$M = (\{s, p, q\}, \{a, b, c\}, R, s, \{s\})$$

where R:

(s, a, p)
(p, b, q)
(q, c, s)

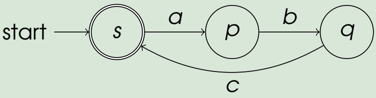


Example automaton

$$M = (\{s, p, q\}, \{a, b, c\}, R, s, \{s\})$$

where R:

(s, a, p)
(p, b, q)
(q, c, s)



Example input: *abbacc*

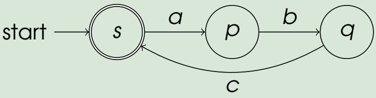
$abbsacc \curvearrowright abpbcc \curvearrowright abqcc \curvearrowright sabc \curvearrowright pbc \curvearrowright qc \curvearrowright s$

Example automaton

$$M = (\{s, p, q\}, \{a, b, c\}, R, s, \{s\})$$

where R:

(s, a, p)
(p, b, q)
(q, c, s)



Example input: *abbacc*

$abbsacc \curvearrowright abpbcc \curvearrowright abqcc \curvearrowright sabc \curvearrowright pbc \curvearrowright qc \curvearrowright s$

Resulting language

- FA: $L(M) = \{abc\}^*$
- JFA: $L(M) = \{w : w \in \{a, b, c\}^*, \ |w|_a = |w|_b = |w|_c\}$

- GJFA and JFA cannot guarantee the order of symbols between jumps.

- GJFA and JFA cannot guarantee the order of symbols between jumps.

Theorem

There is no GJFA M such that $L(M) = \{a\}^*\{b\}^*$.

- GJFA and JFA cannot guarantee the order of symbols between jumps.

Theorem

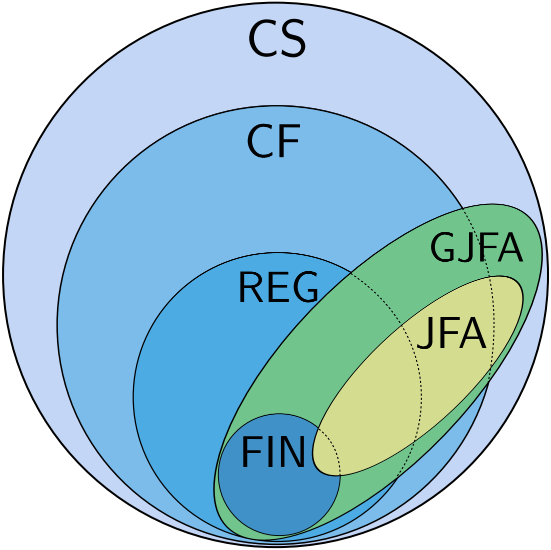
There is no GJFA M such that $L(M) = \{a\}^*\{b\}^*$.

Proof.

By contradiction. Let $K = \{a\}^*\{b\}^*$. Assume that there is a GJFA $M = (Q, \Sigma, R, s, F)$ such that $L(M) = K$.

Let $n = \max\{|y| : (p, y, q) \in R\}$ and $w = a^n b$.

When accepting w, a rule $(p, a^i b, q) \in R$, $0 \leq i < n$, has to be used. However, then M also accepts from the configuration $a^i bsa^{n-i}$ or $sa^i ba^{n-i}$. This implies that $a^i ba^{n-i} \in L(M)$. But that is a contradiction with the assumption that $L(M) = K$. Therefore, there is no GJFA M such that $L(M) = \{a\}^*\{b\}^*$. $\square$


CS
CF
GJFA
REG
JFA
FIN

Jumping $5' \to 3'$ Watson-Crick Finite Automata

Based on

- Radim Kocman, Benedek Nagy, Zbyněk Křivka, and Alexander Meduna
  A Jumping $5' \to 3'$ Watson-Crick Finite Automata Model
  Proceedings of NCMA 2018

- Radim Kocman, Zbyněk Křivka, Alexander Meduna, and Benedek Nagy
  A Jumping $5' \to 3'$ Watson-Crick Finite Automata Model
  Acta Informatica (in review)

Watson-Crick Finite Automata (WKA)

- biology-inspired model (the core model is similar to FA)
- work with the Watson-Crick tape (double-stranded tape, resembles DNA, the elements of the strands are pairwise complements of each other)
- uses two heads (one for each strand of the tape)

Watson-Crick Finite Automata (WKA)

- biology-inspired model (the core model is similar to FA)
- work with the Watson-Crick tape (double-stranded tape, resembles DNA, the elements of the strands are pairwise complements of each other)
- uses two heads (one for each strand of the tape)

$5' \to 3'$ Watson-Crick Finite Automata

- the heads read in the biochemical $5' \to 3'$ direction
- that is physically/mathematically in opposite directions

Watson-Crick Finite Automata (WKA)

- biology-inspired model (the core model is similar to FA)
- work with the Watson-Crick tape (double-stranded tape, resembles DNA, the elements of the strands are pairwise complements of each other)
- uses two heads (one for each strand of the tape)

$5' \rightarrow 3'$ Watson-Crick Finite Automata

- the heads read in the biochemical $5' \rightarrow 3'$ direction
- that is physically/mathematically in opposite directions

Sensing $5' \rightarrow 3'$ Watson-Crick Finite Automata

- the heads sense that they are meeting
- the processing of the input ends if for all pairs of the sequence one of the letters is read
- the tape notation is usually simplified: $\left[\begin{smallmatrix} A \\ T \end{smallmatrix}\right]$ as $a, \ldots$

Combined model

- the combination of GJFA and sensing $5' \rightarrow 3'$ WKA
- two heads as in sensing $5' \rightarrow 3'$ WKA
- each head can traverse the whole input in its direction
- all pairs of symbols are read only once

Expectations

- better accepting power than the non-combined models
- ability to model languages with some crossed agreements

Jumping $5' \to 3'$ WK Automaton

quintuple $M = (V, Q, q_0, F, \delta)$

V (Σ), Q, q_0 (s), F as in LFA, $V \cap \{\#\} = \emptyset$,

$\delta \colon (Q \times V^* \times V^* \times D) \to 2^Q$ (finite),
$D = \{\oplus, \ominus\}$ indicates the mutual position of heads.

Jumping $5' \to 3'$ WK Automaton

$$\textit{quintuple} \;\; M = (V, Q, q_0, F, \delta)$$

V (Σ), Q, q_0 (s), F as in LFA, $V \cap \{\#\} = \emptyset$,

$\delta\colon (Q \times V^* \times V^* \times D) \to 2^Q$ (finite),
$D = \{\oplus, \ominus\}$ indicates the mutual position of heads.

Configuration

$$(q, s, w_1, w_2, w_3)$$

- q is the state
- s is the position of heads
- w_1 is the unprocessed input before the first head
- w_2 is the unprocessed input between the heads
- w_3 is the unprocessed input after the second head

Steps

Let $x, y, u, v, w_2 \in V^*$ and $w_1, w_3 \in (V \cup \{\#\})^*$.

1. $\oplus$-reading: $(q, \oplus, w_1, xw_2y, w_3) \curvearrowright (q', s, w_1\{\#\}^{|x|}, w_2, \{\#\}^{|y|}w_3)$, where $q' \in \delta(q, x, y, \oplus)$, and s is either $\oplus$ if $|w_2| > 0$ or $\ominus$.
2. $\ominus$-reading: $(q, \ominus, w_1y, \varepsilon, xw_3) \curvearrowright (q', \ominus, w_1, \varepsilon, w_3)$, where $q' \in \delta(q, x, y, \ominus)$.
3. $\oplus$-jumping: $(q, \oplus, w_1, uw_2v, w_3) \curvearrowright (q, s, w_1u, w_2, vw_3)$, where s is either $\oplus$ if $|w_2| > 0$ or $\ominus$.
4. $\ominus$-jumping: $(q, \ominus, w_1\{\#\}^*, \varepsilon, \{\#\}^*w_3) \curvearrowright (q, \ominus, w_1, \varepsilon, w_3)$.

In the standard manner, define $\curvearrowright^+$ and $\curvearrowright^*$.

Steps

Let $x, y, u, v, w_2 \in V^*$ and $w_1, w_3 \in (V \cup \{\#\})^*$.

1. $\oplus$-reading: $(q, \oplus, w_1, xw_2y, w_3) \curvearrowright (q', s, w_1\{\#\}^{|x|}, w_2, \{\#\}^{|y|}w_3)$, where $q' \in \delta(q, x, y, \oplus)$, and s is either $\oplus$ if $|w_2| > 0$ or $\ominus$.
2. $\ominus$-reading: $(q, \ominus, w_1y, \varepsilon, xw_3) \curvearrowright (q', \ominus, w_1, \varepsilon, w_3)$, where $q' \in \delta(q, x, y, \ominus)$.
3. $\oplus$-jumping: $(q, \oplus, w_1, uw_2v, w_3) \curvearrowright (q, s, w_1u, w_2, vw_3)$, where s is either $\oplus$ if $|w_2| > 0$ or $\ominus$.
4. $\ominus$-jumping: $(q, \ominus, w_1\{\#\}^*, \varepsilon, \{\#\}^*w_3) \curvearrowright (q, \ominus, w_1, \varepsilon, w_3)$.

In the standard manner, define $\curvearrowright^+$ and $\curvearrowright^*$.

Accepted language

$L(M) = \{w \in V^* : (q_0, \oplus, \varepsilon, w, \varepsilon) \curvearrowright^* (q_f, \ominus, \varepsilon, \varepsilon, \varepsilon),\ q_f \in F\}$

Example automaton

$$M = (\{a, b\}, \{s\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, b, \oplus) = \{s\}$$
$$\delta(s, a, b, \ominus) = \{s\}$$

Example automaton

$$M = (\{a, b\}, \{s\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, b, \oplus) = \{s\}$$
$$\delta(s, a, b, \ominus) = \{s\}$$

Example input: *aaabbb*

$$
\begin{array}{ll}
(s, \oplus, \varepsilon, aaabbb, \varepsilon) \curvearrowright & \oplus\text{-reading} \\
(s, \oplus, \#, aabb, \#) \curvearrowright & \oplus\text{-reading} \\
(s, \oplus, \#\#, ab, \#\#) \curvearrowright & \oplus\text{-reading} \\
(s, \ominus, \#\#\#, \varepsilon, \#\#\#) \curvearrowright & \ominus\text{-jumping} \\
(s, \ominus, \varepsilon, \varepsilon, \varepsilon) &
\end{array}
$$

Example automaton

$$M = (\{a, b\}, \{s\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, b, \oplus) = \{s\}$$
$$\delta(s, a, b, \ominus) = \{s\}$$

Example input: *baabba*

$$\begin{array}{ll}
(s, \oplus, \varepsilon, baabba, \varepsilon) \curvearrowright & \oplus\text{-jumping} \\
(s, \oplus, b, aabb, a) \curvearrowright & \oplus\text{-reading} \\
(s, \oplus, b\#, ab, \#a) \curvearrowright & \oplus\text{-reading} \\
(s, \ominus, b\#\#, \varepsilon, \#\#a) \curvearrowright & \ominus\text{-jumping} \\
(s, \ominus, b, \varepsilon, a) \curvearrowright & \ominus\text{-reading} \\
(s, \ominus, \varepsilon, \varepsilon, \varepsilon) &
\end{array}$$

Example automaton

$$M = (\{a, b\}, \{s\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, b, \oplus) = \{s\}$$
$$\delta(s, a, b, \ominus) = \{s\}$$

Example input: *baabba*

$(s, \oplus, \varepsilon, baabba, \varepsilon) \curvearrowright$	$\oplus$-jumping
$(s, \oplus, b, aabb, a) \curvearrowright$	$\oplus$-reading
$(s, \oplus, b\#, ab, \#a) \curvearrowright$	$\oplus$-reading
$(s, \ominus, b\#\#, \varepsilon, \#\#a) \curvearrowright$	$\ominus$-jumping
$(s, \ominus, b, \varepsilon, a) \curvearrowright$	$\ominus$-reading
$(s, \ominus, \varepsilon, \varepsilon, \varepsilon)$	

Resulting language

$$L(M) = \{w : w \in \{a, b\}^*,\ |w|_a = |w|_b\}$$

- What happens if we remove $\delta(s, a, b, \ominus) = \{s\}$ from M?
  $\rightarrow L(M) = \{a^n b^n : n \geq 0\}$

- What happens if we remove $\delta(s, a, b, \ominus) = \{s\}$ from M?
  $\rightarrow L(M) = \{a^n b^n : n \geq 0\}$
- And if we use only $\delta(s, a, \varepsilon, \oplus) = \{s\}$ and $\delta(s, \varepsilon, b, \oplus) = \{s\}$?
  $\rightarrow L(M) = \{a\}^*\{b\}^*$

- What happens if we remove $\delta(s, a, b, \ominus) = \{s\}$ from M?
  $\rightarrow L(M) = \{a^n b^n : n \geq 0\}$
- And if we use only $\delta(s, a, \varepsilon, \oplus) = \{s\}$ and $\delta(s, \varepsilon, b, \oplus) = \{s\}$?
  $\rightarrow L(M) = \{a\}^*\{b\}^*$

- **REG** $\subset$ **JWK**
- **LIN** $\subset$ **JWK**
- $\{w_1 w_2 : w_1 \in \{a, b\}^*,\ w_2 \in \{c, d\}^*,\ |w_1|_a = |w_2|_c,\ |w_1|_b = |w_2|_d\} \in$ **JWK** which is a non-context-free language
- **JWK** $\subset$ **CS**

Theorem

There is no jumping $5' \to 3'$ WK automaton M such that $L(M) = \{a^n b^n c^n : n \geq 0\}$.

Theorem

There is no jumping $5' \to 3'$ WK automaton M such that $L(M) = \{a^n b^n c^n : n \geq 0\}$.

- Intuitively, the automaton needs to periodically remove symbols from three different positions in the input. But we have only two heads that can move in one direction.

Theorem

There is no jumping $5' \to 3'$ WK automaton M such that $L(M) = \{a^n b^n c^n : n \geq 0\}$.

- Intuitively, the automaton needs to periodically remove symbols from three different positions in the input. But we have only two heads that can move in one direction.
- How to rigorously prove it?

JWKFA – Undefinable Languages

FIT

Theorem

There is no jumping $5' \to 3'$ *WK automaton M such that* $L(M) = \{a^n b^n c^n : n \geq 0\}$.

- Intuitively, the automaton needs to periodically remove symbols from three different positions in the input. But we have only two heads that can move in one direction.
- How to rigorously prove it?
  - The automaton can guarantee the order of symbols in certain cases. We cannot use the JFA technique. ☹

JWKFA – Undefinable Languages

Theorem

There is no jumping $5' \rightarrow 3'$ WK automaton M such that $L(M) = \{a^n b^n c^n : n \geq 0\}$.

- Intuitively, the automaton needs to periodically remove symbols from three different positions in the input. But we have only two heads that can move in one direction.
- How to rigorously prove it?
  - The automaton can guarantee the order of symbols in certain cases. We cannot use the JFA technique. ☹
  - The symbols can be mixed so it is not easy to derive a meaningful pumping lemma. ☹

Theorem

There is no jumping $5' \to 3'$ WK automaton M such that $L(M) = \{a^n b^n c^n : n \geq 0\}$.

- Intuitively, the automaton needs to periodically remove symbols from three different positions in the input. But we have only two heads that can move in one direction.
- How to rigorously prove it?
  - The automaton can guarantee the order of symbols in certain cases. We cannot use the JFA technique. ☹
  - The symbols can be mixed so it is not easy to derive a meaningful pumping lemma. ☹
  - We need a different proof technique:
    $\to$ introducing the new debt lemma.

Parikh Vector

The Parikh vector associated to a string $x \in V^*$ with respect to the alphabet $V = \{a_1, a_2, \ldots, a_n\}$ is
$\Psi_V(x) = (|x|_{a_1}, |x|_{a_2}, \ldots, |x|_{a_n})$.

For $L \subseteq V^*$ we define $\Psi_V(L) = \{\Psi_V(x) : x \in L\}$.

Parikh Vector

The Parikh vector associated to a string $x \in V^*$ with respect to the alphabet $V = \{a_1, a_2, \dots, a_n\}$ is $\Psi_V(x) = (|x|_{a_1}, |x|_{a_2}, \dots, |x|_{a_n})$.

For $L \subseteq V^*$ we define $\Psi_V(L) = \{\Psi_V(x) : x \in L\}$.

Example strings

$$
\begin{array}{lll}
V = \{a, b, c\}, & x = abbccc & \Rightarrow \Psi_V(x) = (1, 2, 3) \\
V = \{a, b, c, d\}, & x = abbccc & \Rightarrow \Psi_V(x) = (1, 2, 3, 0) \\
V = \{a, b, c, d\}, & x = cbabcc & \Rightarrow \Psi_V(x) = (1, 2, 3, 0) \\
V = \{a, b, c, d\}, & x = \varepsilon & \Rightarrow \Psi_V(x) = (0, 0, 0, 0)
\end{array}
$$

Parikh Vector

The Parikh vector associated to a string $x \in V^*$ with respect to the alphabet $V = \{a_1, a_2, \ldots, a_n\}$ is
$\Psi_V(x) = (|x|_{a_1}, |x|_{a_2}, \ldots, |x|_{a_n})$.

For $L \subseteq V^*$ we define $\Psi_V(L) = \{\Psi_V(x) : x \in L\}$.

Example language

Let $V = \{a, b, c\}$ and $L = \{a^n b^n c^n : n \geq 0\}$. Then, $\Psi_V(L) = \{$

$$
\begin{array}{lcl}
x = \varepsilon & \Rightarrow & \Psi_V(x) = (0,0,0) \\
x = abc & \Rightarrow & \Psi_V(x) = (1,1,1) \\
x = aabbcc & \Rightarrow & \Psi_V(x) = (2,2,2) \\
x = aaabbbccc & \Rightarrow & \Psi_V(x) = (3,3,3) \\
\ldots & &
\end{array}
$$

$\} = \{(0,0,0), (1,1,1), (2,2,2), (3,3,3), \ldots\} = \{(n,n,n) : n \geq 0\}$.

Definition

Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \rightarrow 3'$ WK automaton, where $V = \{a_1, \dots, a_n\}$. Following the computation of M on an input $w \in V^*$, let $o = (o_1, \dots, o_n)$ be the Parikh vector built by the processed (read) symbols from w: At first, for the starting configuration, set $o = \Psi_V(\varepsilon)$. For the following configurations, whenever M makes a $\oplus/\ominus$-reading step from some q to q' according to $q' \in \delta(q, u, v, s)$, set $o = o + \Psi_V(uv)$. Using the Parikh mapping of $L(M)$, we define $\Delta(o) = \{\sum_{i=1}^{n}(m_i - o_i) : (m_1, \dots, m_n) \in \Psi_V(L(M)),\ m_i \geq o_i,\ 1 \leq i \leq n\} \cup \{\infty\}$. Finally, we define the debt of the current configuration of M as $\min \Delta(o)$.

Definition

Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton, where $V = \{a_1, \ldots, a_n\}$. Following the computation of M on an input $w \in V^*$, let $o = (o_1, \ldots, o_n)$ be the Parikh vector built by the processed (read) symbols from w: At first, for the starting configuration, set $o = \Psi_V(\varepsilon)$. For the following configurations, whenever M makes a $\oplus/\ominus$-reading step from some q to q' according to $q' \in \delta(q, u, v, s)$, set $o = o + \Psi_V(uv)$. Using the Parikh mapping of $L(M)$, we define $\Delta(o) = \{\sum_{i=1}^{n}(m_i - o_i) : (m_1, \ldots, m_n) \in \Psi_V(L(M)),\ m_i \geq o_i,\ 1 \leq i \leq n\} \cup \{\infty\}$. Finally, we define the debt of the current configuration of M as $\min \Delta(o)$.

1. We are counting the processed symbols in the Parikh Vector $o = (o_1, \ldots, o_n)$.
2. The debt of the current configuration of M is the minimum number of symbols that we need to add to o so that it matches some Parikh vector from $\Psi_V(L(M))$.

Example automaton

Let $V = \{a, b, c\}$. Assume that there is a jumping $5' \to 3'$ WK automaton $M = (V, Q, q_0, F, \delta)$ such that $L(M) = \{a^n b^n c^n : n \geq 0\}$.

Therefore, $\Psi_V(L(M)) = \{(n, n, n) : n \geq 0\}$.

Example steps

$$
\begin{array}{lll}
(s, \oplus, \varepsilon, aabbcc, \varepsilon) \curvearrowright & o = (0,0,0) & \min \Delta(o) = 0 \\
(?, \oplus, \#, abbcc, \varepsilon) \curvearrowright & o = (1,0,0) & \min \Delta(o) = 2 \\
(?, \oplus, \#a, bbcc, \varepsilon) \curvearrowright & o = (1,0,0) & \min \Delta(o) = 2 \\
(?, \oplus, \#a\#, bc, \#) \curvearrowright & o = (1,1,1) & \min \Delta(o) = 0 \\
(?, \ominus, \#a\#\#, \varepsilon, \#\#) \curvearrowright & o = (1,2,2) & \min \Delta(o) = 1 \\
(?, \ominus, \#a, \varepsilon, \varepsilon) \curvearrowright & o = (1,2,2) & \min \Delta(o) = 1 \\
(?, \ominus, \#, \varepsilon, \varepsilon) \curvearrowright & o = (2,2,2) & \min \Delta(o) = 0 \\
(?, \ominus, \#, \varepsilon, \varepsilon) \curvearrowright & o = (2,2,2) & \min \Delta(o) = 0 \\
(?, \ominus, \varepsilon, \varepsilon, \varepsilon) \curvearrowright & o = (2,2,2) & \min \Delta(o) = 0
\end{array}
$$

Debt lemma

Let L be a language, and let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \rightarrow 3'$ WK automaton. If $L(M) = L$, M accepts all $w \in L$ using only configurations that have their debt bounded by some constant k for M.

Debt lemma

Let L be a language, and let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \rightarrow 3'$ WK automaton. If $L(M) = L$, M accepts all $w \in L$ using only configurations that have their debt bounded by some constant k for M.

Example automaton

$$M = (\{a, b\}, \{s\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, b, \oplus) = \{s\}$$
$$\delta(s, a, b, \ominus) = \{s\}$$

$$L(M) = \{w : w \in \{a, b\}^*, \ |w|_a = |w|_b\}$$

Debt lemma

Let L be a language, and let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton. If $L(M) = L$, M accepts all $w \in L$ using only configurations that have their debt bounded by some constant k for M.

Example automaton

$$M = (\{a, b\}, \{s\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, b, \oplus) = \{s\}$$
$$\delta(s, a, b, \ominus) = \{s\}$$

$L(M) = \{w : w \in \{a, b\}^*, \ |w|_a = |w|_b\}$

$k = 0$ is sufficient ☺

You can go to Bonus for the proof.

Theorem

There is no jumping $5' \rightarrow 3'$ WK automaton M such that $L(M) = \{a^n b^n c^n : n \geq 0\}$.

FIT

Theorem

There is no jumping $5' \to 3'$ WK automaton M such that $L(M) = \{a^n b^n c^n : n \geq 0\}$.

Proof (1/3).

Basic idea. Considering any sufficiently large constant k, we show that M cannot process all symbols of $a^{10k} b^{10k} c^{10k}$ using only configurations that have their debt bounded by k.

Formal proof. (sketch) By contradiction. Let $L = \{a^n b^n c^n : n \geq 0\}$, and let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton such that $L(M) = L$.

Consider any k such that $k > \max\{|uv| : \delta(q, u, v, s) \neq \emptyset,\ u, v \in V^*\}$.

Represent the debt of the configuration as $\langle d_a, d_b, d_c \rangle$.

For all traversed configurations must hold $d_a + d_b + d_c \leq k$.

Let $w = a^{10k} b^{10k} c^{10k}$.

Theorem

There is no jumping $5' \rightarrow 3'$ *WK automaton* M *such that* $L(M) = \{a^n b^n c^n : n \geq 0\}$.

Proof (2/3).

First, we explore the maximum number of symbols that M can read from w before the heads meet. Starting from $(q_0, \oplus, \varepsilon, w, \varepsilon)$ $\langle 0, 0, 0 \rangle$ and until the position $\ominus$ is reached. Consider the optimal reading strategy to process the maximum number of symbols from $a^{10k} b^{10k} c^{10k}$:

1. M processes (with multiple steps) a^k and c^k and reaches $\langle 0, k, 0 \rangle$,
2. M reads l symbols together in one step (balanced number of a's, b's, and c's) while keeping $\langle 0, k, 0 \rangle$, $l < k$,
3. M processes b^{2k} and a^k (or c^k) and reaches $\langle 0, 0, k \rangle$ (or $\langle k, 0, 0 \rangle$).

No further reading is possible; this strategy processed $5k + l$ symbols.

Theorem

There is no jumping $5' \rightarrow 3'$ *WK automaton M such that* $L(M) = \{a^n b^n c^n : n \geq 0\}$.

Proof (3/3).

Second, when the heads meet, $a^{>4k} b^{>4k} c^{>4k}$ has yet to be processed. Consider one of the optimal reading strategies:

1. the heads are between b's and c's,
2. the debt of the current configuration is $\langle 0, k, 0 \rangle$,
3. M processes b^{2k} and c^k and reaches $\langle k, 0, 0 \rangle$.

No further reading is possible; this strategy processed $3k$ symbols.

M is not able to process more than $8k + l$ symbols; but the input contains $30k$ symbols. Consequently, there is no constant k that bounds the debt of configurations of M. $\square$

Theorem

There is no jumping $5' \rightarrow 3'$ WK automaton M such that $L(M) = \{w \in \{a, b, c\}^* : |w|_a = |w|_b = |w|_c\}$.

Proof (1/10).

...

Theorem

There is no jumping $5' \to 3'$ WK automaton M such that $L(M) = \{w \in \{a,b,c\}^* : |w|_a = |w|_b = |w|_c\}$.

Proof (1/10).

. . . **NO**

Proof.

$\Psi_V(\{w \in \{a,b,c\}^* : |w|_a = |w|_b = |w|_c\}) = \Psi_V(\{a^n b^n c^n : n \geq 0\})$

$w = a^{10k} b^{10k} c^{10k}$

Since the debt depends only on o and Ψ_V,
the proof is analogous. $\square$

JWKFA – Language Families (1/2)

- **JWK** is incomparable with **GJFA** and **JFA**.
- **JWK** and **CF** are incomparable.

- **JWK** is incomparable with **GJFA** and **JFA**.
- **JWK** and **CF** are incomparable.

Restrictions

N stateless, i.e., with only one state: if $Q = F = \{q_0\}$

F all-final, i.e., with only final states: if $Q = F$

S simple (at most one head moves in a step)

1 1-limited (exactly one letter is being read in a step)

Further variations such as **NS**, **FS**, **N1**, and **F1** WK automata can be identified in a straightforward way by using multiple constraints.

JWKFA – Language Families (2/2)

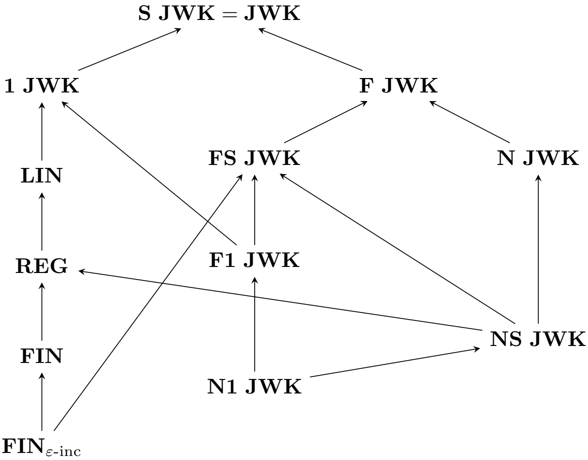


Figure: If there is an arrow from family X to family Y in the figure, then $X \subset Y$. Furthermore, if there is no path (following the arrows) between families X and Y, then X and Y are incomparable.

Conclusion

- The debt lemma was used only in JWKFAs so far.
- It can work in any automaton model that reads at least semi-continuously and where the steps depend only on the current state (not the previous readings, e.g., no stack).
- It can work in FAs.

Welcome at the end of this presentation!

And now Bonus. . .

Definition

Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \rightarrow 3'$ WK automaton. Assuming some states $q, q' \in Q$ and a mutual position of heads $s \in \{\oplus, \ominus\}$, we say that q' is reachable from q and s if there exists a configuration (q, s, w_1, w_2, w_3) such that $(q, s, w_1, w_2, w_3) \curvearrowright^* (q', s', w_1', w_2', w_3')$ in M, $s' \in \{\oplus, \ominus\}$, $w_1, w_2, w_3, w_1', w_2', w_3' \in (V \cup \{\#\})^*$.

Definition

Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton. Assuming some states $q, q' \in Q$ and a mutual position of heads $s \in \{\oplus, \ominus\}$, we say that q' is reachable from q and s if there exists a configuration (q, s, w_1, w_2, w_3) such that $(q, s, w_1, w_2, w_3) \curvearrowright^* (q', s', w_1', w_2', w_3')$ in M, $s' \in \{\oplus, \ominus\}$, $w_1, w_2, w_3, w_1', w_2', w_3' \in (V \cup \{\#\})^*$.

Example automaton

$$M = (\{a\}, \{s, p, q\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, \varepsilon, \oplus) = \{p\}$$
$$\delta(s, a, \varepsilon, \ominus) = \{q\}$$

Definition

Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton. Assuming some states $q, q' \in Q$ and a mutual position of heads $s \in \{\oplus, \ominus\}$, we say that q' is reachable from q and s if there exists a configuration (q, s, w_1, w_2, w_3) such that $(q, s, w_1, w_2, w_3) \curvearrowright^* (q', s', w_1', w_2', w_3')$ in M, $s' \in \{\oplus, \ominus\}$, $w_1, w_2, w_3, w_1', w_2', w_3' \in (V \cup \{\#\})^*$.

Example automaton

$$M = (\{a\}, \{s, p, q\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, \varepsilon, \oplus) = \{p\}$$
$$\delta(s, a, \varepsilon, \ominus) = \{q\}$$

p is reachable from s and $\oplus$

Definition

Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton. Assuming some states $q, q' \in Q$ and a mutual position of heads $s \in \{\oplus, \ominus\}$, we say that q' is reachable from q and s if there exists a configuration (q, s, w_1, w_2, w_3) such that $(q, s, w_1, w_2, w_3) \curvearrowright^* (q', s', w_1', w_2', w_3')$ in M, $s' \in \{\oplus, \ominus\}$, $w_1, w_2, w_3, w_1', w_2', w_3' \in (V \cup \{\#\})^*$.

Example automaton

$$M = (\{a\}, \{s, p, q\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, \varepsilon, \oplus) = \{p\}$$
$$\delta(s, a, \varepsilon, \ominus) = \{q\}$$

p is reachable from s and $\oplus$
p is not reachable from s and $\ominus$

Definition

Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton. Assuming some states $q, q' \in Q$ and a mutual position of heads $s \in \{\oplus, \ominus\}$, we say that q' is reachable from q and s if there exists a configuration (q, s, w_1, w_2, w_3) such that $(q, s, w_1, w_2, w_3) \curvearrowright^* (q', s', w_1', w_2', w_3')$ in M, $s' \in \{\oplus, \ominus\}$, $w_1, w_2, w_3, w_1', w_2', w_3' \in (V \cup \{\#\})^*$.

Example automaton

$$M = (\{a\}, \{s, p, q\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, \varepsilon, \oplus) = \{p\}$$
$$\delta(s, a, \varepsilon, \ominus) = \{q\}$$

p is reachable from s and $\oplus$
p is not reachable from s and $\ominus$
q is reachable from s and $\oplus$

Definition

Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \rightarrow 3'$ WK automaton. Assuming some states $q, q' \in Q$ and a mutual position of heads $s \in \{\oplus, \ominus\}$, we say that q' is reachable from q and s if there exists a configuration (q, s, w_1, w_2, w_3) such that $(q, s, w_1, w_2, w_3) \curvearrowright^* (q', s', w_1', w_2', w_3')$ in M, $s' \in \{\oplus, \ominus\}$, $w_1, w_2, w_3, w_1', w_2', w_3' \in (V \cup \{\#\})^*$.

Example automaton

$$M = (\{a\}, \{s, p, q\}, s, \{s\}, \delta)$$

where δ:

$$\delta(s, a, \varepsilon, \oplus) = \{p\}$$
$$\delta(s, a, \varepsilon, \ominus) = \{q\}$$

p is reachable from s and $\oplus$
p is not reachable from s and $\ominus$
q is reachable from s and $\oplus$
q is reachable from s and $\ominus$

Lemma

Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton, and let $q \in Q$ and $s \in \{\oplus, \ominus\}$ such that $f \in F$ is reachable from q and s. When $(q_0, \oplus, \varepsilon, w, \varepsilon) \curvearrowright^* (q, s, w_1, w_2, w_3)$ in M, $w \in V^*$, $w_1, w_2, w_3 \in (V \cup \{\#\})^*$, there exists $w' \in L(M)$ such that M starting with w' can reach q and s' ($s' = s$ or $s' = \ominus$) by using the same sequence of $\oplus/\ominus$-reading steps as in $(q_0, \oplus, \varepsilon, w, \varepsilon) \curvearrowright^* (q, s, w_1, w_2, w_3)$ and the rest of w' can be processed with a limited number of steps bounded by some constant k for M.

Lemma

Let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton, and let $q \in Q$ and $s \in \{\oplus, \ominus\}$ such that $f \in F$ is reachable from q and s. When $(q_0, \oplus, \varepsilon, w, \varepsilon) \curvearrowright^* (q, s, w_1, w_2, w_3)$ in M, $w \in V^*$, $w_1, w_2, w_3 \in (V \cup \{\#\})^*$, there exists $w' \in L(M)$ such that M starting with w' can reach q and s' ($s' = s$ or $s' = \ominus$) by using the same sequence of $\oplus/\ominus$-reading steps as in $(q_0, \oplus, \varepsilon, w, \varepsilon) \curvearrowright^* (q, s, w_1, w_2, w_3)$ and the rest of w' can be processed with a limited number of steps bounded by some constant k for M.

1. On a string w with a sequence of steps we reach q and s.
2. A final state is reachable from q and s.
3. There exists some string w' such that we can reach q and s' with the same sequence of steps.
4. We can finish accepting w' with a limited number of additional steps.

Proof.

(idea)

(1) If f is reachable from q and s, there has to exist a sequence of state transitions from $(Q \times \{\oplus, \ominus\})^+$ such that $(p_0, s_0) \cdots (p_n, s_n)$, $p_0 = q$, $s_0 = s'$, $p_n = f$, $s_n = \ominus$, all pairs are unique, ...
This sequence has to be finite and bounded by some constant.

(2) Represent the complete sequence as $(p_0, s_0) \cdots (p_m, s_m)$. At first, for all $i = 0, \dots, m$, set $a_i = \varepsilon$, $b_i = \varepsilon$, $c_i = \varepsilon$, $d_i = \varepsilon$. If $p_{i+1} \in \delta(p_i, u_i, v_i, s_i)$ is used, then if $s_i = \oplus$, set $a_i = u_i$ and $b_i = v_i$, otherwise if $s_i = \ominus$, set $c_i = u_i$ and $d_i = v_i$.

(3) $w' = a_0 \cdots a_m d_m \cdots d_0 c_0 \cdots c_m b_m \cdots b_0 \in L(M)$ □

JWKFA – Why the Debt Lemma Holds (4/4)

Debt lemma

Let L be a language, and let $M = (V, Q, q_0, F, \delta)$ be a jumping $5' \to 3'$ WK automaton. If $L(M) = L$, M accepts all $w \in L$ using only configurations that have their debt bounded by some constant k for M.

Proof.

(idea) By contradiction.
(1) Assume that M does not accept all $w \in L$ exclusively using only configurations that have their debt bounded by some constant k for M, then M can accept some $w \in L$ over a configuration for which the debt cannot be bounded by any k.

(2) Due to previous lemmas, if final state is reachable there is some w' such that $\min \Delta(o)$ must be bounded by some constant.

(3) M cannot accept w over a state q and a mutual position of heads s from which no final state $f \in F$ is reachable.

(4) Consequently, when M accepts w, it must be done over configurations with the debt $\leq k$. But that is a contradiction. □