Jumping Finite Automata

Alexander Meduna and Petr Zemek

Brno University of Technology, Faculty of Information Technology
IT4Innovations Centre of Excellence
Božetěchova 2, 612 00 Brno, Czech Republic
http://www.fit.vutbr.cz/∼{meduna, izemek}

Based on

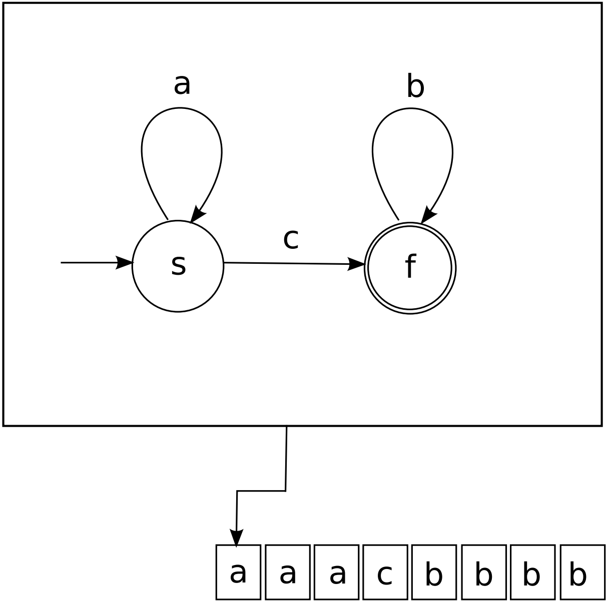
Alexander Meduna and Petr Zemek.
Jumping Finite Automata.
International Journal of Foundations of Computer Science
(to appear in 2013).

Supported by the IT4I Centre of Excellence CZ.1.05/1.1.00/02.0070.

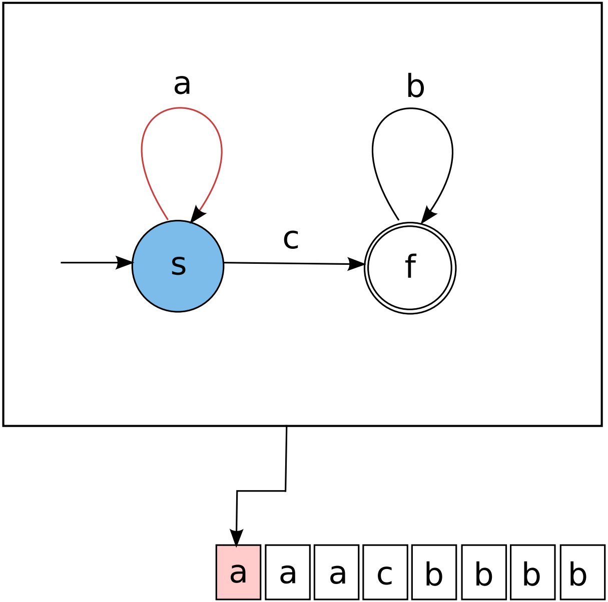
Contents

- **Introduction**
- **Definitions and Examples**
- **Results**
- **Concluding Remarks and Discussion**

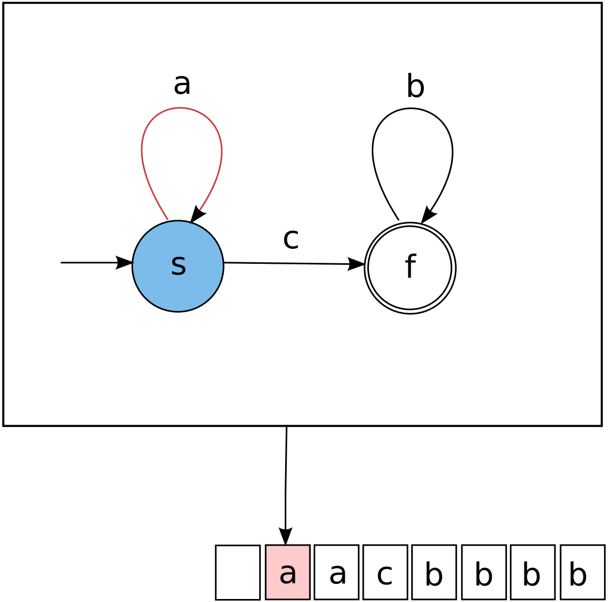
Finite Automata

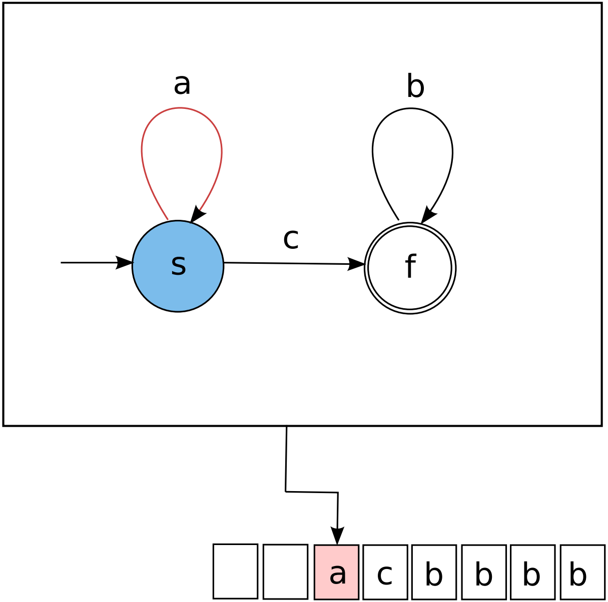


Finite Automata

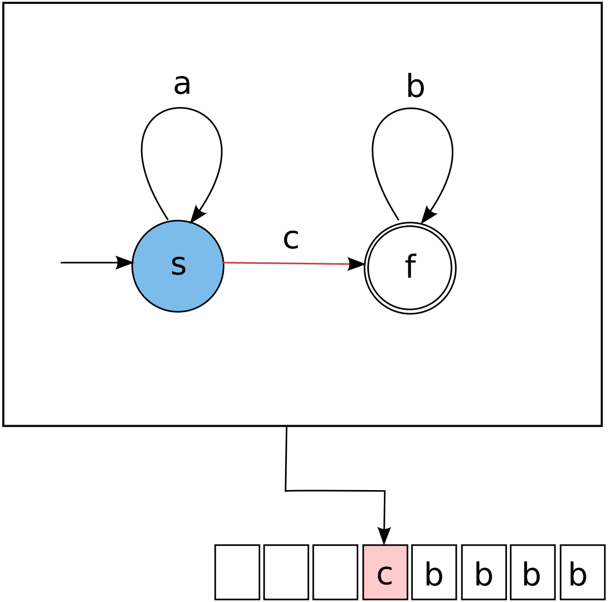


Finite Automata

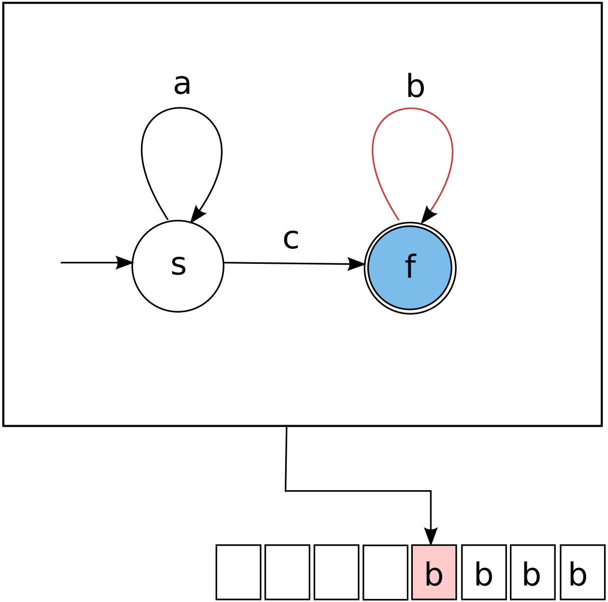


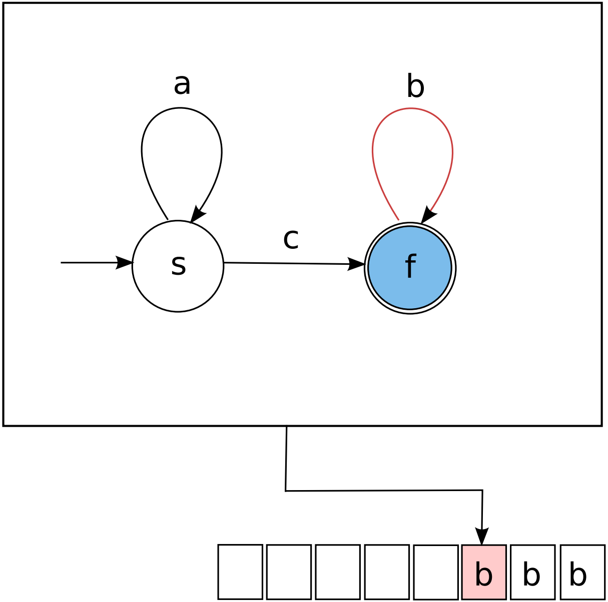

a
b
c
s
f
a
c
b
b
b
b

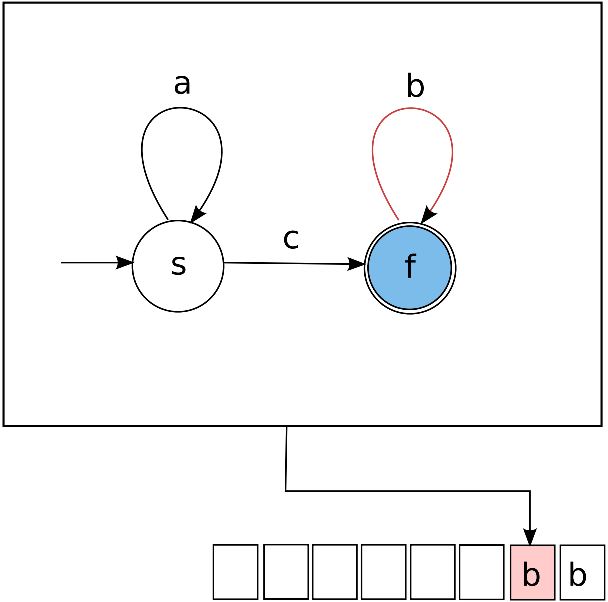
Finite Automata



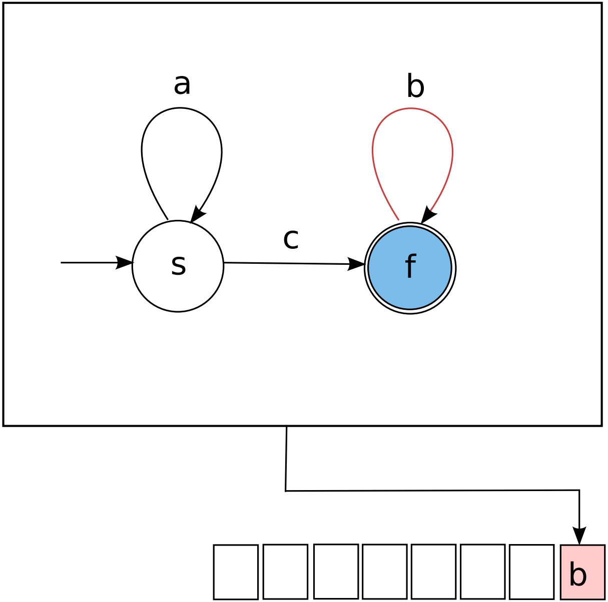
Finite Automata

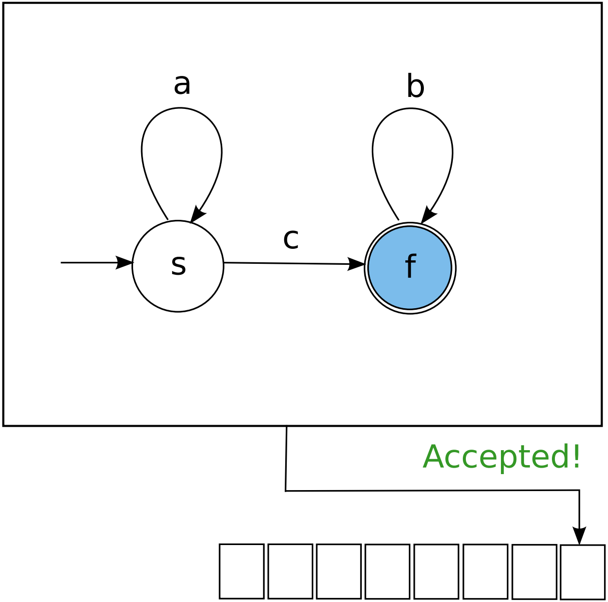



a
b
c
s
f
b
b
b

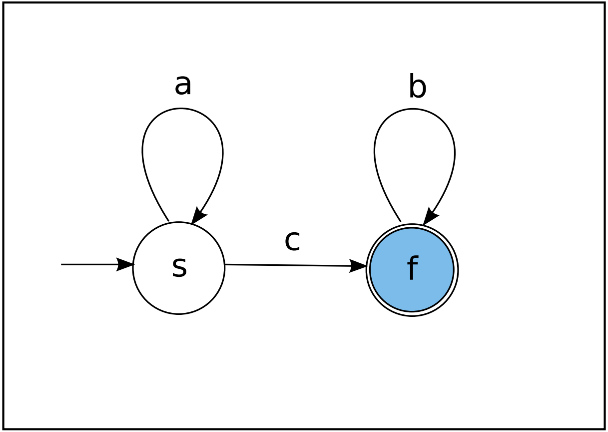

a
b
c
s
f
b
b

Finite Automata

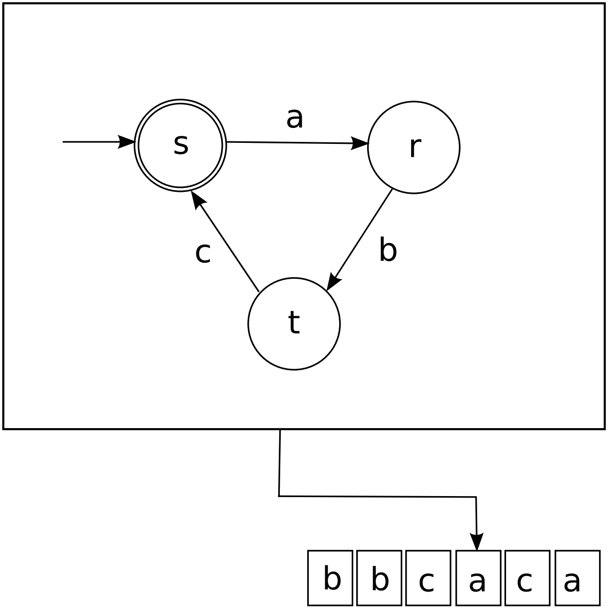



a
b
c
s
f
Accepted!

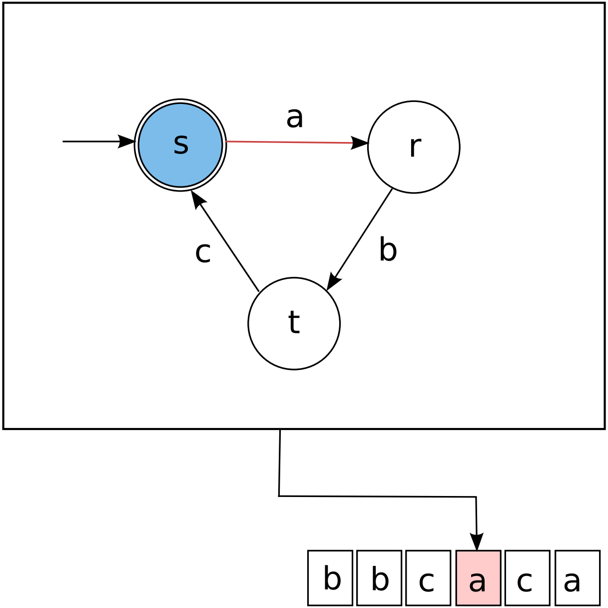
Finite Automata

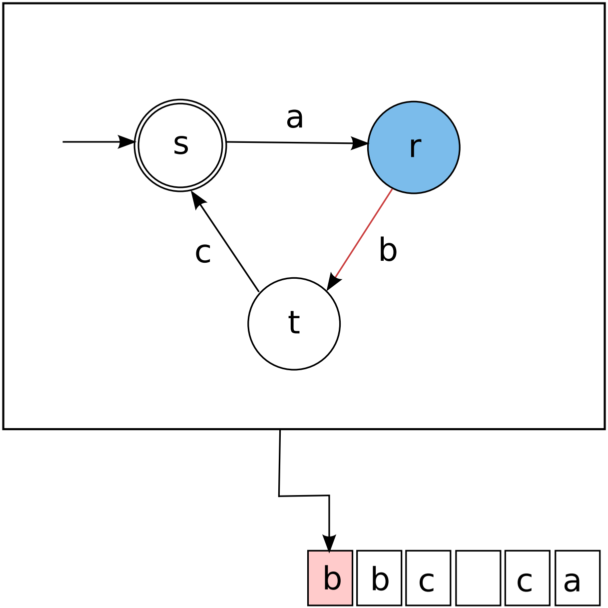


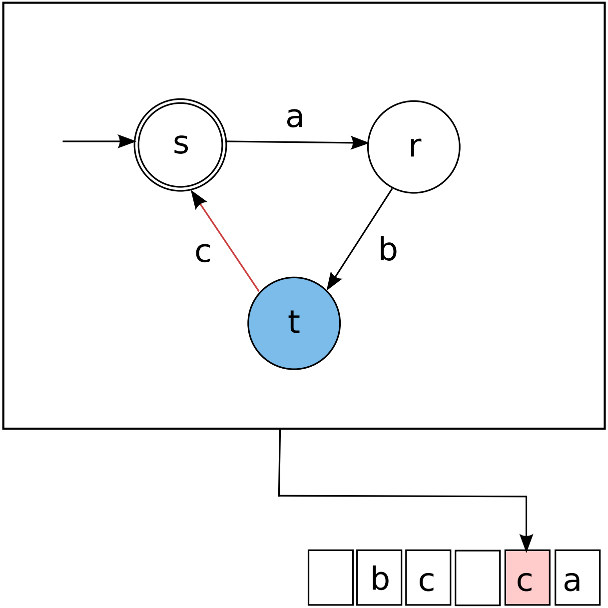
Accepted language: $\{a\}^*\{c\}\{b\}^*$


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

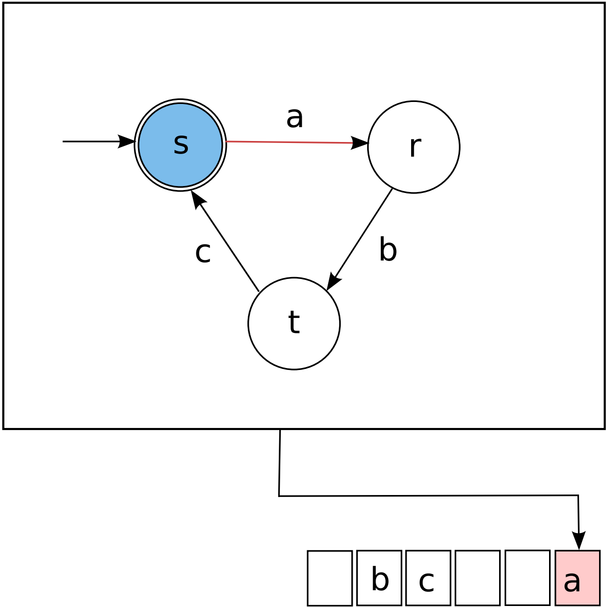
Jumping Finite Automata



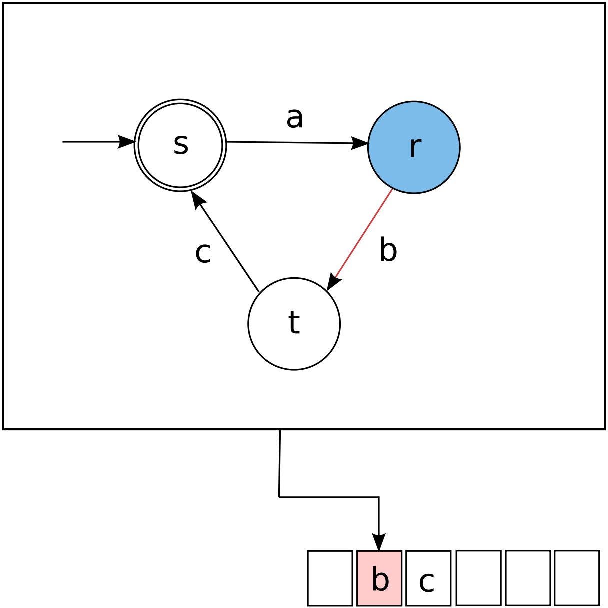

a
s
r
c
b
t
b
b
c
c
a


s
a
r
c
b
t
b
c
c
a

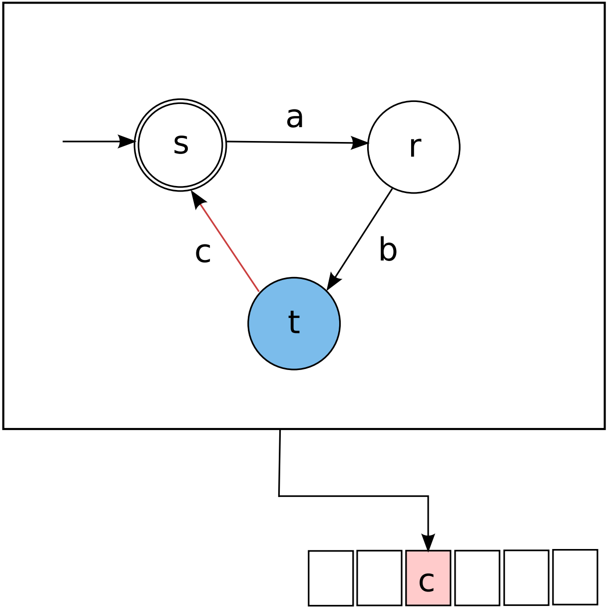
Jumping Finite Automata



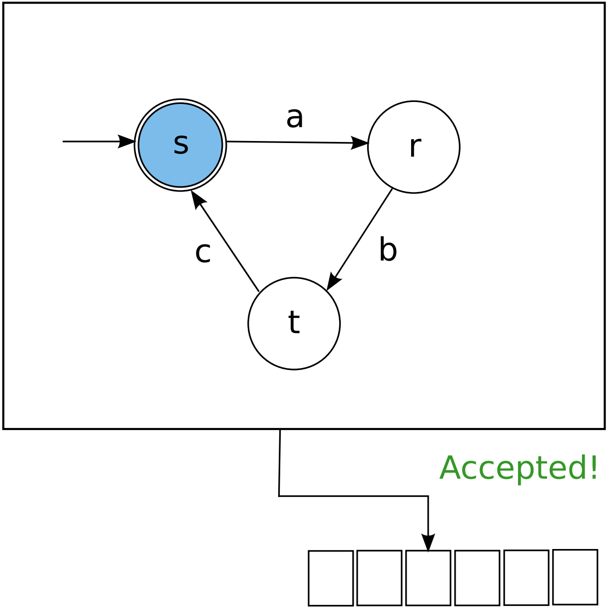
Jumping Finite Automata

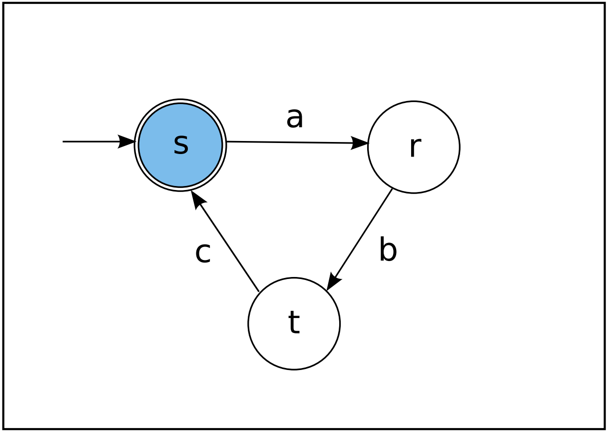


Jumping Finite Automata



Jumping Finite Automata





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

Definition

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

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

If all rules $py \rightarrow q \in R$ 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

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

- $\curvearrowright^n$ intuitively, a sequence of n jumps ($n \geq 0$); mathematically, the nth power of $\curvearrowright$
- $\curvearrowright^*$ 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 $py \to q \in R$, then M makes a *jump* from $xpyz$ to $x'qz'$, symbolically written as

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

$\curvearrowright^n$ intuitively, a sequence of n jumps ($n \geq 0$); mathematically, the nth power of $\curvearrowright$

$\curvearrowright^*$ 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

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

Example

The JFA

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

with

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

accepts

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

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:

$$bacbc\underline{s}a$$

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:

$$bacbc\underline{s}a \curvearrowright bac\underline{r}bc \quad [sa \to r]$$

Example #1

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:

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

Example #1

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:

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

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:

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

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:

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

Example

The JFA

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

with

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

accepts

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

For instance:

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

Example

The GJFA

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

with

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

accepts

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

Example

The GJFA

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

with

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

accepts

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

For instance:

$$bb\underline{s}baa$$

Example #2

Example

The GJFA

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

with

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

accepts

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

For instance:

$$bb\underline{s}baa \curvearrowright bb\underline{f}a \quad [sba \to f]$$

Example #2

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:

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

Example #2

Example

The GJFA

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

with

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

accepts

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

For instance:

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

Example

The GJFA

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

with

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

accepts

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

For instance:

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

Power of GJFAs and JFAs

Theorem

Let K be an arbitrary language. Then, K is accepted by a JFA only if $K = \mathrm{perm}(K)$.

Theorem

Let K be an arbitrary language. Then, K is accepted by a JFA only if $K = \mathrm{perm}(K)$.

Proof Idea

When using $pa \rightarrow q$, a can appear on any position. □

Theorem

Let K be an arbitrary language. Then, K is accepted by a JFA only if $K = \text{perm}(K)$.

Proof Idea

When using $pa \rightarrow q$, a can appear on any position. □

Corollary

There is no JFA that accepts $\{a, b\}^*\{ba\}\{a, b\}^*$.

Theorem

Let K be an arbitrary language. Then, K is accepted by a JFA only if $K = \text{perm}(K)$.

Proof Idea

When using $pa \rightarrow q$, a can appear on any position. □

Corollary

There is no JFA that accepts $\{a, b\}^*\{ba\}\{a, b\}^*$.

Theorem

GJFAs are strictly stronger than JFAs.

Theorem

Let K be an arbitrary language. Then, K is accepted by a JFA only if $K = \mathrm{perm}(K)$.

Proof Idea

When using $pa \rightarrow q$, a can appear on any position. □

Corollary

There is no JFA that accepts $\{a, b\}^*\{ba\}\{a, b\}^*$.

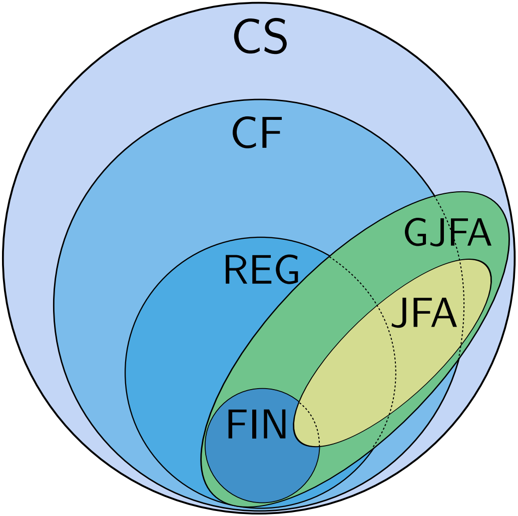
Theorem

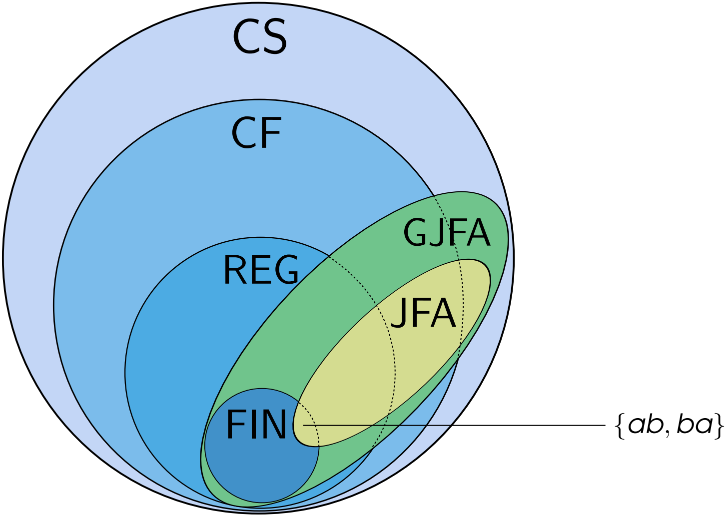
GJFAs are strictly stronger than JFAs.

Proof Idea

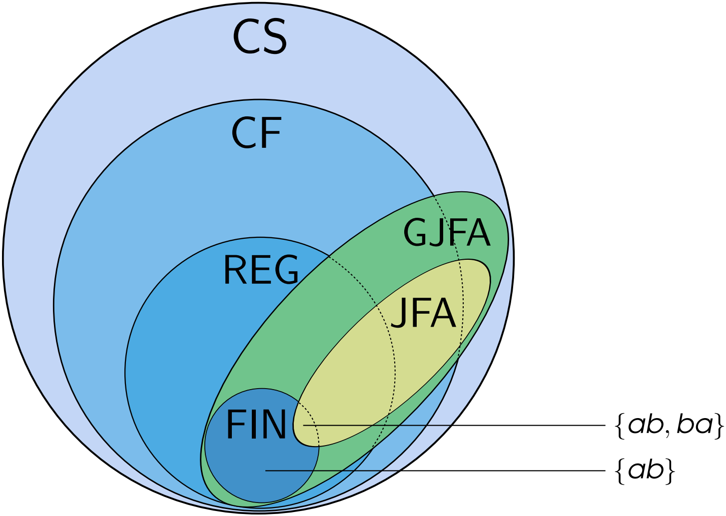
The language $\{a, b\}^*\{ba\}\{a, b\}^*$ is accepted by the GJFA from Example #2. □

Relations Between Language Families

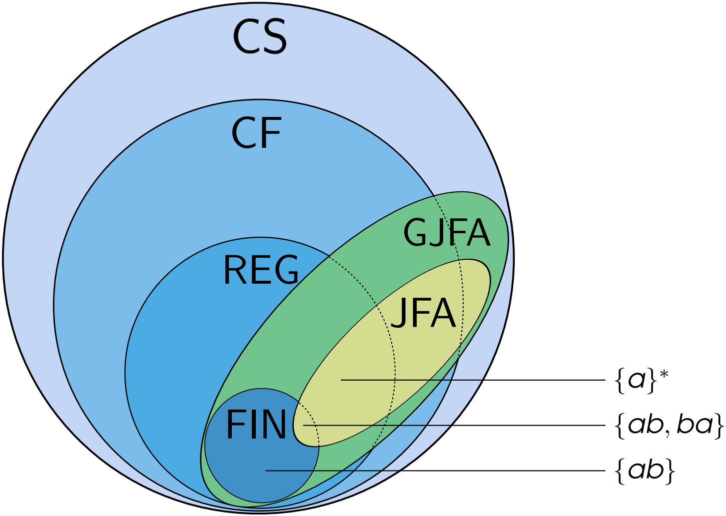


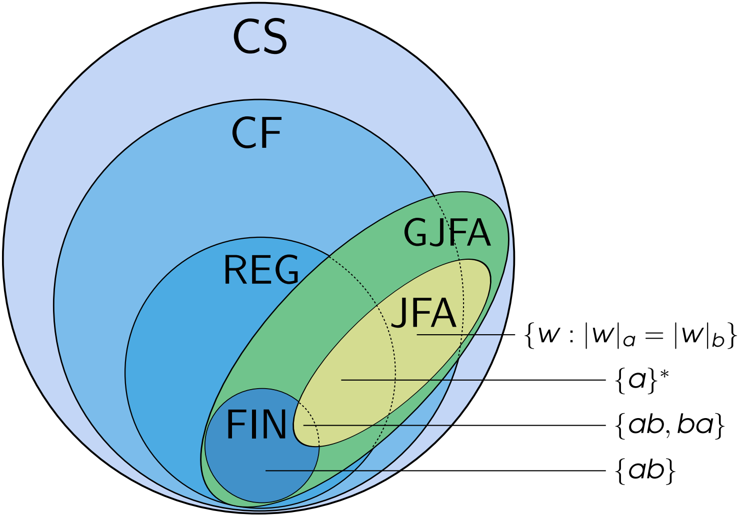

CS
CF
REG
GJFA
JFA
FIN
{ab, ba}

Relations Between Language Families

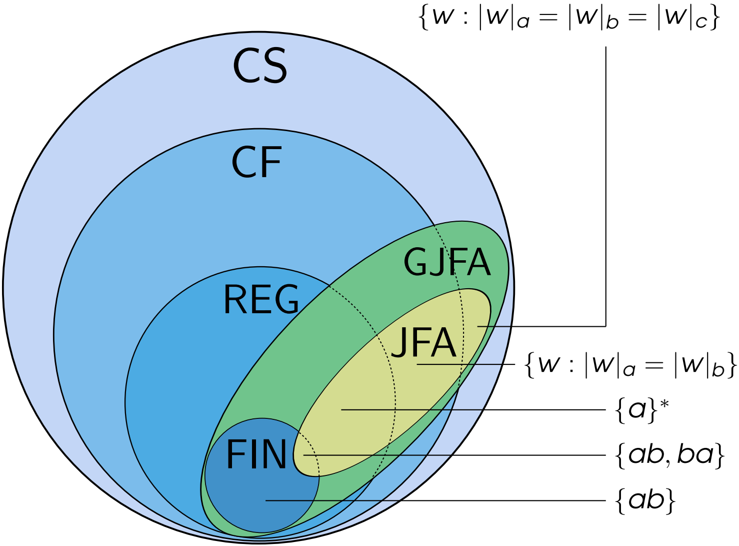


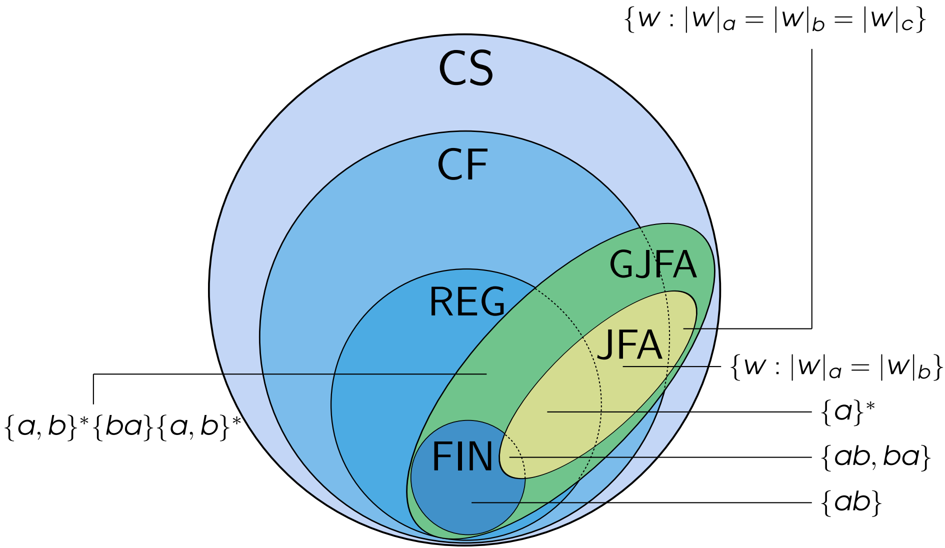
Relations Between Language Families



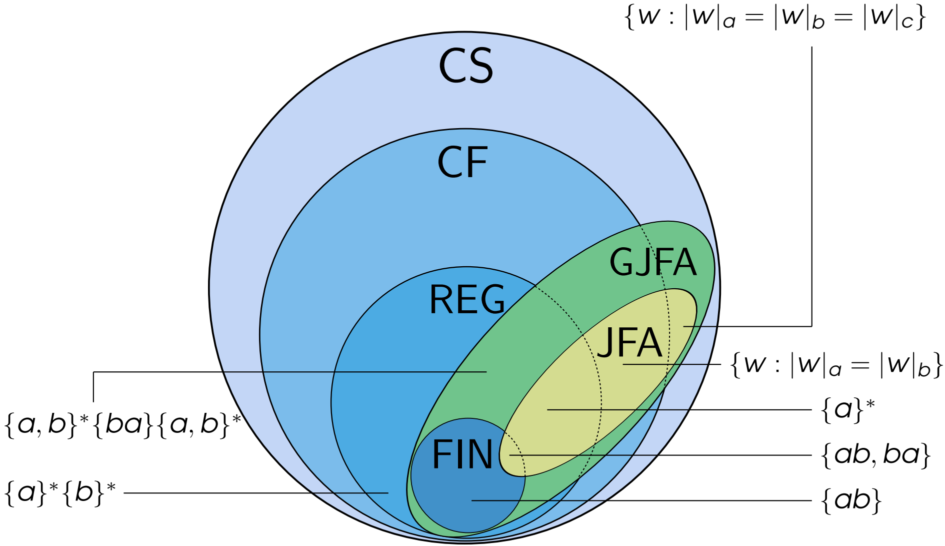

CS
CF
GJFA
REG
JFA
$\{w : |w|_a = |w|_b\}$
$\{a\}^*$
FIN
$\{ab, ba\}$
$\{ab\}$

Relations Between Language Families

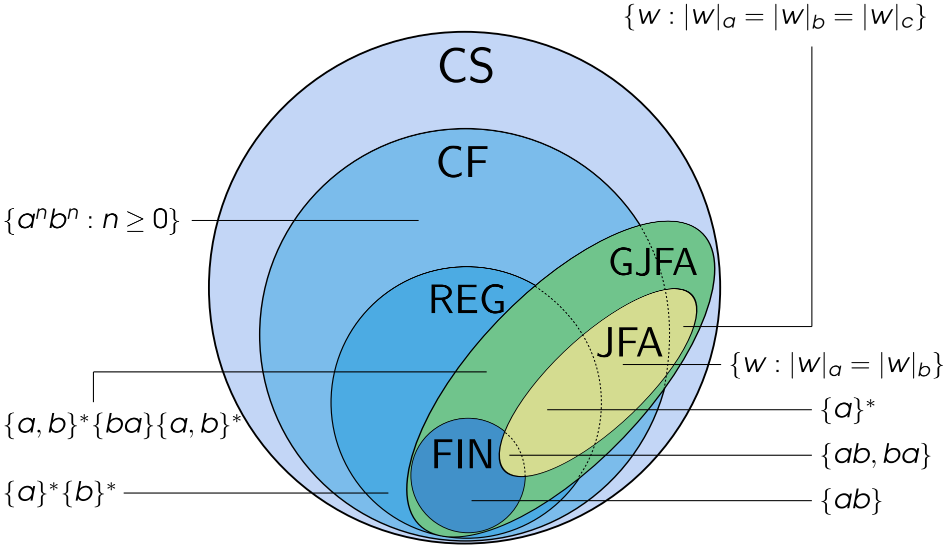


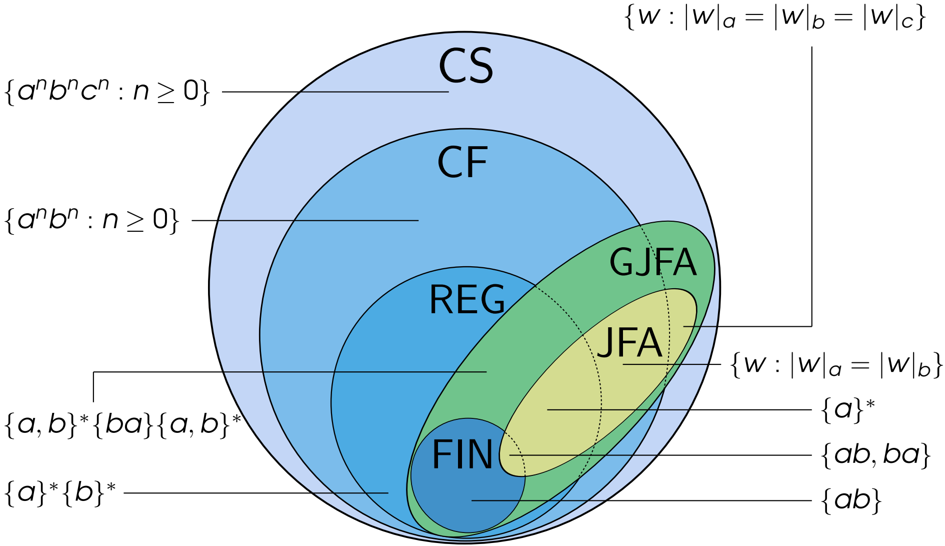

$\{w : |w|_a = |w|_b = |w|_c\}$
CS
CF
GJFA
REG
JFA
$\{w : |w|_a = |w|_b\}$
$\{a, b\}^*\{ba\}\{a, b\}^*$
$\{a\}^*$
FIN
$\{ab, ba\}$
$\{ab\}$

Relations Between Language Families



Relations Between Language Families




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

By analogy with finite automata:

- removal of ε-*moves* $\quad (p \to q$ and $qa \to r \quad \Rightarrow \quad pa \to r)$
- making JFAs deterministic

By analogy with finite automata:

- removal of ε-*moves* $(p \to q$ and $qa \to r \;\Rightarrow\; pa \to r)$
- making JFAs deterministic

Theorem

Every unary language accepted by a JFA is regular.

By analogy with finite automata:

- removal of ε-*moves* $(p \to q$ and $qa \to r \quad \Rightarrow \quad pa \to r)$
- making JFAs deterministic

Theorem

Every unary language accepted by a JFA is regular.

Proof Idea

In unary languages, it does not matter where the automaton jumps. □

By analogy with finite automata:

- removal of ε-*moves* $(p \to q$ and $qa \to r \quad \Rightarrow \quad pa \to r)$
- making JFAs deterministic

Theorem

Every unary language accepted by a JFA is regular.

Proof Idea

In unary languages, it does not matter where the automaton jumps. □

Corollary

The language of primes

$$\{a^p : p \text{ is a prime number}\}$$

cannot be accepted by any JFA.

Closure Properties

Theorem

JFA *is closed under union.*

Theorem

JFA *is closed under union.*

Proof

We have: Two JFAs

- $M_1 = (Q_1, \Sigma_1, R_1, s_1, F_1)$
- $M_2 = (Q_2, \Sigma_2, R_2, s_2, F_2)$ $\quad (Q_1 \cap Q_2 = \emptyset)$

We need: JFA $H = (Q, \Sigma, R, s, F)$ such that $L(H) = L(M_1) \cup L(M_2)$

Construction:

$$\begin{aligned} Q &= Q_1 \cup Q_2 \cup \{s\} && (s \notin Q_1 \cup Q_2) \\ \Sigma &= \Sigma_1 \cup \Sigma_2 \\ R &= R_1 \cup R_2 \cup \{s \to s_1, s \to s_2\} \\ F &= F_1 \cup F_2 \end{aligned}$$

□

Closure Properties – Continued

Theorem

JFA *is not closed under concatenation.*

Theorem

JFA *is not closed under concatenation.*

Proof

- Consider $K_1 = \{a\}$ and $K_2 = \{b\}$.
- The JFA $M_1 = (\{s, f\}, \{a\}, \{sa \rightarrow f\}, s, \{f\})$ accepts K_1.
- The JFA $M_2 = (\{s, f\}, \{b\}, \{sb \rightarrow f\}, s, \{f\})$ accepts K_2.
- However, there is no JFA that accepts $K_1K_2 = \{ab\}$. $\square$

Closure Properties – Summary

	GJFA	JFA
union	+	+
intersection	−	+
concatenation	−	−
intersection with reg. lang.	−	−
complement	−	+
shuffle	?	+
mirror image	?	+
Kleene star	?	−
Kleene plus	?	−
substitution	−	−
regular substitution	−	−
finite substitution	+	−
homomorphism	+	−
ε-free homomorphism	+	−
inverse homomorphism	+	+

Decidability – Summary

	GJFA	JFA
membership	+	+
emptiness	+	+
finiteness	+	+
infiniteness	+	+

Definition

A GJFA $M = (Q, \Sigma, R, s, F)$ is of *degree* n, where $n \geq 0$, if $py \rightarrow q \in R$ implies that $|y| \leq n$.

Definition

A GJFA $M = (Q, \Sigma, R, s, F)$ is of *degree n*, where $n \geq 0$, if $py \rightarrow q \in R$ implies that $|y| \leq n$.

Example

The GJFA $M = (\{s, p, f\}, \{a, b, c\}, R, s, \{f\})$ with

$$R = \{sabc \rightarrow p, pcc \rightarrow f, fa \rightarrow f\}$$

is of degree 3.

Definition

A GJFA $M = (Q, \Sigma, R, s, F)$ is of *degree n*, where $n \geq 0$, if $py \rightarrow q \in R$ implies that $|y| \leq n$.

Example

The GJFA $M = (\{s, p, f\}, \{a, b, c\}, R, s, \{f\})$ with

$$R = \{sabc \rightarrow p, pcc \rightarrow f, fa \rightarrow f\}$$

is of degree 3.

GJFA$_n$ the family of languages accepted by GJFAs of degree n

Definition

A GJFA $M = (Q, \Sigma, R, s, F)$ is of *degree n*, where $n \geq 0$, if $py \rightarrow q \in R$ implies that $|y| \leq n$.

Example

The GJFA $M = (\{s, p, f\}, \{a, b, c\}, R, s, \{f\})$ with

$$R = \{sabc \rightarrow p, pcc \rightarrow f, fa \rightarrow f\}$$

is of degree 3.

GJFA$_n$ the family of languages accepted by GJFAs of degree n

Theorem

$\mathbf{GJFA}_n \subset \mathbf{GJFA}_{n+1}$ *for all* $n \geq 0$

Definition

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

$$wx\underline{p}yz \; {}_{l}\!\curvearrowright \; w\underline{q}xz$$

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

Definition

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

$$wx\underline{p}yz \ _l\!\curvearrowright \ w\underline{q}xz$$

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

Definition

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

$$w\underline{p}yxz \ _r\!\curvearrowright \ wx\underline{q}z$$

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

Definition

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

$$w x \underline{p} y z \; {}_l\!\curvearrowright \; w \underline{q} x z$$

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

Definition

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

$$w \underline{p} y x z \; {}_r\!\curvearrowright \; w x \underline{q} z$$

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

${}_l$**GJFA**	GJFAs using only left jumps
${}_l$**JFA**	JFAs using only left jumps
${}_r$**GJFA**	GJFAs using only right jumps
${}_r$**JFA**	JFAs using only right jumps

Theorem

$_r\mathbf{GJFA} = {}_r\mathbf{JFA} = \mathbf{REG}$

Theorem

${}_r\mathbf{GJFA} = {}_r\mathbf{JFA} = \mathbf{REG}$

Proof Idea

- ${}_r\mathbf{JFA} = \mathbf{REG}$ simulating a finite automaton
- ${}_r\mathbf{GJFA} = \mathbf{REG}$ simulating a *general finite automaton* □

Left and Right Jumps – Results

Theorem

$_r\mathbf{GJFA} = {}_r\mathbf{JFA} = \mathbf{REG}$

Proof Idea

- $_r\mathbf{JFA} = \mathbf{REG}$ simulating a finite automaton
- $_r\mathbf{GJFA} = \mathbf{REG}$ simulating a *general finite automaton* □

Theorem

$_l\mathbf{JFA} - \mathbf{REG} \neq \emptyset$

Theorem

$_r\mathbf{GJFA} = {_r\mathbf{JFA}} = \mathbf{REG}$

Proof Idea

- $_r\mathbf{JFA} = \mathbf{REG}$ simulating a finite automaton
- $_r\mathbf{GJFA} = \mathbf{REG}$ simulating a *general finite automaton* □

Theorem

$_l\mathbf{JFA} - \mathbf{REG} \neq \emptyset$

Proof Idea

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

with

$$R = \{sa \to p, pb \to s, sb \to q, qa \to s\}$$

accepts

$$_lL(M) = \{w : |w|_a = |w|_b\}$$ □

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a GJFA. Set

$$
\begin{aligned}
{}^{b}L(M) &= \{w \in \Sigma^* : \underline{s}w \curvearrowright^* \underline{f} \text{ with } f \in F\} && \text{(beginning)} \\
{}^{a}L(M) &= \{uv : u, v \in \Sigma^*, u\underline{s}v \curvearrowright^* \underline{f} \text{ with } f \in F\} && \text{(anywhere)} \\
{}^{e}L(M) &= \{w \in \Sigma^* : w\underline{s} \curvearrowright^* \underline{f} \text{ with } f \in F\} && \text{(end)}
\end{aligned}
$$

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a GJFA. Set

$$
\begin{array}{lcll}
{}^{b}L(M) & = & \{w \in \Sigma^* : \underline{s}w \curvearrowright^* \underline{f} \text{ with } f \in F\} & \text{(beginning)} \\
{}^{a}L(M) & = & \{uv : u, v \in \Sigma^*, u\underline{s}v \curvearrowright^* \underline{f} \text{ with } f \in F\} & \text{(anywhere)} \\
{}^{e}L(M) & = & \{w \in \Sigma^* : w\underline{s} \curvearrowright^* \underline{f} \text{ with } f \in F\} & \text{(end)}
\end{array}
$$

^{b}**GJFA**	GJFAs starting at the beginning
^{a}**GJFA**	GJFAs starting anywhere
^{e}**GJFA**	GJFAs starting at the end
^{b}**JFA**	JFAs starting at the beginning
^{a}**JFA**	JFAs starting anywhere
^{e}**JFA**	JFAs starting at the end

Definition

Let $M = (Q, \Sigma, R, s, F)$ be a GJFA. Set

$$
\begin{aligned}
{}^{b}L(M) &= \{w \in \Sigma^* : \underline{s}w \curvearrowright^* \underline{f} \text{ with } f \in F\} && \text{(beginning)}\\
{}^{a}L(M) &= \{uv : u, v \in \Sigma^*, u\underline{s}v \curvearrowright^* \underline{f} \text{ with } f \in F\} && \text{(anywhere)}\\
{}^{e}L(M) &= \{w \in \Sigma^* : w\underline{s} \curvearrowright^* \underline{f} \text{ with } f \in F\} && \text{(end)}
\end{aligned}
$$

^{b}**GJFA**	GJFAs starting at the beginning
^{a}**GJFA**	GJFAs starting anywhere
^{e}**GJFA**	GJFAs starting at the end
^{b}**JFA**	JFAs starting at the beginning
^{a}**JFA**	JFAs starting anywhere
^{e}**JFA**	JFAs starting at the end

Observations:

- ${}^{a}L(M) = L(M)$
- ^{a}**GJFA** $=$ **GJFA** and ^{a}**JFA** $=$ **JFA**

A Variety of Start Configurations – Results

Theorem

${}^{a}\mathbf{JFA} \subset {}^{b}\mathbf{JFA}$

Theorem

${}^{a}\mathbf{JFA} \subset {}^{b}\mathbf{JFA}$

Proof Idea

The JFA

$$M = (\{s, f\}, \{a, b\}, \{sa \rightarrow f, fb \rightarrow f\}, s, \{f\})$$

satisfies ${}^{b}L(M) = \{a\}\{b\}^{*}$ $(\{a\}\{b\}^{*} \notin {}^{a}\mathbf{JFA})$. □

Theorem

${}^{a}\mathbf{JFA} \subset {}^{b}\mathbf{JFA}$

Proof Idea

The JFA

$$M = (\{s, f\}, \{a, b\}, \{sa \rightarrow f, fb \rightarrow f\}, s, \{f\})$$

satisfies ${}^{b}L(M) = \{a\}\{b\}^*$ $(\{a\}\{b\}^* \notin {}^{a}\mathbf{JFA})$. □

Theorem

${}^{a}\mathbf{GJFA} \subset {}^{b}\mathbf{GJFA}$

Theorem

^{a}**JFA** $\subset$ ^{b}**JFA**

Proof Idea

The JFA

$$M = (\{s, f\}, \{a, b\}, \{sa \rightarrow f, fb \rightarrow f\}, s, \{f\})$$

satisfies $^{b}L(M) = \{a\}\{b\}^{*}$ ($\{a\}\{b\}^{*} \notin$ ^{a}**JFA**). □

Theorem

^{a}**GJFA** $\subset$ ^{b}**GJFA**

Theorem

^{e}**GJFA** $=$ ^{a}**GJFA** *and* ^{e}**JFA** $=$ ^{a}**JFA**

Conclusion and Open Problem Areas

- closure properties of **GJFA** (shuffle, Kleene star, Kleene plus, and mirror image)
- other decision problems of **GJFA** and **JFA**, like equivalence, universality, inclusion, or regularity
- the effect of left jumps to the power of JFAs and GJFAs (we only know that ${}_l\mathbf{JFA} - \mathbf{REG} \neq \emptyset$)
- strict determinism
- applications: verification of a relation concerning the number of symbol occurrences (genetics)

Discussion