Jumping Finite Automata: New Results

Part One: Solved Questions

Radim Kocman and Zbyněk Křivka

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

{ikocman,krivka}@fit.vutbr.cz

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

LTA 2016 (December 5, 2016)

Table of Contents

Part One Based on ...

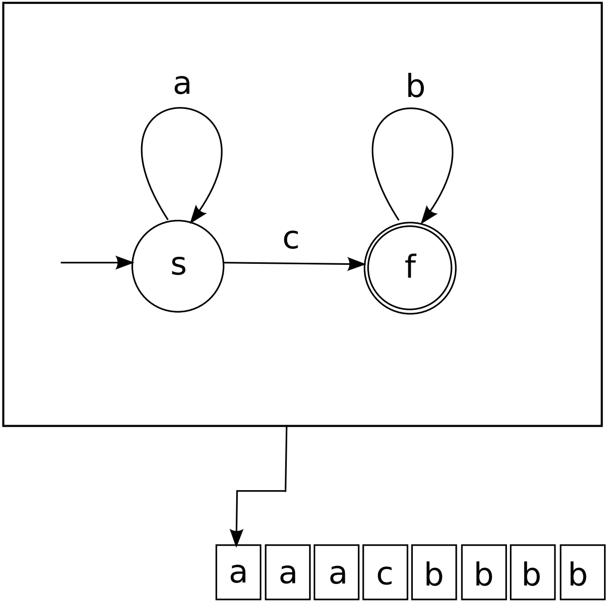
Alexander Meduna, Petr Zemek: Jumping Finite Automata. Int. J. Found. Comput. Sci. 23(7): 1555-1578 (2012)

Vojtěch Vorel: Two Results on Discontinuous Input Processing. DCFS 2016: 205-216

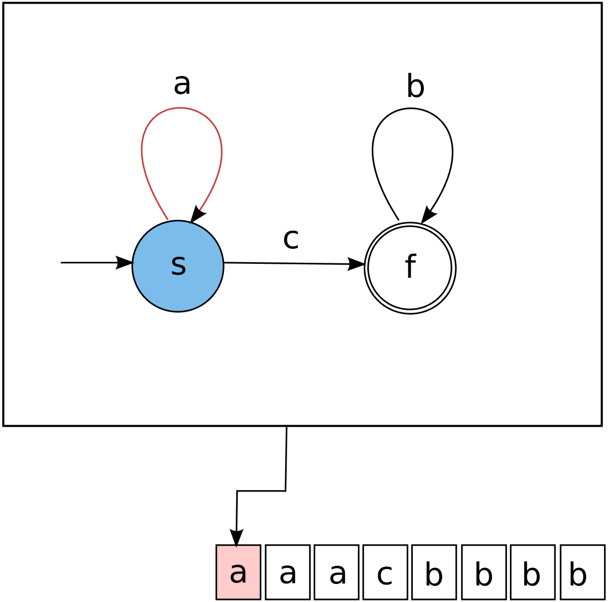
Vojtěch Vorel: On Basic Properties of Jumping Finite Automata. Int. J. Found. Comput. Sci. (conditionally accepted; 2015)

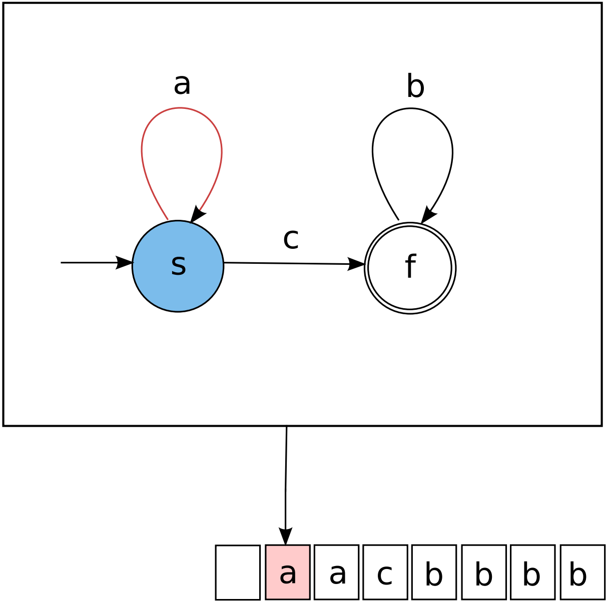
Henning Fernau, Meenakshi Paramasivan, Markus L. Schmid, Vojtěch Vorel: Characterization and Complexity Results on Jumping Finite Automata. Theoret. Comput. Sci. (in press, 2016)

Classical Finite Automata

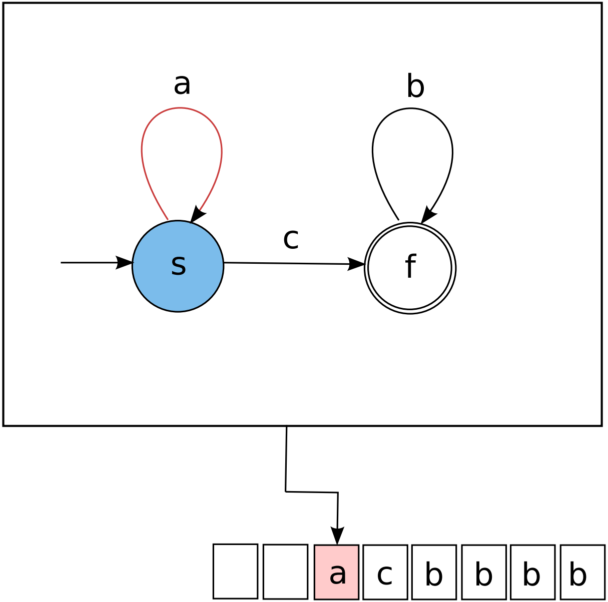


Classical Finite Automata

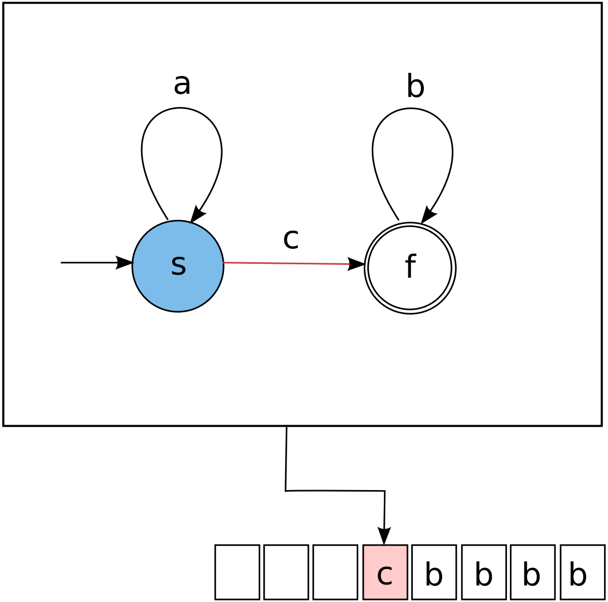


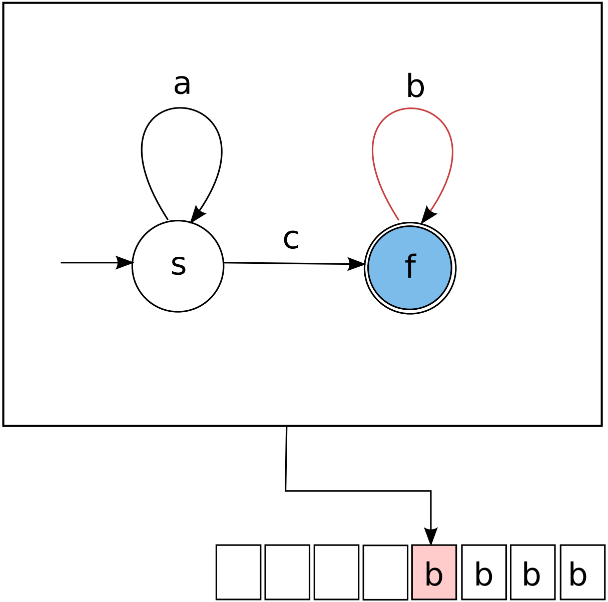

a
b
c
s
f
a
a
c
b
b
b
b

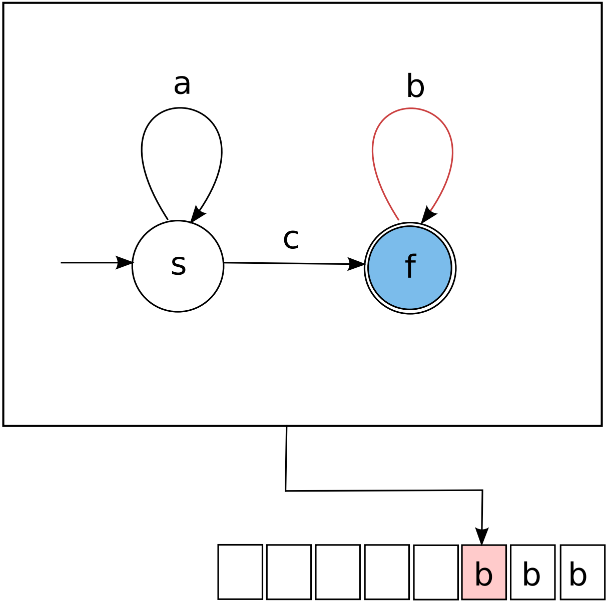
Classical Finite Automata



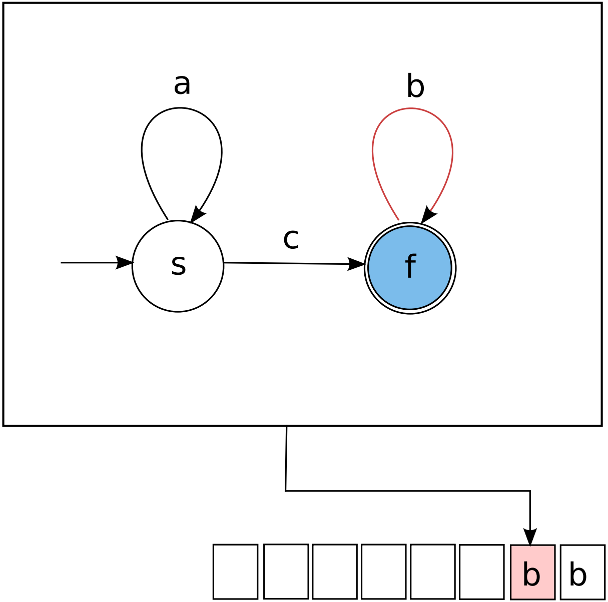
Classical Finite Automata



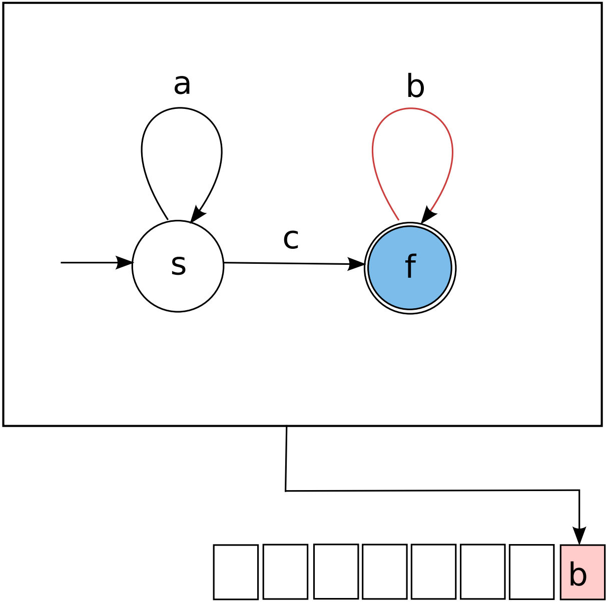

a
b
s
c
f
b
b
b
b

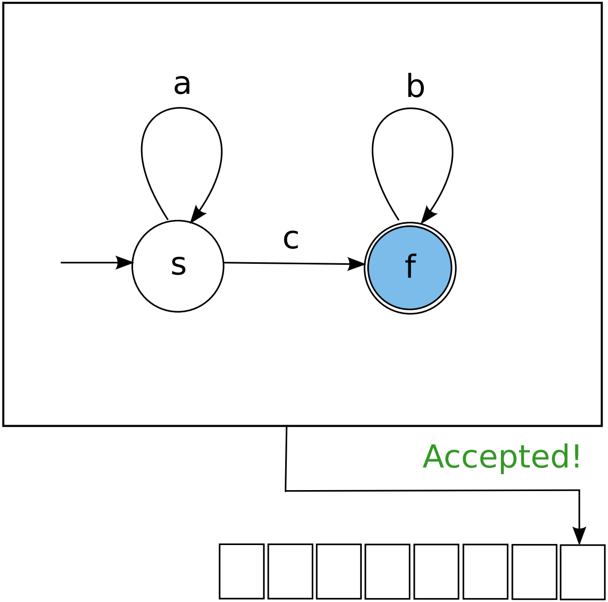

a
b
c
s
f
b
b
b

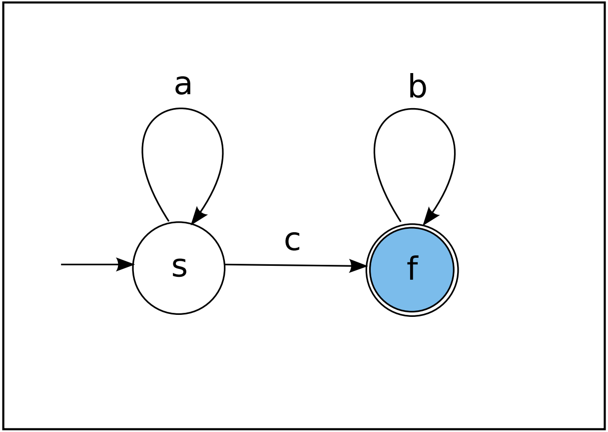
Classical Finite Automata



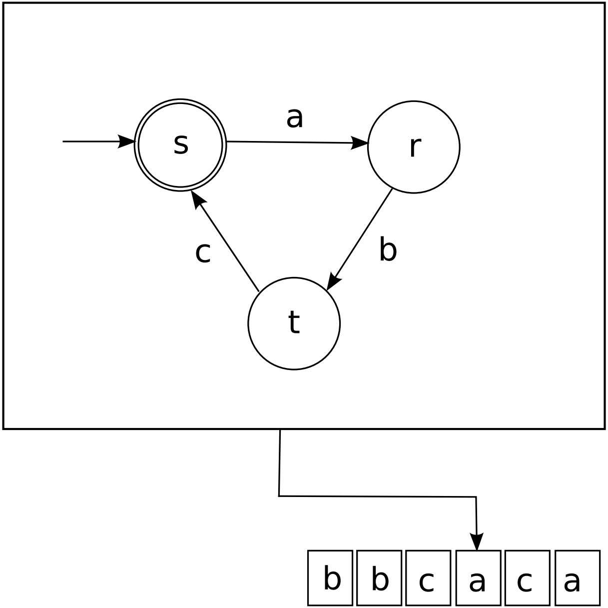
Classical Finite Automata

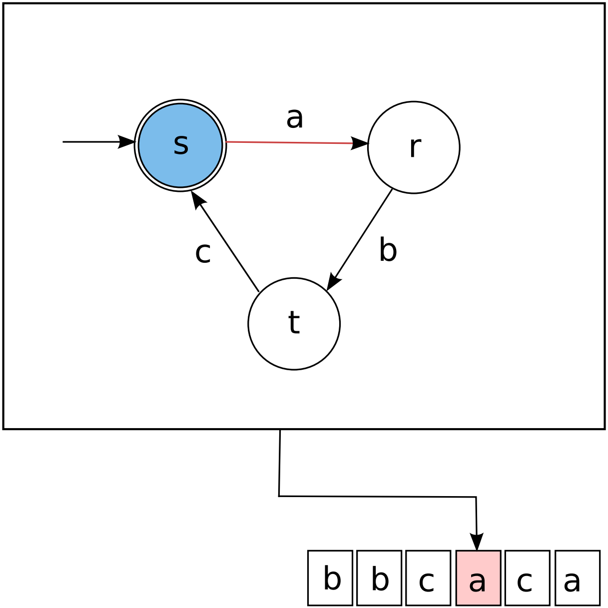



a
b
c
s
f
Accepted!

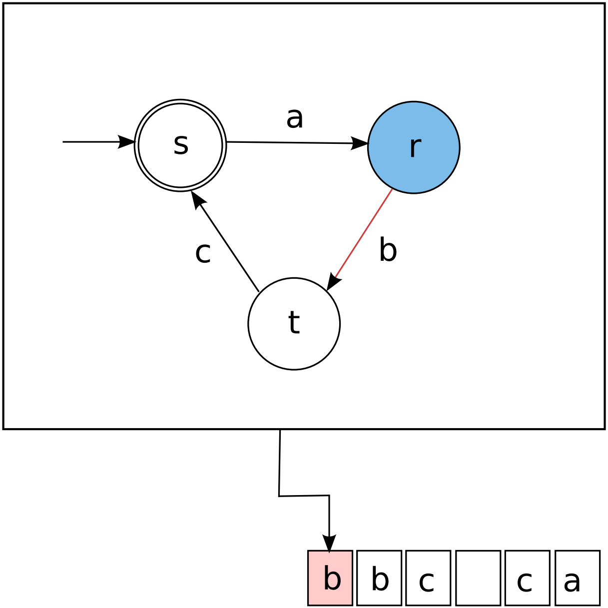


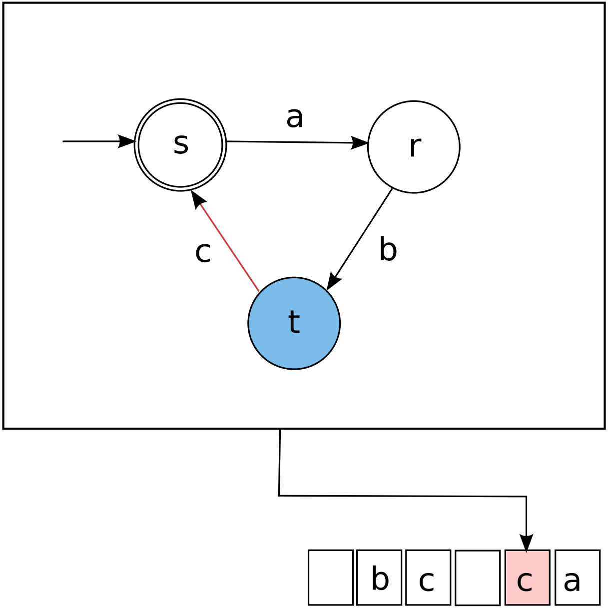
Accepted language: $\{a\}^*\{c\}\{b\}^*$


a
s
r
b
c
t
b
b
c
a
c
a

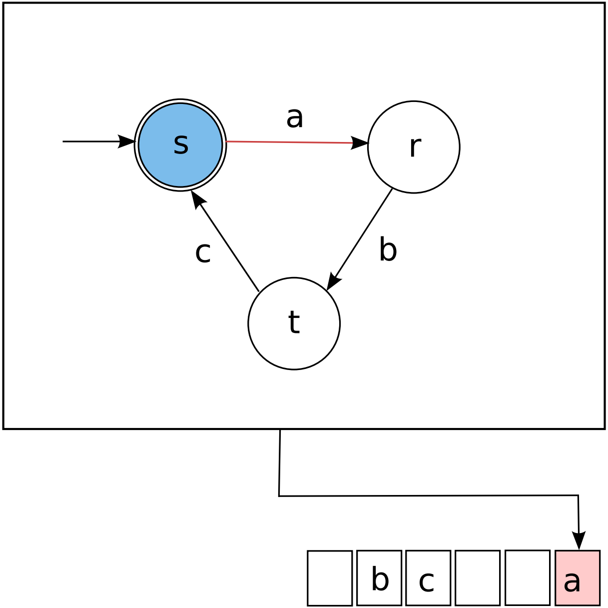

a
s
r
c
b
t
b
b
c
a
c
a

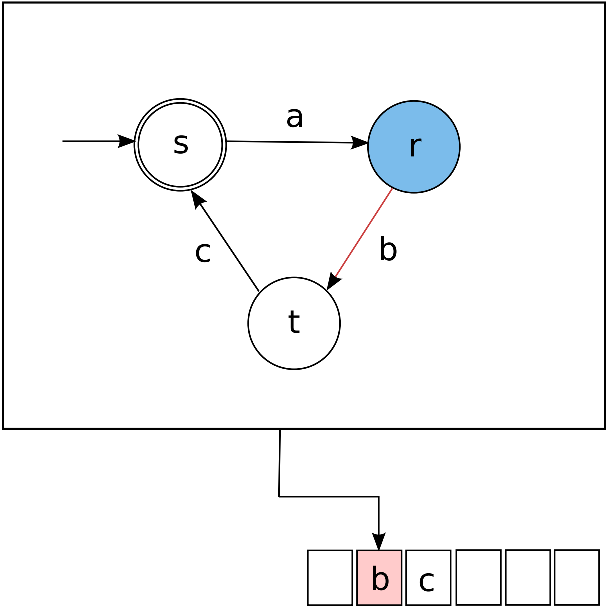
Jumping Finite Automata



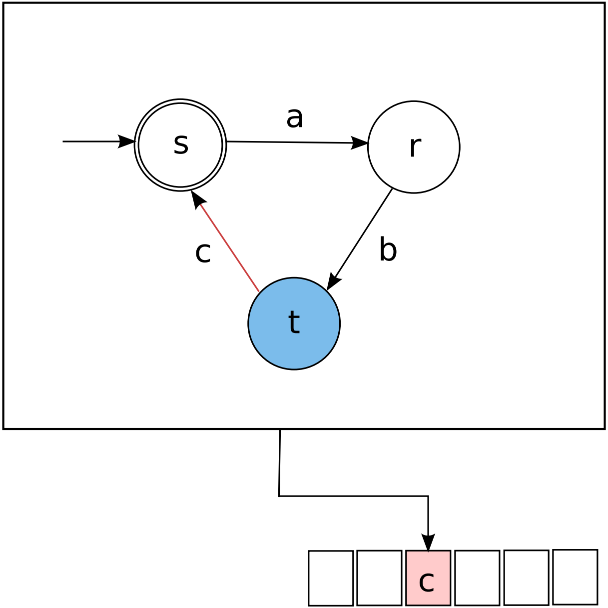


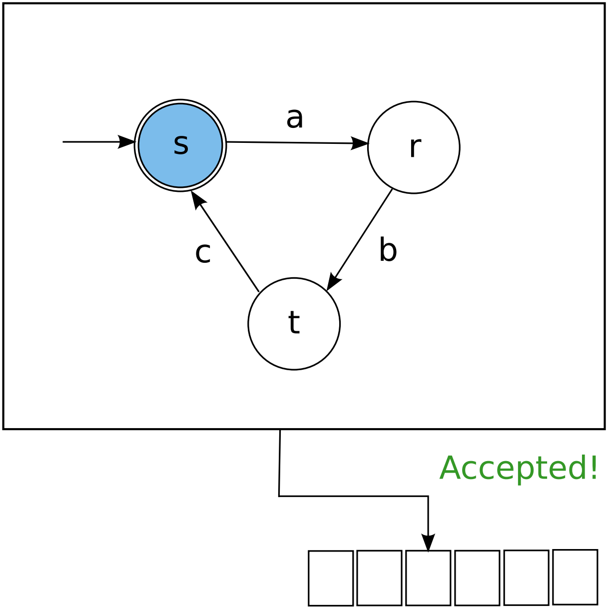
s
a
r
c
b
t
b
c
c
a

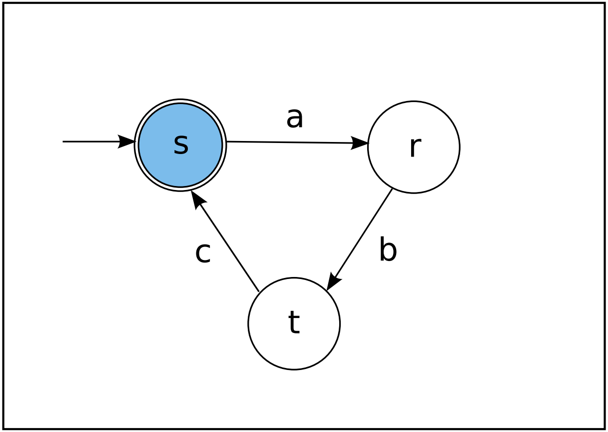

a
s
r
c
b
t
b
c
a


a
s
r
c
b
t
b
c

Jumping Finite Automata




s
a
r
c
b
t
Accepted!



Accepted language: $\{w \in \{a, b, c\}^* : |w|_a = |w|_b = |w|_c\}$

Definition (Meduna, Zemek (2012))

A general jumping finite automaton (GJFA) is a quintuple

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

where

- Q is a finite set of states;
- Σ is the input alphabet;
- R is a finite set of rules of the form

$$py \rightarrow q \qquad (p, q \in Q, y \in \Sigma^*)$$

- s is the start state;
- F is a set of final states.

Definition (Meduna, Zemek (2012))

A general jumping finite automaton (GJFA) is a quintuple

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

where

- Q is a finite set of states;
- Σ is the input alphabet;
- R is a finite set of rules of the form

$$py \to q \qquad (p, q \in Q, y \in \Sigma^*)$$

- s is the start state;
- F is a set of final states.

Definition

If all rules $py \to q \in R$ satisfy $|y| \leq 1$, then M is a jumping finite automaton (JFA).

Definition

If $x, z, x', z', y \in \Sigma^*$ such that $xz = x'z'$ and $py \rightarrow q \in R$, then M makes a jump from $xpyz$ to $x'qz'$, symbolically written as

$$x\underline{p}yz \curvearrowright x'\underline{q}z'$$

Definition

If $x, z, x', z', y \in \Sigma^*$ such that $xz = x'z'$ and $py \to q \in R$, then M makes a jump from $xpyz$ to $x'qz'$, symbolically written as

$$x\underline{p}yz \curvearrowright x'\underline{q}z'$$

$\curvearrowright^*$ intuitively, a sequence of jumps (possibly empty); mathematically, the reflexive-transitive closure of $\curvearrowright$

Definition

If $x, z, x', z', y \in \Sigma^*$ such that $xz = x'z'$ and $py \rightarrow q \in R$, then M makes a jump from $xpyz$ to $x'qz'$, symbolically written as

$$x\underline{p}yz \curvearrowright x'\underline{q}z'$$

$\curvearrowright^*$ intuitively, a sequence of jumps (possibly empty); mathematically, the reflexive-transitive closure of $\curvearrowright$

Definition

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

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

Note: Hereafter, a family of languages defined by model X is denoted by $\mathscr{L}(X)$.

Example

The JFA

$$M = (\{s, r, t\}, \{a, b, c\}, R, s, \{s\})$$

with

$$R = \{sa \to r, rb \to t, tc \to s\}$$

accepts

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

For instance:

$$\begin{array}{llll} bacbc\underline{s}a & \curvearrowright & bac\underline{r}bc & [sa \to r] \\ & \curvearrowright & bac\underline{t}c & [rb \to t] \\ & \curvearrowright & b\underline{s}ac & [tc \to s] \\ & \curvearrowright & \underline{r}bc & [sa \to r] \\ & \curvearrowright & \underline{t}c & [rb \to t] \\ & \curvearrowright & \underline{s} & [tc \to s] \end{array}$$

Example

The GJFA

$$H = (\{s, f\}, \{a, b\}, R, s, \{f\}),$$

with

$$R = \{sba \rightarrow f, fa \rightarrow f, fb \rightarrow f\}$$

accepts

$$L(H) = \{a, b\}^* \{ba\} \{a, b\}^*$$

For instance:

$$\begin{array}{llll} bb\underline{s}baa & \curvearrowright & bb\underline{f}a & [sba \rightarrow f] \\ & \curvearrowright & \underline{f}bb & [fa \rightarrow f] \\ & \curvearrowright & \underline{f}b & [fb \rightarrow f] \\ & \curvearrowright & \underline{f} & [fb \rightarrow f] \end{array}$$

Definition

The shuffle operation, denoted by $\sqcup\!\sqcup$, is defined by

$$u \sqcup\!\sqcup v = \left\{ x_1 y_1 x_2 y_2 \dots x_n y_n \,:\, \begin{array}{c} u = x_1 x_2 \dots x_n, v = y_1 y_2 \dots y_n \\ x_i, y_i \in \Sigma^*, 1 \leq i \leq n, n \geq 1 \end{array} \right\},$$

$$L_1 \sqcup\!\sqcup L_2 = \bigcup_{u \in L_1, v \in L_2} (u \sqcup\!\sqcup v),$$

for $u, v \in \Sigma^*$ and $L_1, L_2 \subseteq \Sigma^*$.

Definition

The shuffle operation, denoted by $\sqcup\!\sqcup$, is defined by

$$u \sqcup\!\sqcup v = \left\{ x_1 y_1 x_2 y_2 \ldots x_n y_n \; : \; \begin{array}{r} u = x_1 x_2 \ldots x_n, v = y_1 y_2 \ldots y_n \\ x_i, y_i \in \Sigma^*, 1 \leq i \leq n, n \geq 1 \end{array} \right\},$$

$$L_1 \sqcup\!\sqcup L_2 = \bigcup_{u \in L_1, v \in L_2} (u \sqcup\!\sqcup v),$$

for $u, v \in \Sigma^*$ and $L_1, L_2 \subseteq \Sigma^*$.

Example

$$ab \sqcup\!\sqcup cd = \{abcd, acdb, cdab, acbd, cadb, cabd\}$$

Definition

For $L \subseteq \Sigma^*$, the iterated shuffle of L is

$$L^{\sqcup\!\sqcup,*} = \bigcup_{n=0}^{\infty} L^{\sqcup\!\sqcup,n},$$

where

$$L^{\sqcup\!\sqcup,0} = \{\varepsilon\}$$

and

$$L^{\sqcup\!\sqcup,i} = L^{\sqcup\!\sqcup,i-1} \sqcup\!\sqcup\, L,$$

where $i \geq 1$.

Definition

All permutations of w, denoted by $\mathrm{perm}(w)$, is defined as

$$\mathrm{perm}(\varepsilon) = \{\varepsilon\}$$
$$\mathrm{perm}(au) = \{a\} \sqcup\!\sqcup \mathrm{perm}(u)$$

where $a \in \Sigma$ and $u \in \Sigma^*$.

For $L \subseteq \Sigma^*$, $\mathrm{perm}(L) = \bigcup_{w \in L} \mathrm{perm}(w)$.

Definition

All permutations of w, denoted by $\text{perm}(w)$, is defined as

$$\text{perm}(\varepsilon) = \{\varepsilon\}$$
$$\text{perm}(au) = \{a\} \sqcup\!\sqcup \text{perm}(u)$$

where $a \in \Sigma$ and $u \in \Sigma^*$.

For $L \subseteq \Sigma^*$, $\text{perm}(L) = \bigcup_{w \in L} \text{perm}(w)$.

Example

$$\text{perm}(abc) = \{abc, acb, cba, bac, bca, cab\}$$

Definition

All permutations of w, denoted by $\mathrm{perm}(w)$, is defined as

$$\mathrm{perm}(\varepsilon) = \{\varepsilon\}$$
$$\mathrm{perm}(au) = \{a\} \sqcup\!\sqcup \mathrm{perm}(u)$$

where $a \in \Sigma$ and $u \in \Sigma^*$.

For $L \subseteq \Sigma^*$, $\mathrm{perm}(L) = \bigcup_{w \in L} \mathrm{perm}(w)$.

Example

$$\mathrm{perm}(abc) = \{abc, acb, cba, bac, bca, cab\}$$

Proposition

For $u, v \in \Sigma^*$, $\mathrm{perm}(u) = \mathrm{perm}(v)$ if and only if $\psi_\Sigma(u) = \psi_\Sigma(v)$.

Definitions – SHUF Expressions

Definition (Jantzen (1979))

Let Σ be an alphabet. The (atomic) SHUF expressions are

- $\emptyset$
- ε
- $w \in \Sigma^+$

If r, s are SHUF expressions, then

- $(r + s)$
- $(r \sqcup\!\sqcup s)$
- $r^{\sqcup\!\sqcup,*}$

are SHUF expressions. They denote the corresponding languages as expected.

Definition (Fernau et al. (2016))

A SHUF expression is an α-SHUF expression, if its atoms are only $\emptyset$, ε, or single symbols $a \in \Sigma$.

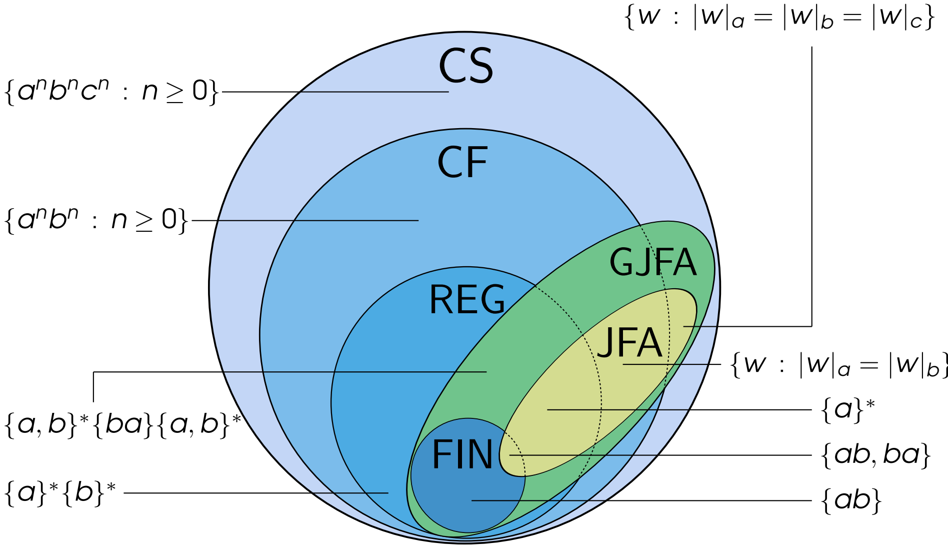
Definitions – SHUF Expressions

FIT

Example

The language from Example #1 can be denoted by the following α-SHUF expression

$$(a \sqcup\!\sqcup\, b \sqcup\!\sqcup\, c)^{\sqcup\!\sqcup,*}$$


$\{w : |w|_a = |w|_b = |w|_c\}$
CS
$\{a^n b^n c^n : n \geq 0\}$
CF
$\{a^n b^n : n \geq 0\}$
GJFA
REG
JFA
$\{w : |w|_a = |w|_b\}$
$\{a\}^*$
$\{a, b\}^*\{ba\}\{a, b\}^*$
FIN
$\{ab, ba\}$
$\{a\}^*\{b\}^*$
$\{ab\}$

Power of JFAs

Theorem (Meduna, Zemek (2012) & Fernau et al. (2016))

$\mathscr{L}(JFA) = \mathrm{perm}(\boldsymbol{REG}) = \mathrm{perm}(\boldsymbol{CF}) = \mathrm{perm}(\boldsymbol{PSL})$

Theorem (Meduna, Zemek (2012) & Fernau et al. (2016))

$\mathscr{L}(JFA) = \text{perm}(\boldsymbol{REG}) = \text{perm}(\boldsymbol{CF}) = \text{perm}(\boldsymbol{PSL})$

Corollary (Fernau et al. (2016))

$\mathscr{L}(JFA)$ is closed under intersection and under complementation.

Theorem (Meduna, Zemek (2012) & Fernau et al. (2016))

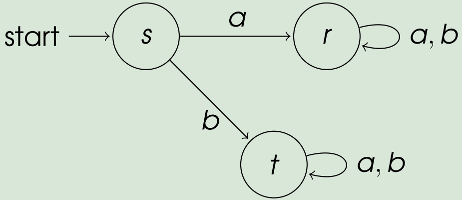
$\mathscr{L}(JFA) = \text{perm}(\boldsymbol{REG}) = \text{perm}(\boldsymbol{CF}) = \text{perm}(\boldsymbol{PSL})$

Corollary (Fernau et al. (2016))

$\mathscr{L}(JFA)$ is closed under intersection and under complementation.

Example

Standard complementation technique does not work for JFAs.



- For $F = \{r\}$, it accepts all words that contains at least one a.
- If $F = \{s, t\}$, it accepts all words that contain at least one b.

Theorem (Fernau et al. (2016))

$\mathscr{L}(\alpha\text{-SHUF}) = \mathscr{L}(JFA)$.

Proof Idea

- $\supseteq$: If $L \in \mathscr{L}(JFA)$, there exists regular L' such that $L = \text{perm}(L')$. Then, RE R' denotes L'. Then, we find an α-SHUF expression R with $L = \text{perm}(L(R')) = L(R)$.
- $\subseteq$: Let α-SHUF expression R describes L. Construct RE R' by replacing all $\sqcup\!\sqcup$ by $\cdot$ and ${}^{\sqcup\!\sqcup,*}$ by *, so $L(R) = \text{perm}(L(R'))$. As $\text{perm}(L(R')) \in \mathbf{REG}$, $L \in \mathscr{L}(JFA)$. $\square$

Theorem (Fernau et al. (2016))

$\mathscr{L}(\alpha\text{-SHUF}) = \mathscr{L}(JFA)$.

Proof Idea

- $\supseteq$: If $L \in \mathscr{L}(JFA)$, there exists regular L' such that $L = \text{perm}(L')$. Then, RE R' denotes L'. Then, we find an α-SHUF expression R with $L = \text{perm}(L(R')) = L(R)$.
- $\subseteq$: Let α-SHUF expression R describes L. Construct RE R' by replacing all $\sqcup\!\sqcup$ by $\cdot$ and ${}^{\sqcup\!\sqcup,*}$ by *, so $L(R) = \text{perm}(L(R'))$. As $\text{perm}(L(R')) \in \textbf{REG}$, $L \in \mathscr{L}(JFA)$. □

Corollary

$\mathscr{L}(JFA)$ is closed under iterated shuffle.

Theorem (Fernau et al. (2016))

$\mathscr{L}$(GJFA) and $\mathscr{L}$(SHUF) are incomparable.

Proof Idea

- Let $M = (\{s\}, \Sigma, \{sab \rightarrow s, scd \rightarrow s\}, s, \{s\})$. $L(M) \notin \mathscr{L}(\text{SHUF})$.
- $L(ac \sqcup\!\sqcup (bd)^{\sqcup\!\sqcup,*})$ is not accepted by any *GJFA*. $\square$

Theorem (Fernau et al. (2016))

$\mathscr{L}$(GJFA) and $\mathscr{L}$(SHUF) are incomparable.

Proof Idea

- Let $M = (\{s\}, \Sigma, \{sab \rightarrow s, scd \rightarrow s\}, s, \{s\})$. $L(M) \notin \mathscr{L}(\text{SHUF})$.
- $L(ac \sqcup\!\sqcup (bd)^{\sqcup\!\sqcup,*})$ is not accepted by any *GJFA*. □

Lemma (Fernau et al. (2016))

$\{ab\}^{\sqcup\!\sqcup,*} \in (\mathscr{L}(\mathit{GJFA}) \cap \mathscr{L}(\text{SHUF})) - \mathscr{L}(\mathit{JFA})$.

Theorem (Fernau et al. (2016))

$\mathscr{L}(GJFA)$ and $\mathscr{L}(\text{SHUF})$ are incomparable.

Proof Idea

- Let $M = (\{s\}, \Sigma, \{sab \to s, scd \to s\}, s, \{s\})$. $L(M) \notin \mathscr{L}(\text{SHUF})$.
- $L(ac \sqcup\!\sqcup (bd)^{\sqcup\!\sqcup,*})$ is not accepted by any *GJFA*. □

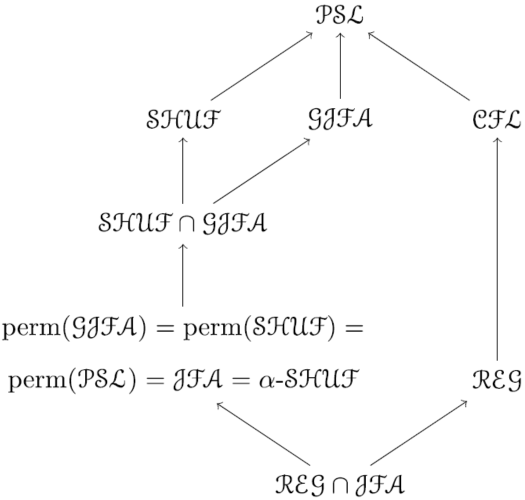
Lemma (Fernau et al. (2016))

$\{ab\}^{\sqcup\!\sqcup,*} \in (\mathscr{L}(GJFA) \cap \mathscr{L}(\text{SHUF})) - \mathscr{L}(JFA)$.

Theorem (Fernau et al. (2016))

$\mathscr{L}(JFA) = \text{perm}(\boldsymbol{REG}) = \text{perm}(\boldsymbol{CF}) = \text{perm}(\boldsymbol{PSL})$
$= \text{perm}(\mathscr{L}(GJFA)) = \text{perm}(\mathscr{L}(\text{SHUF}))$

Relations Between Language Families II (Fernau et al., 2016)



Theorem (Vorel (2015), Theorem 2)

$\mathscr{L}$(GJFA) is not closed under Kleene star, Kleene plus, ε-free and general homomorphism and finite substitution.

Proof

- We have $\{ab\} \in \mathscr{L}(GJFA)$, but $\{ab\}^* \notin \mathscr{L}(GJFA)$.
- Since $\mathscr{L}(GJFA)$ is closed under union, $\{ab\}^+ \notin \mathscr{L}(GJFA)$.
- Consider ε-free homomorphism $\varphi\colon \{a\}^* \to \{a, b\}^*$ with $\varphi(a) = ab$.
- For $L = \{a\}^* \in \mathscr{L}(GJFA)$, $\varphi(L) = \{ab\}^* \notin \mathscr{L}(GJFA)$.
- In addition, φ is a general homomorphism and finite substitution as well. $\square$

Closure Properties – Summary

	$\mathscr{L}(GJFA)$	$\mathscr{L}(JFA)$
union	$+$	$+$
intersection	$-^{*}$(Vorel, 2015)	$+$
concatenation	$-$	$-$
intersection with reg. lang.	$-$	$-$
complement	$-$	$+^{*}$(Fernau et al., 2016)
shuffle	$-$ (Vorel, 2015)	$+$
iterated shuffle	?	$+$ (Fernau et al., 2016)
mirror image	$+$ (Vorel, 2015)	$+$
Kleene star	$-$ (Vorel, 2015)	$-$
Kleene plus	$-$ (Vorel, 2015)	$-$
substitution	$-$	$-$
regular substitution	$-$	$-$
finite substitution	$-$ (Vorel, 2015)*	$-$
homomorphism	$-$ (Vorel, 2015)*	$-$
ε-free homomorphism	$-$ (Vorel, 2015)*	$-$
inverse homomorphism	$-$ (Vorel, 2015)*	$+$

Note: * marks corrections. (Meduna, Zemek, 2012) when the source is not specified.

Decidability – Summary by Meduna, Zemek (2012)

	$\mathscr{L}(GJFA)$	$\mathscr{L}(JFA)$
membership	$+$	$+$
emptiness	$+$	$+$
finiteness	$+$	$+$
infiniteness	$+$	$+$

Theorem (Vorel (2016), Thm. 1)

Given a GJFA $M = (Q, \Sigma, R, s, F)$*, it is undecidable whether* $L(M) = \Sigma^*$.

Proof Idea

By reduction from universality of context-free grammar to the universality of GJFA.

Theorem (Vorel (2015), Thm. 6)

Given GJFA M_1 *and* M_2 *over an 8-letter alphabet, it is undecidable whether* $L(M_1) \cap L(M_2) = \emptyset$.

Proof Idea

Using a prefix-disjoint instance of the Post correspondence problem over a range alphabet.

Decidability – Extended Summary

FIT

	$\mathscr{L}(GJFA)$	$\mathscr{L}(JFA)$
membership	+	+
emptiness	+	+
finiteness	+	+
infiniteness	+	+
universality	− (Vorel, 2016)	+ (Fernau et al., 2016)
disjointness	− (Vorel, 2016)[1]	+ (Fernau et al., 2016)

[1] GJFAs are over an 8-letter alphabet.

Note on Parsing of Fixed JFA

Scan over w and store the current state and the Parikh mapping (as Σ fixed, use working tape of non-det. logspace machine). Thus, $\mathscr{L}(JFA) \subseteq \mathrm{NL} \subseteq \mathrm{P}$.

Note on Parsing of Fixed JFA

Scan over w and store the current state and the Parikh mapping (as Σ fixed, use working tape of non-det. logspace machine). Thus, $\mathscr{L}(JFA) \subseteq \mathrm{NL} \subseteq \mathrm{P}$.

Theorem (Fernau et al. (2016))

Unless ETH fails, there is no algorithm that, for given JFA M with state set Q and a given word w, decides whether $w \in L(M)$ and runs in time $O^*(2^{o(|Q|)})$.

Note on ETH

Often, Exponential Time Hypothesis (ETH) is used to state computational complexity results.
If ETH holds, then $P \neq NP$.

Problem	*GJFA*	*GJFA* $\|\Sigma\| = k$	*JFA*	*JFA* $\|\Sigma\| = k$
Fixed word	NP-C	NP-C*	P	P
Universal word	NP-C	NP-C*	NP-C	P
Non-disjointness	Und.	Und.	NP-C	P
Non-universality	Und.	NP-H	NP-H	NP-C

Note: * marks results from (Fernau et al., 2016). NP-C = NP-complete; NP-H = NP-hard, membership in NP unknown; Und. = undecidable.

Open Problem Areas

- closure property of $\mathscr{L}(GJFA)$ (iterated shuffle?)
- other decision problems of $\mathscr{L}(GJFA)$ and $\mathscr{L}(JFA)$, like equivalence and inclusion
- variants of JFA and GJFA (determinism, parallel, regulated, ...)

Thank you for your attention!

Part Two follows!

Additional References

M. Jantzen: Eigenschaften von Petrinetzsprachen. Technical report IFI-HH-B-64