

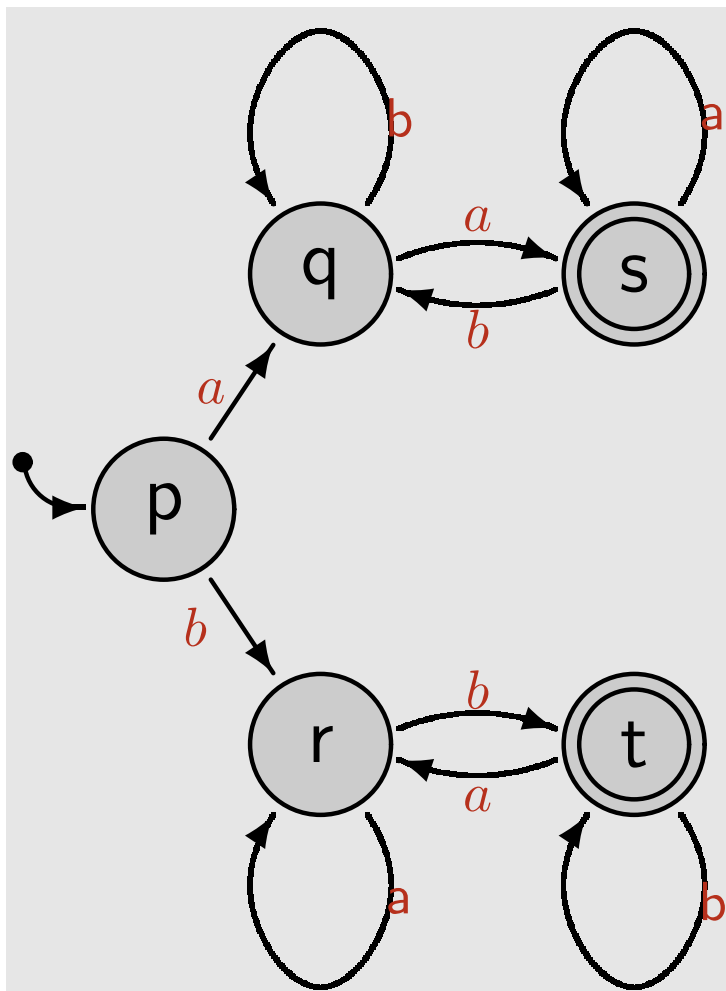
Stefan Sokołowski

JĘZYKI FORMALNE I METODY KOMPILACJI

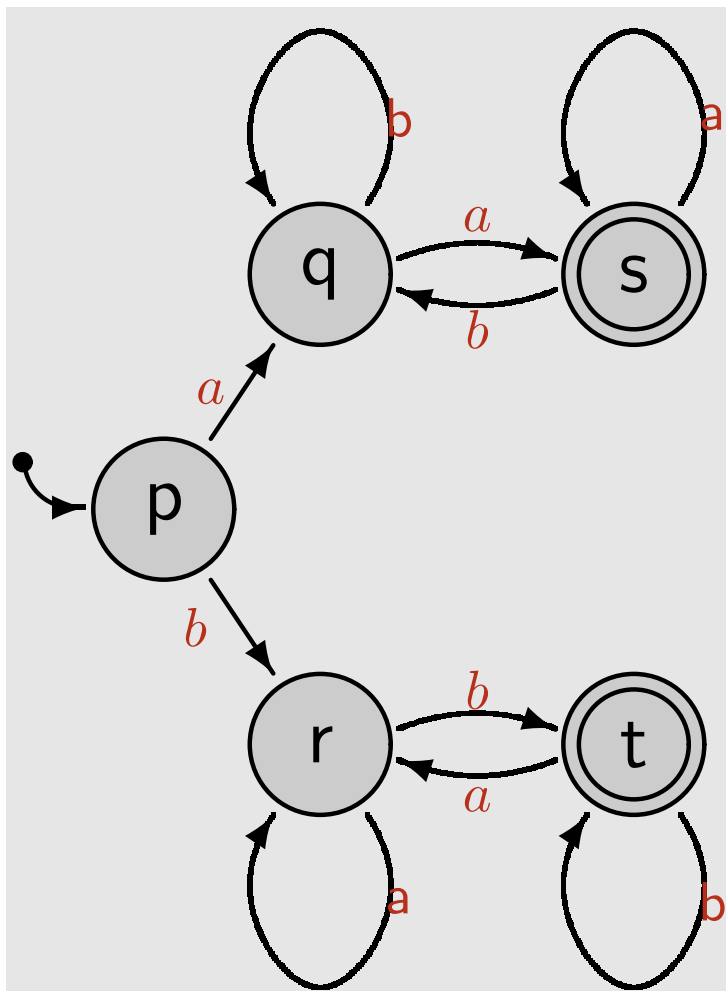
wykład 4

**Inst. Informatyki Stosowanej, ANS
Elbląg, 2025/2026**

Akceptory: automaty skończone



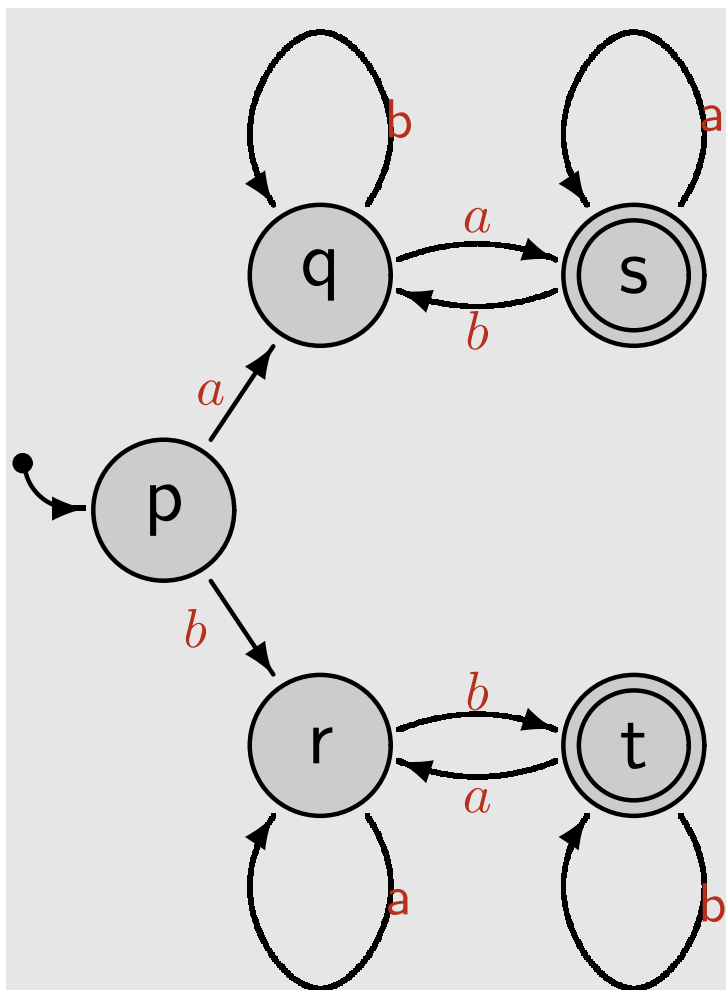
Akceptory: automaty skończone



Akceptacja słowa — od stanu początkowego do akceptującego.

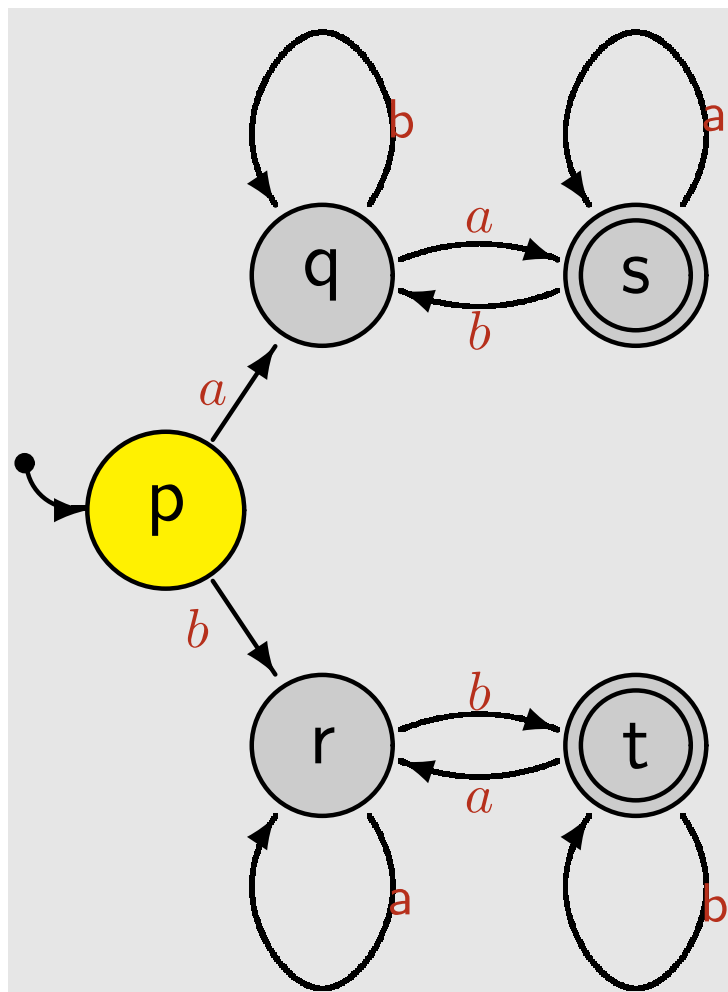
Akceptory: automaty skończone

Słowo *abab*



Akceptacja słowa — od stanu początkowego do akceptującego.

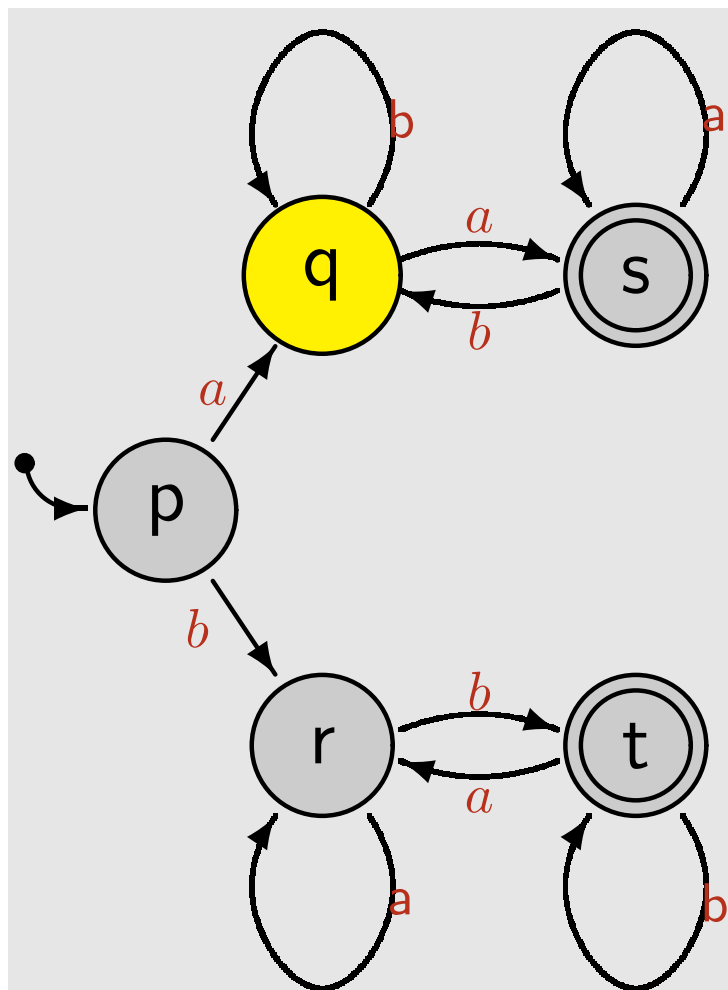
Akceptory: automaty skończone



Słowo *abab*:
 $\delta^*\langle p, abab \rangle$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone

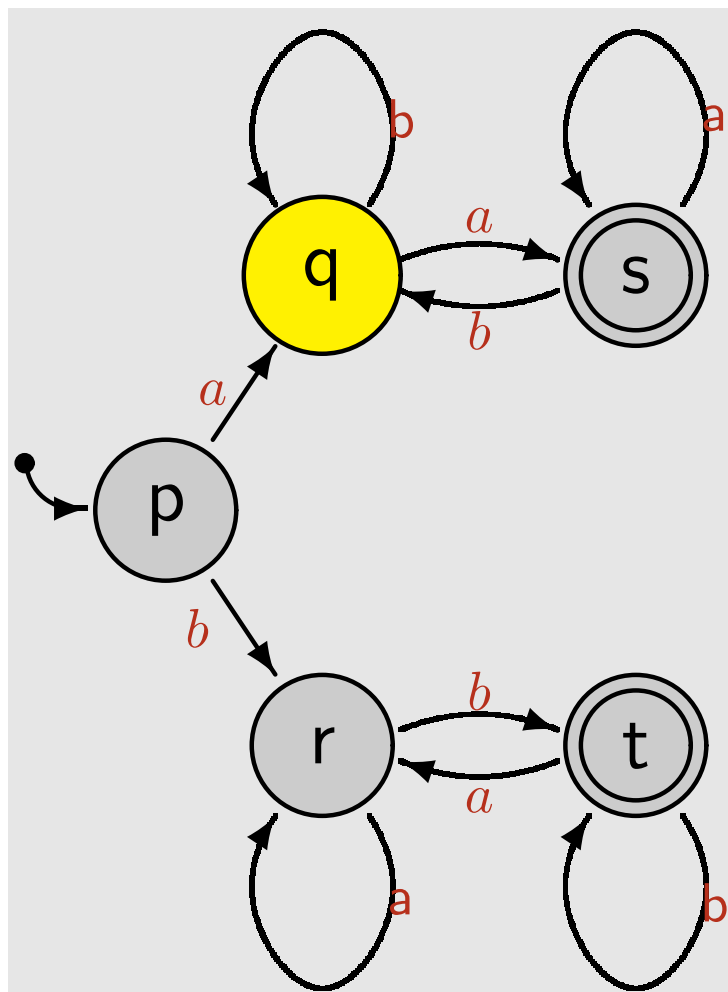


Słowo *abab*:

$$\delta^*\langle p, abab \rangle = \delta^*\langle q, bab \rangle$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone

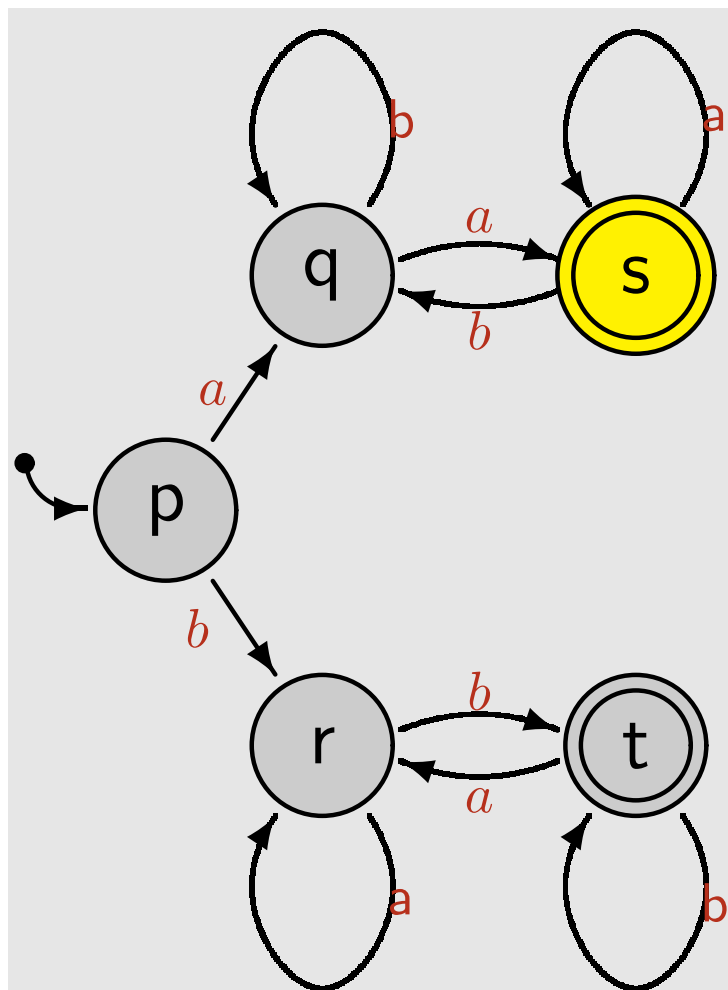


Słowo *abab*:

$$\delta^*\langle p, abab \rangle = \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone

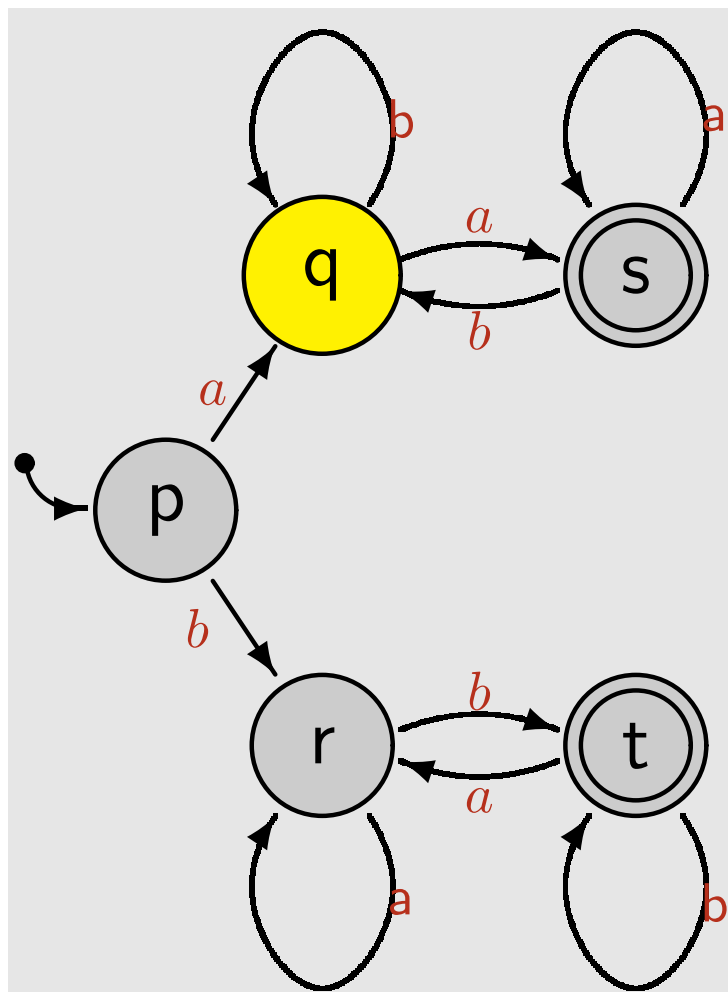


Słowo *abab*:

$$\begin{aligned} \delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle \end{aligned}$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone

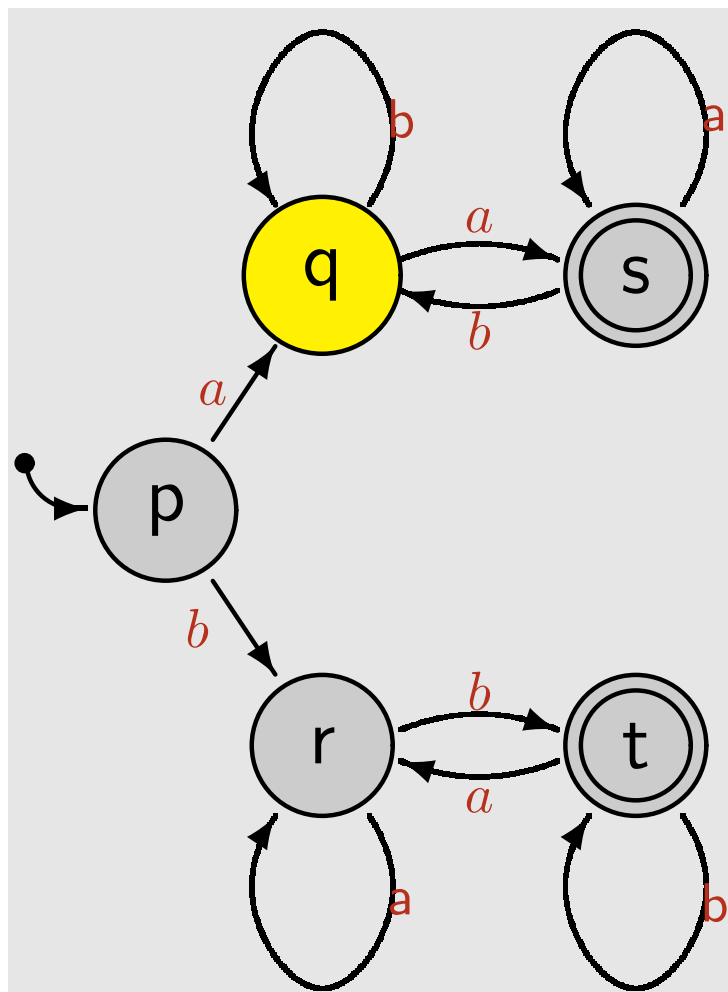


Słowo *abab*:

$$\begin{aligned} \delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle \end{aligned}$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



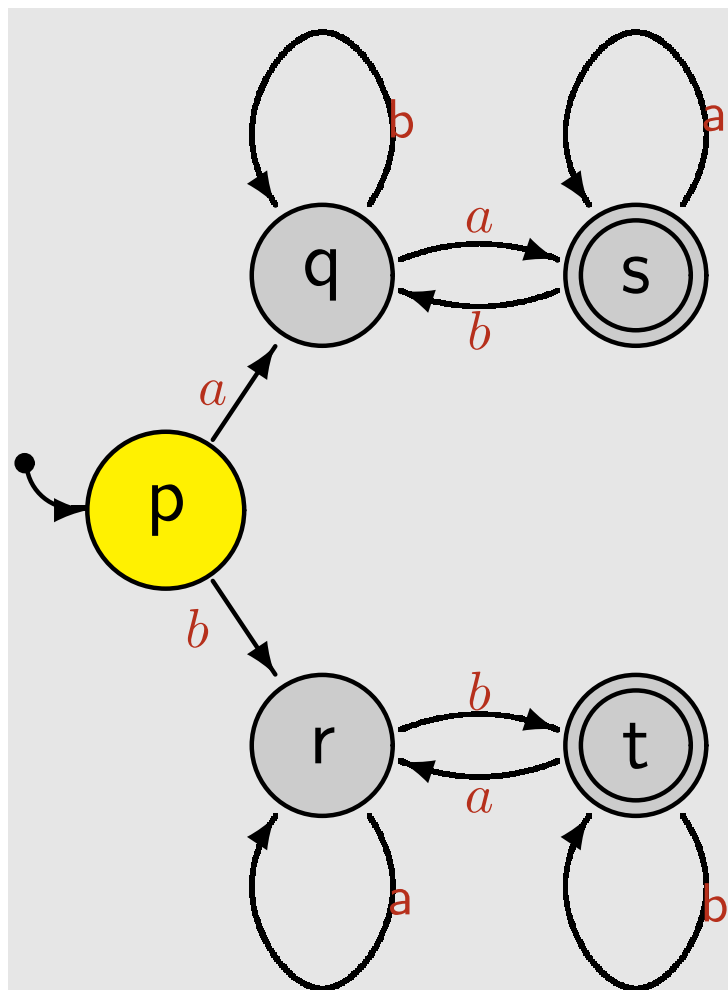
Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

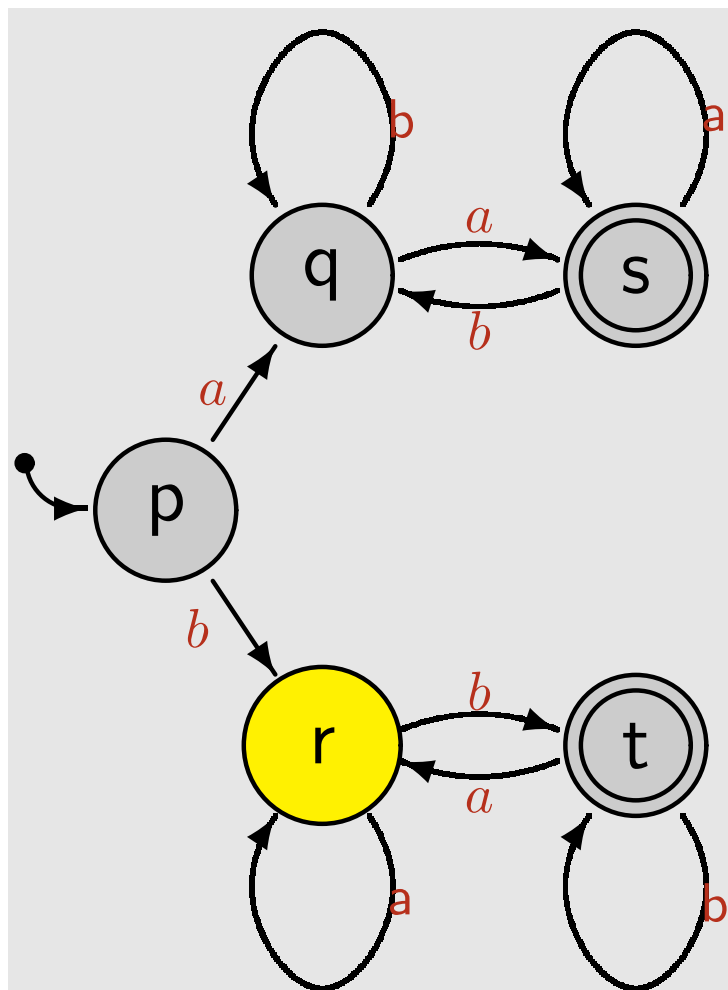
odrzucone.

Słowo *baab*:

$$\delta^*\langle p, baab \rangle$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

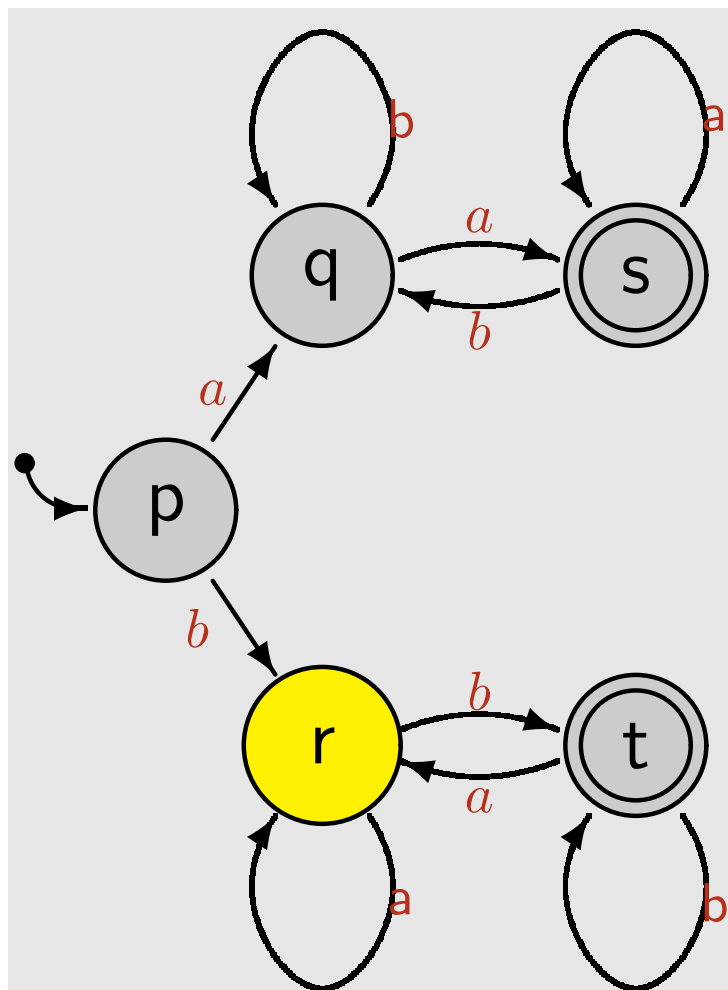
odrzucone.

Słowo *baab*:

$$\delta^*\langle p, baab \rangle = \delta^*\langle r, aab \rangle$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

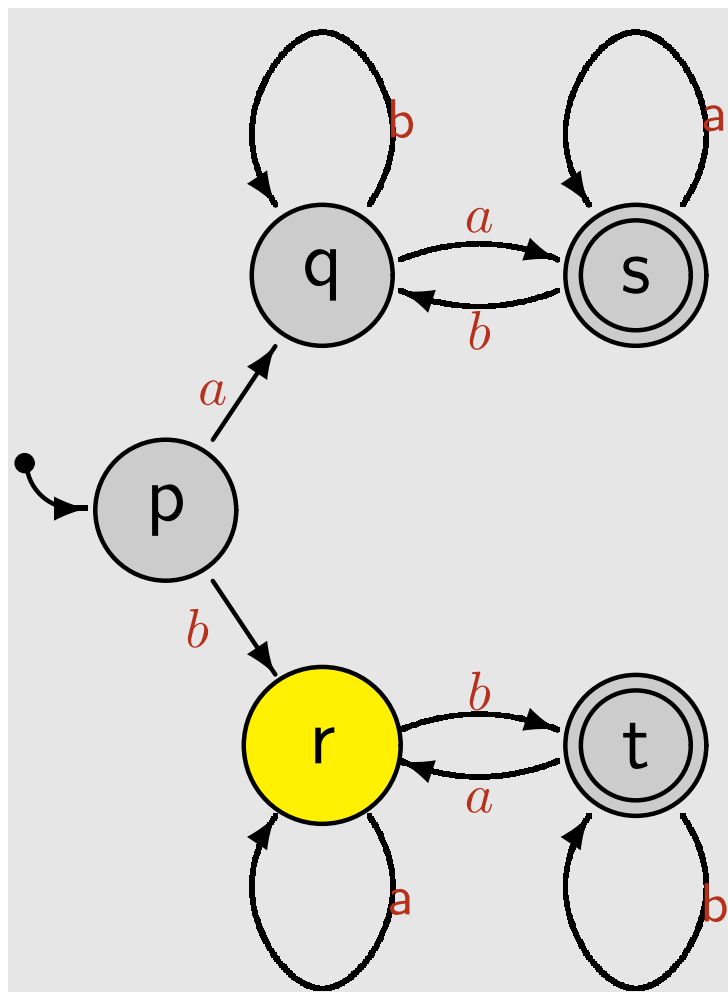
odrzucone.

Słowo *baab*:

$$\delta^*\langle p, baab \rangle = \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

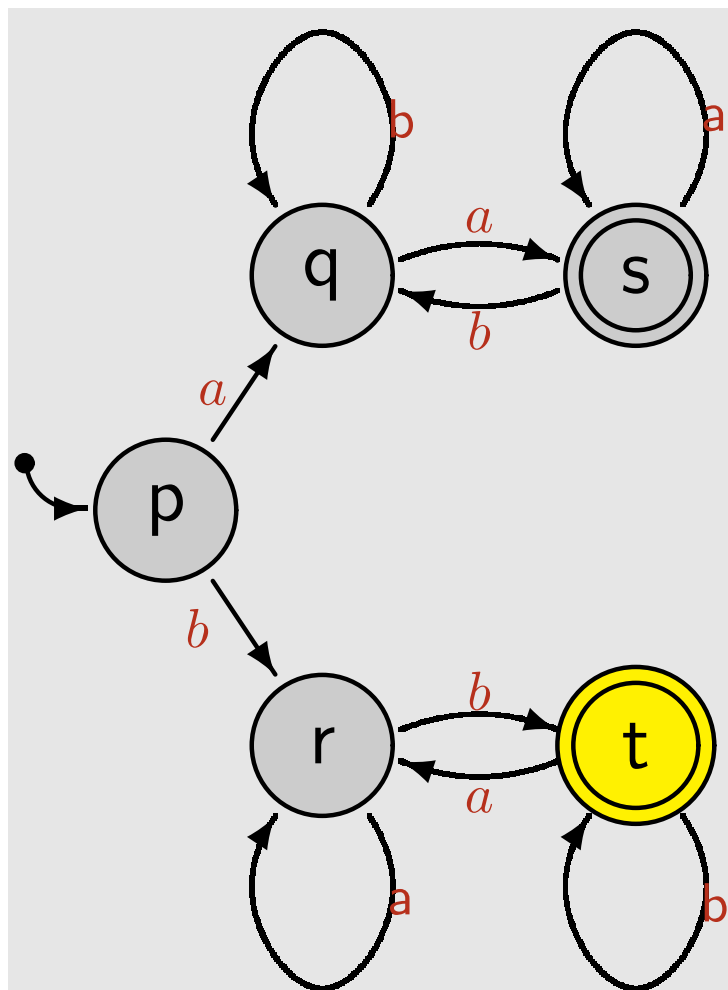
odrzucone.

Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle\end{aligned}$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

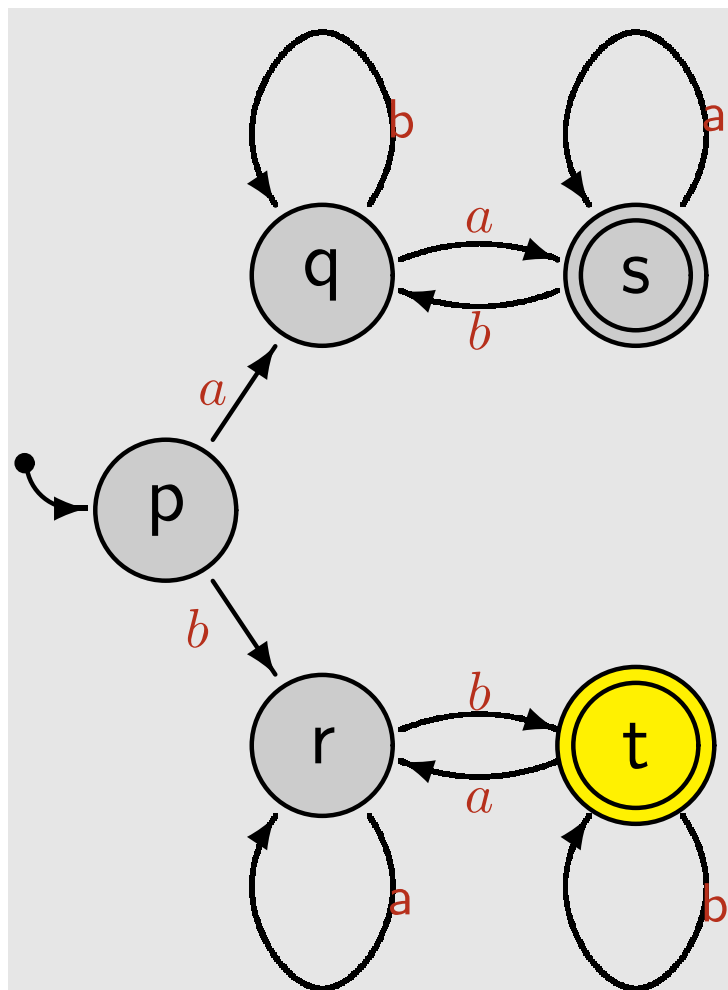
odrzucone.

Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle\end{aligned}$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

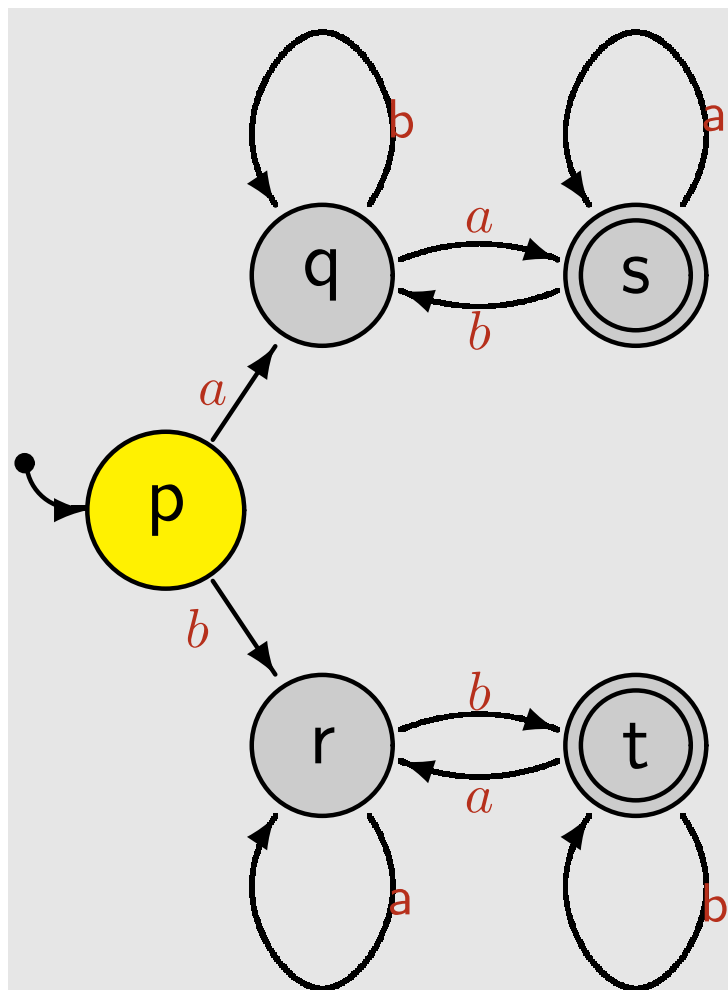
Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Słowo *baab*:

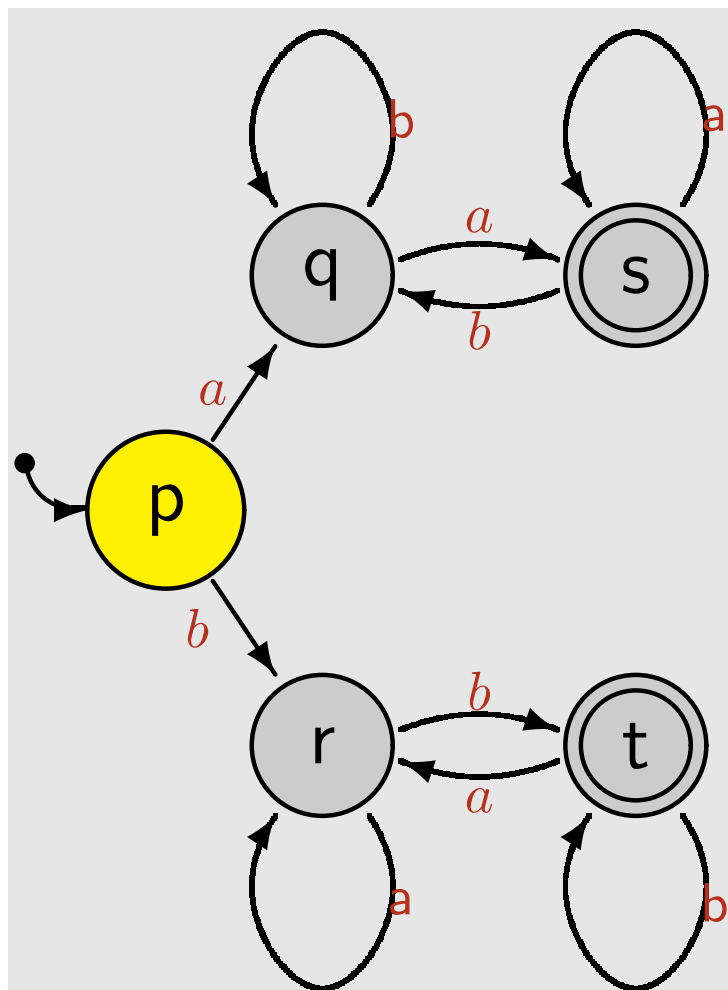
$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

Słowo λ : $\delta^*\langle p, \lambda \rangle$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Słowo *baab*:

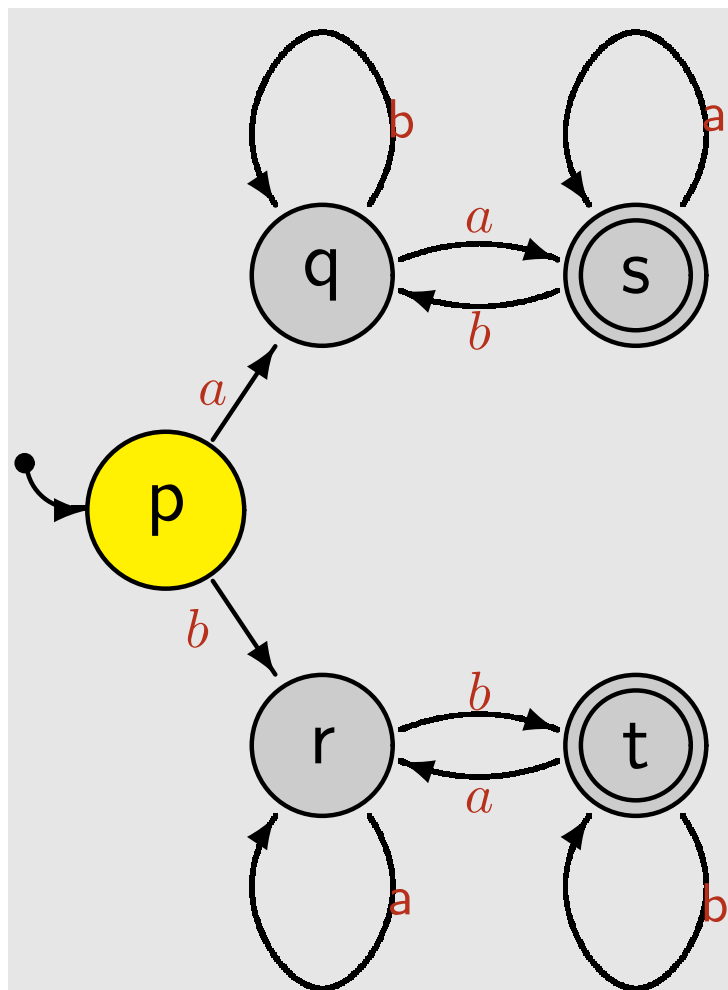
$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

Słowo λ : $\delta^*\langle p, \lambda \rangle = p$ *odrzucone.*

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

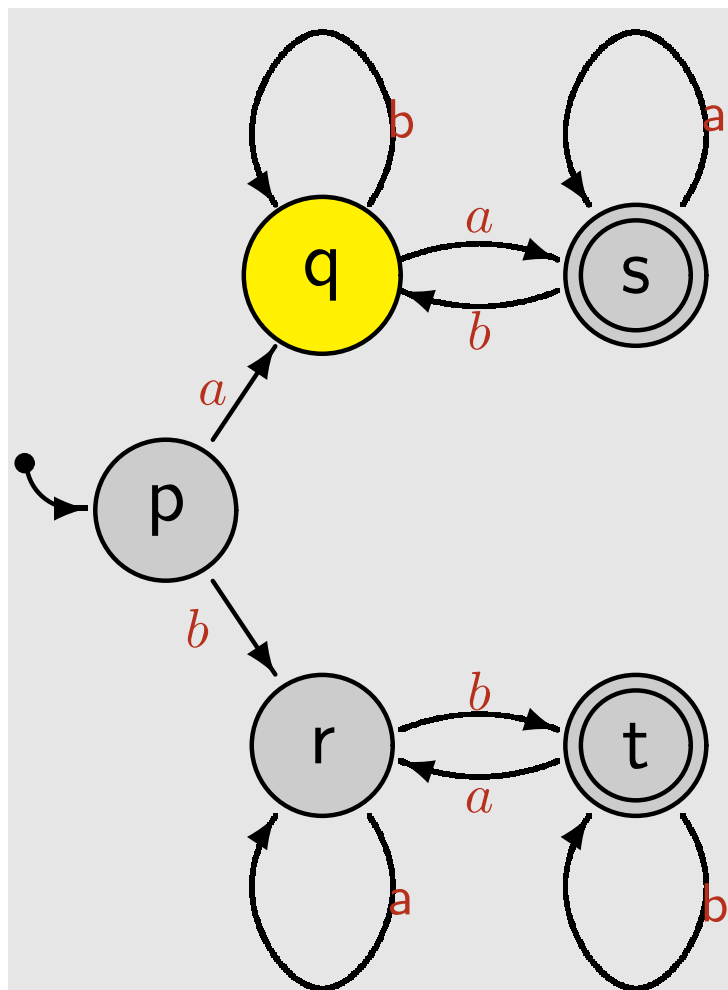
Słowo λ : $\delta^*\langle p, \lambda \rangle = p$ *odrzucone.*

Słowo *abbaba*:

$$\delta^*\langle p, abbaba \rangle$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

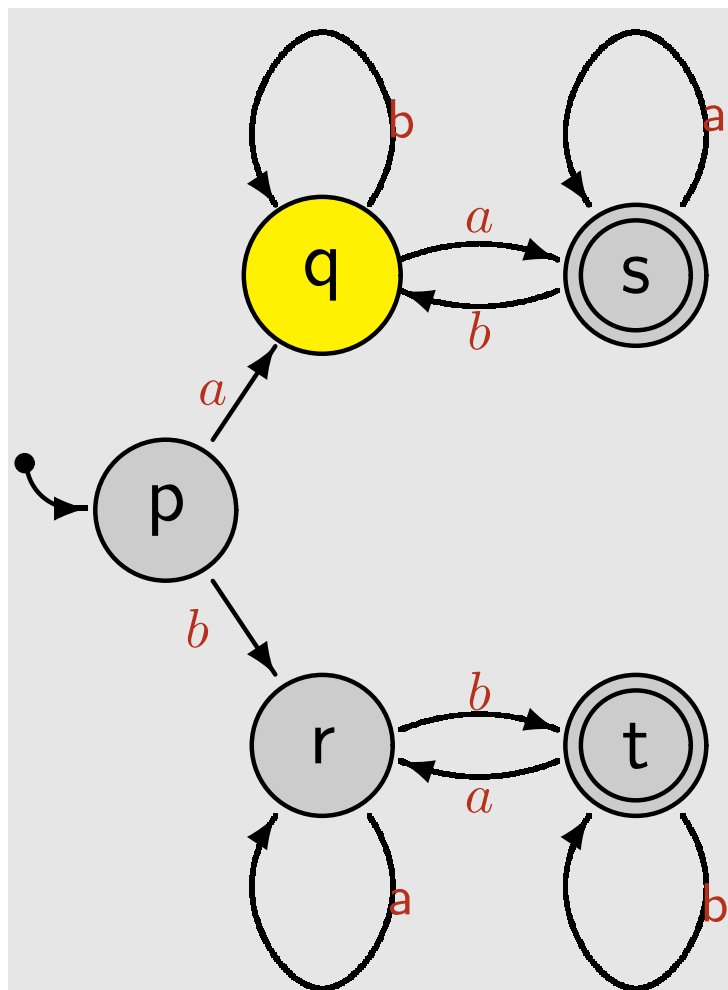
Słowo λ : $\delta^*\langle p, \lambda \rangle = p$ *odrzucone.*

Słowo *abbaba*:

$$\delta^*\langle p, abbaba \rangle = \delta^*\langle q, bbaba \rangle$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

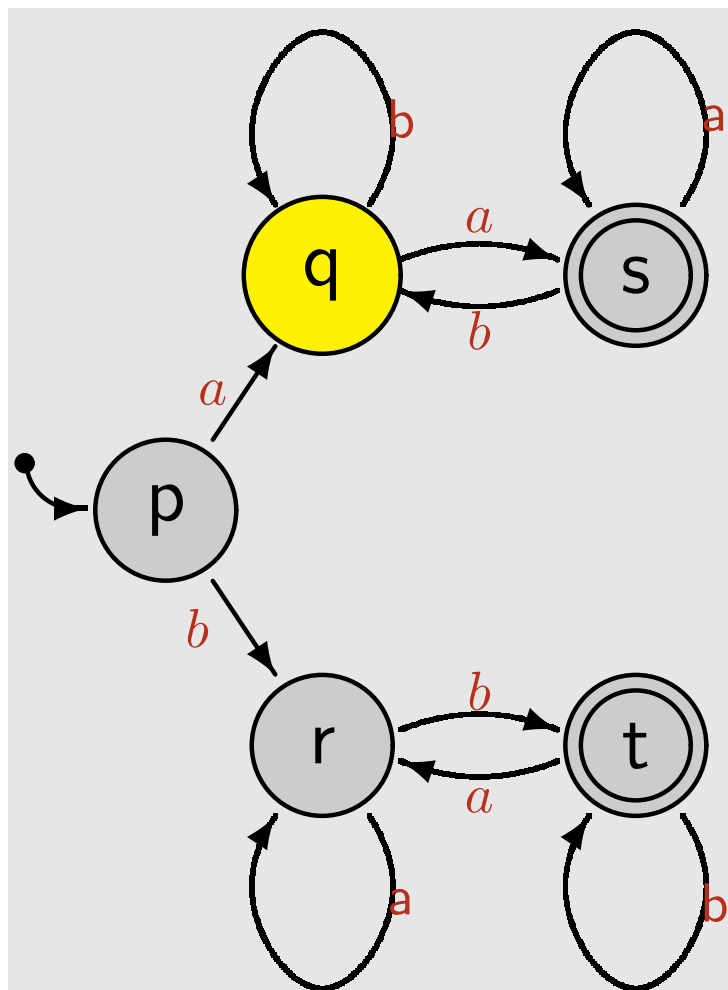
Słowo λ : $\delta^*\langle p, \lambda \rangle = p$ *odrzucone.*

Słowo *abbaba*:

$$\begin{aligned}\delta^*\langle p, abbaba \rangle &= \delta^*\langle q, bbaba \rangle \\ &= \delta^*\langle q, baba \rangle\end{aligned}$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

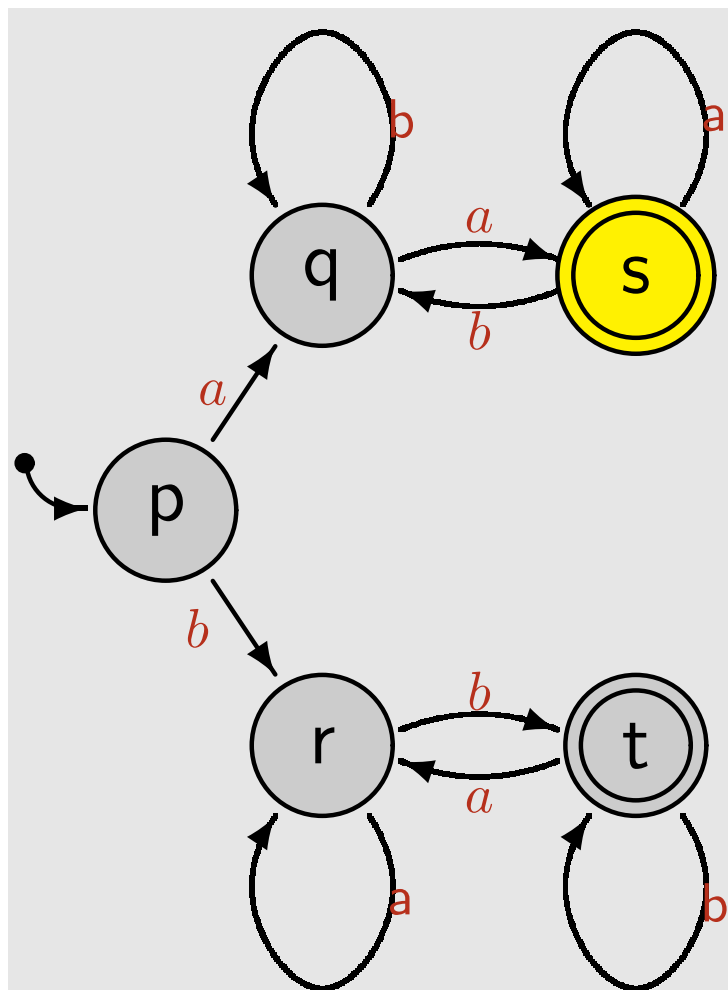
Słowo λ : $\delta^*\langle p, \lambda \rangle = p$ *odrzucone.*

Słowo *abbaba*:

$$\begin{aligned}\delta^*\langle p, abbaba \rangle &= \delta^*\langle q, bbaba \rangle \\ &= \delta^*\langle q, baba \rangle = \delta^*\langle q, aba \rangle\end{aligned}$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

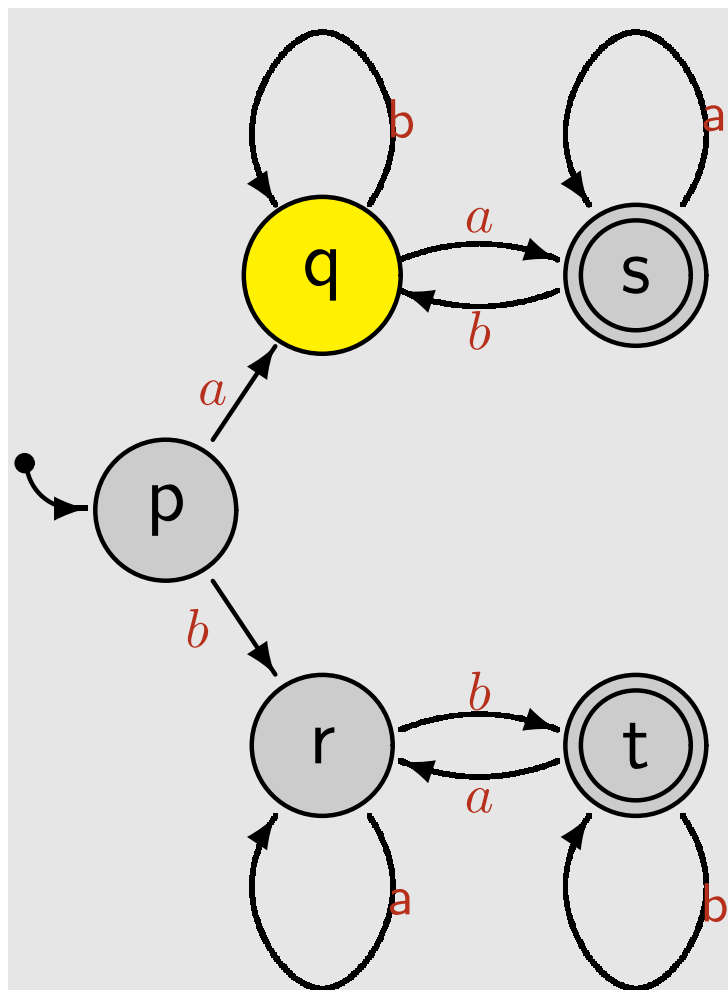
Słowo λ : $\delta^*\langle p, \lambda \rangle = p$ odrzucone.

Słowo *abbaba*:

$$\begin{aligned}\delta^*\langle p, abbaba \rangle &= \delta^*\langle q, bbaba \rangle \\ &= \delta^*\langle q, baba \rangle = \delta^*\langle q, aba \rangle \\ &= \delta^*\langle s, ba \rangle\end{aligned}$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

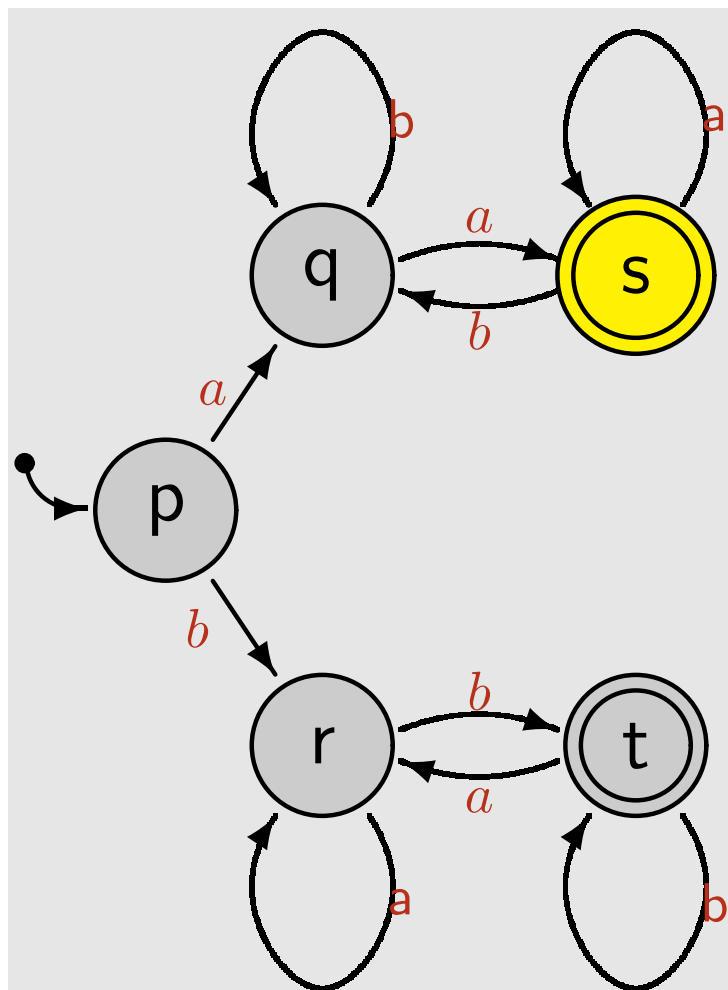
Słowo λ : $\delta^*\langle p, \lambda \rangle = p$ odrzucone.

Słowo *abbaba*:

$$\begin{aligned}\delta^*\langle p, abbaba \rangle &= \delta^*\langle q, bbaba \rangle \\ &= \delta^*\langle q, baba \rangle = \delta^*\langle q, aba \rangle \\ &= \delta^*\langle s, ba \rangle = \delta^*\langle q, a \rangle\end{aligned}$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

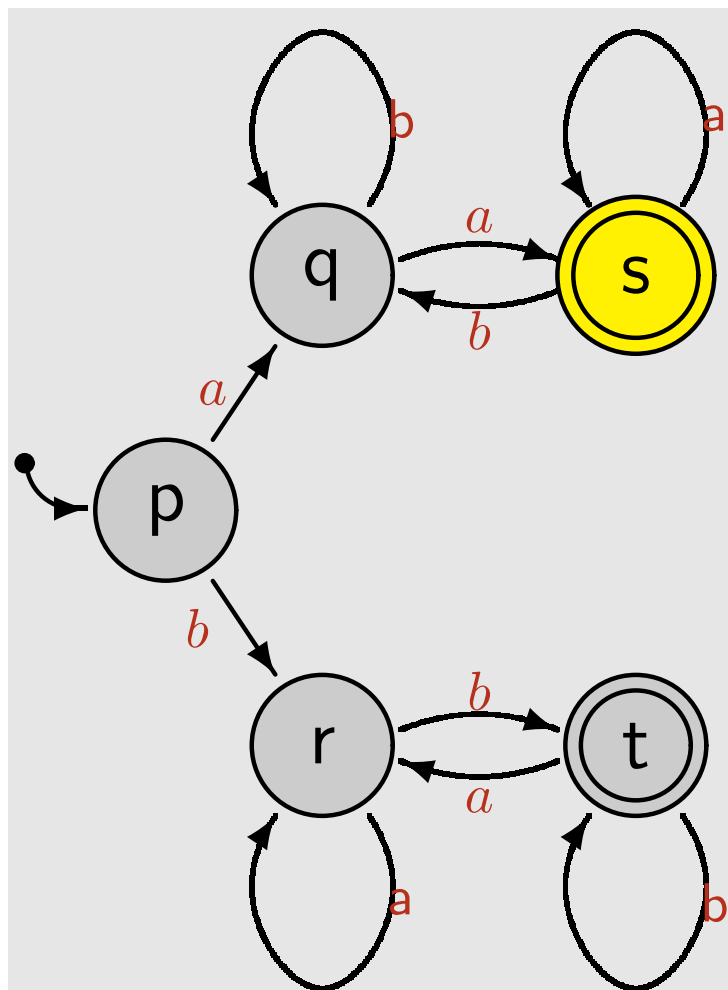
Słowo λ : $\delta^*\langle p, \lambda \rangle = p$ *odrzucone*.

Słowo *abbaba*:

$$\begin{aligned}\delta^*\langle p, abbaba \rangle &= \delta^*\langle q, bbaba \rangle \\ &= \delta^*\langle q, baba \rangle = \delta^*\langle q, aba \rangle \\ &= \delta^*\langle s, ba \rangle = \delta^*\langle q, a \rangle = \delta^*\langle s, \lambda \rangle\end{aligned}$$

Akceptacja słowa — od stanu początkowego do akceptującego.

Akceptory: automaty skończone



Słowo *abab*:

$$\begin{aligned}\delta^*\langle p, abab \rangle &= \delta^*\langle q, bab \rangle = \delta^*\langle q, ab \rangle \\ &= \delta^*\langle s, b \rangle = \delta^*\langle q, \lambda \rangle = q\end{aligned}$$

odrzucone.

Słowo *baab*:

$$\begin{aligned}\delta^*\langle p, baab \rangle &= \delta^*\langle r, aab \rangle = \delta^*\langle r, ab \rangle \\ &= \delta^*\langle r, b \rangle = \delta^*\langle t, \lambda \rangle = t\end{aligned}$$

zaakceptowane.

Słowo λ : $\delta^*\langle p, \lambda \rangle = p$ odrzucone.

Słowo *abbaba*:

$$\begin{aligned}\delta^*\langle p, abbaba \rangle &= \delta^*\langle q, bbaba \rangle \\ &= \delta^*\langle q, baba \rangle = \delta^*\langle q, aba \rangle \\ &= \delta^*\langle s, ba \rangle = \delta^*\langle q, a \rangle = \delta^*\langle s, \lambda \rangle = s\end{aligned}$$

zaakceptowane.

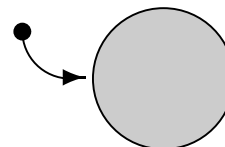
Akceptacja słowa — od stanu początkowego do akceptującego.

Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*

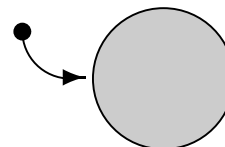
Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*



Konstrukcja automatu skończonego

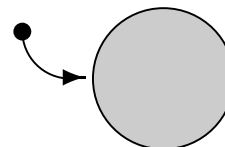
- dokładnie jeden stan *początkowy*



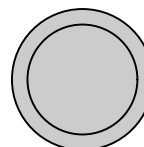
- dowolnie wiele stanów *końcowych*
(*akceptujących*)

Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*

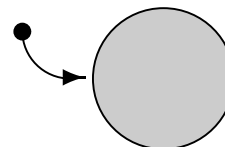


- dowolnie wiele stanów *końcowych*
(*akceptujących*)

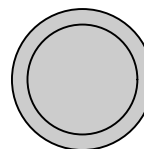


Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*



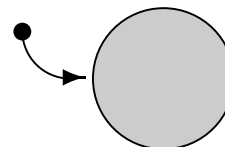
- dowolnie wiele stanów *końcowych*
(*akceptujących*)



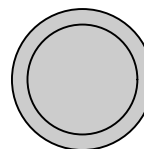
- między stanami *strzałki* (*tranzycje*)

Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*



- dowolnie wiele stanów *końcowych*
(*akceptujących*)

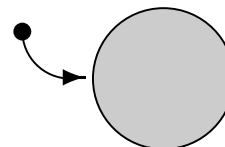


- między stanami *strzałki* (*tranzycje*)

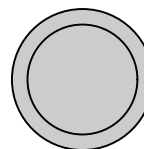


Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*



- dowolnie wiele stanów *końcowych*
(*akceptujących*)



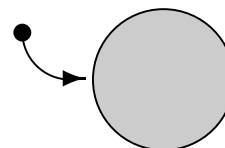
- między stanami *strzałki* (*tranzycje*)



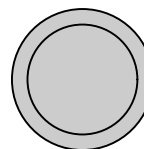
- na strzałkach stoją **pojedyncze** litery alfabetu

Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*



- dowolnie wiele stanów *końcowych*
(*akceptujących*)



- między stanami *strzałki* (*tranzycje*)

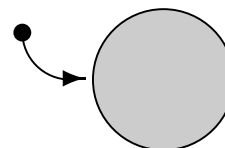


- na strzałkach stoją **pojedyncze** litery alfabetu

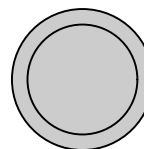


Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*



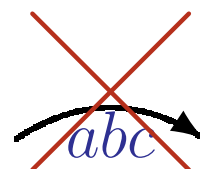
- dowolnie wiele stanów *końcowych*
(*akceptujących*)



- między stanami *strzałki* (*tranzycje*)

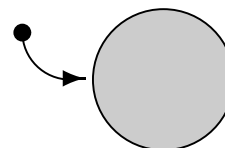


- na strzałkach stoją **pojedyncze** litery alfabetu

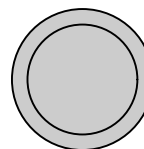


Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*



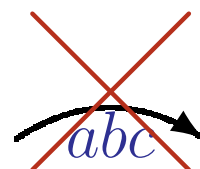
- dowolnie wiele stanów *końcowych*
(*akceptujących*)



- między stanami *strzałki* (*tranzycje*)

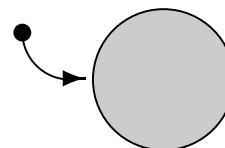


- na strzałkach stoją **pojedyncze** litery alfabetu

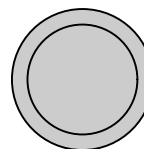


Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*



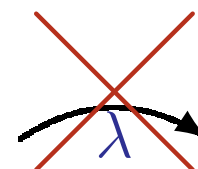
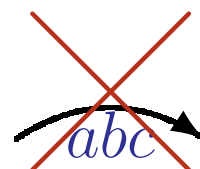
- dowolnie wiele stanów *końcowych*
(*akceptujących*)



- między stanami *strzałki* (*tranzycje*)

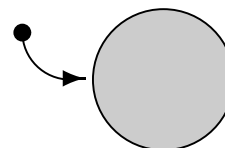


- na strzałkach stoją **pojedyncze** litery alfabetu

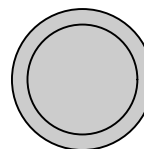


Konstrukcja automatu skończonego

- dokładnie jeden stan *początkowy*



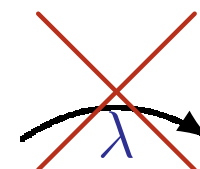
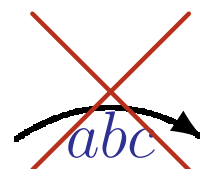
- dowolnie wiele stanów *końcowych*
(*akceptujących*)



- między stanami *strzałki* (*tranzycje*)

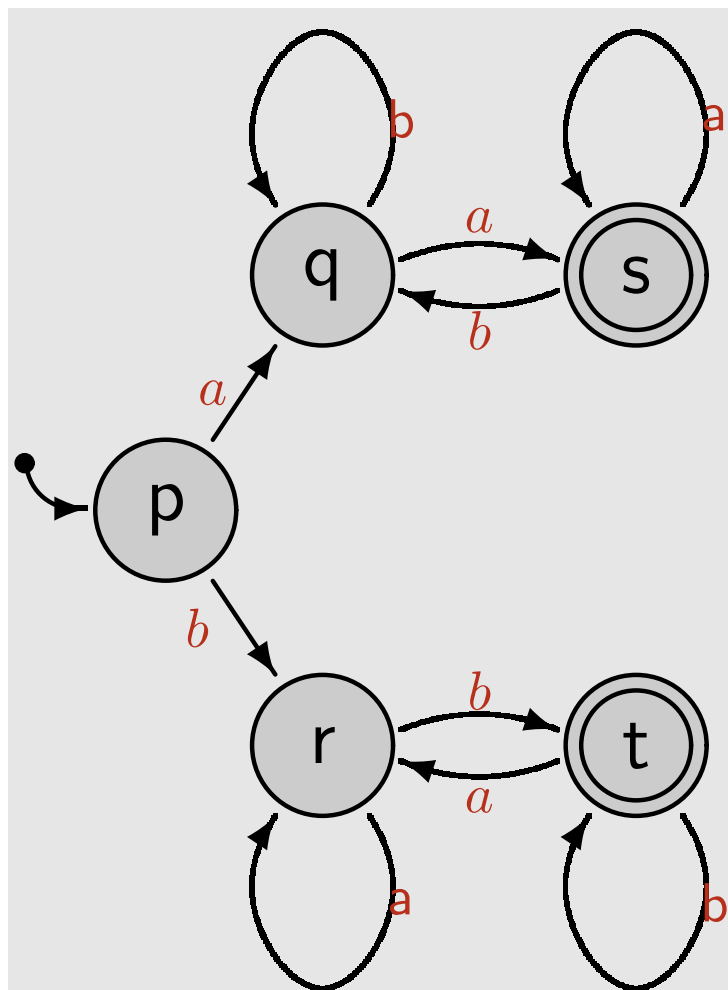


- na strzałkach stoją **pojedyncze** litery alfabetu



- z każdego stanu wychodzi po **jednej** strzałce dla **każdej** litery alfabetu

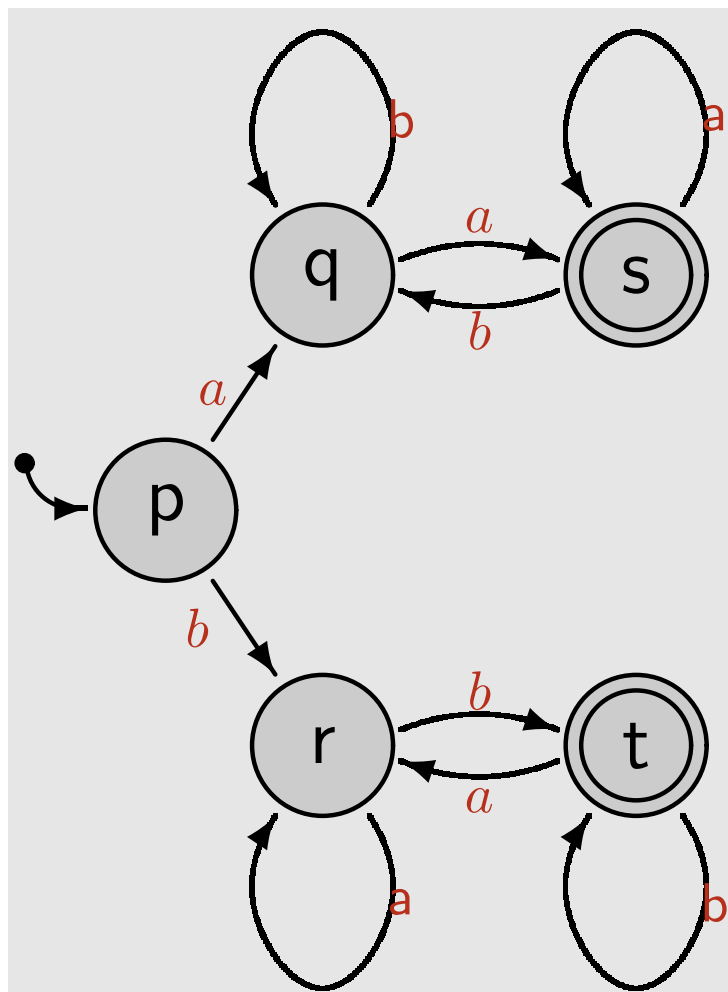
Akceptory: automaty skończone



Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

Akceptory: automaty skończone

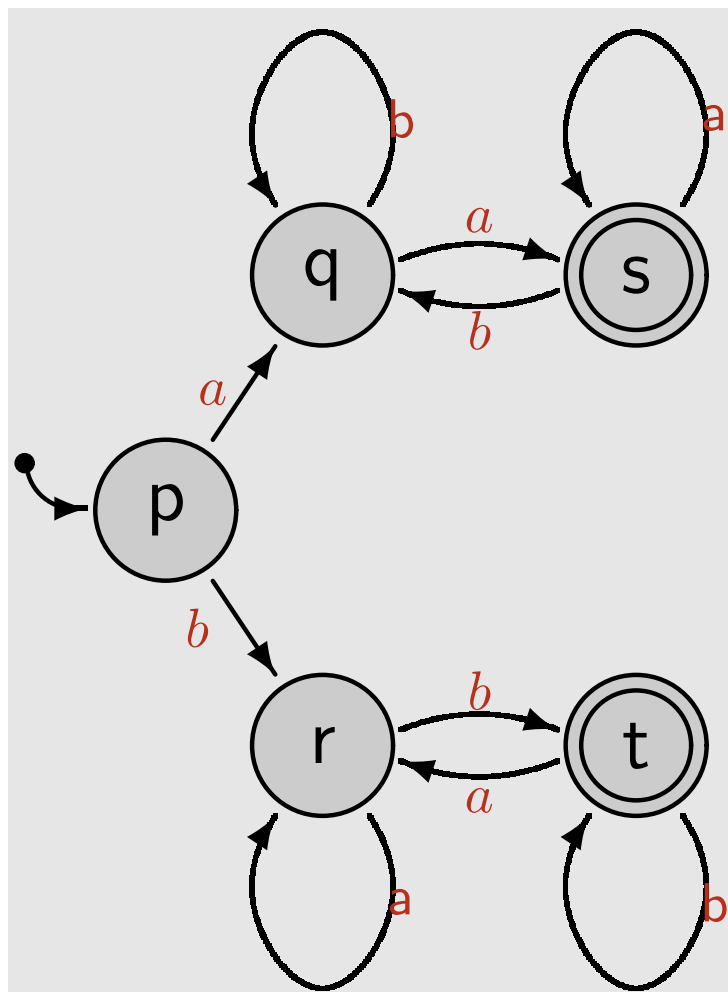


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\delta^* \langle p, baab \rangle$$

Akceptory: automaty skończone

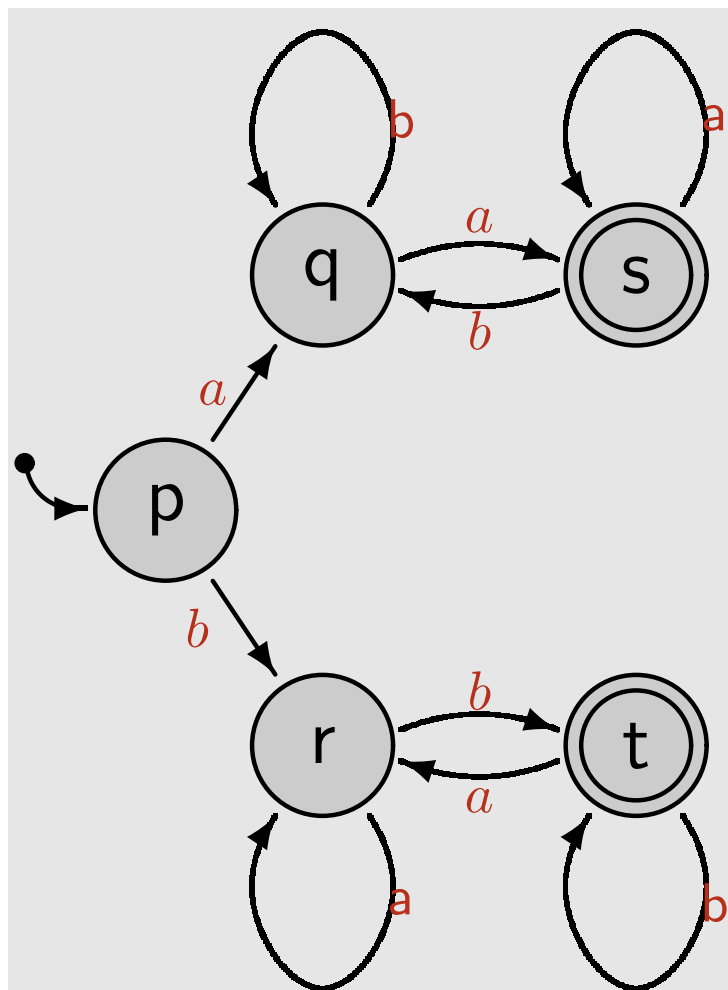


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\delta^* \langle p, baab \rangle = \delta^* \langle \delta \langle p, b \rangle, aab \rangle$$

Akceptory: automaty skończone

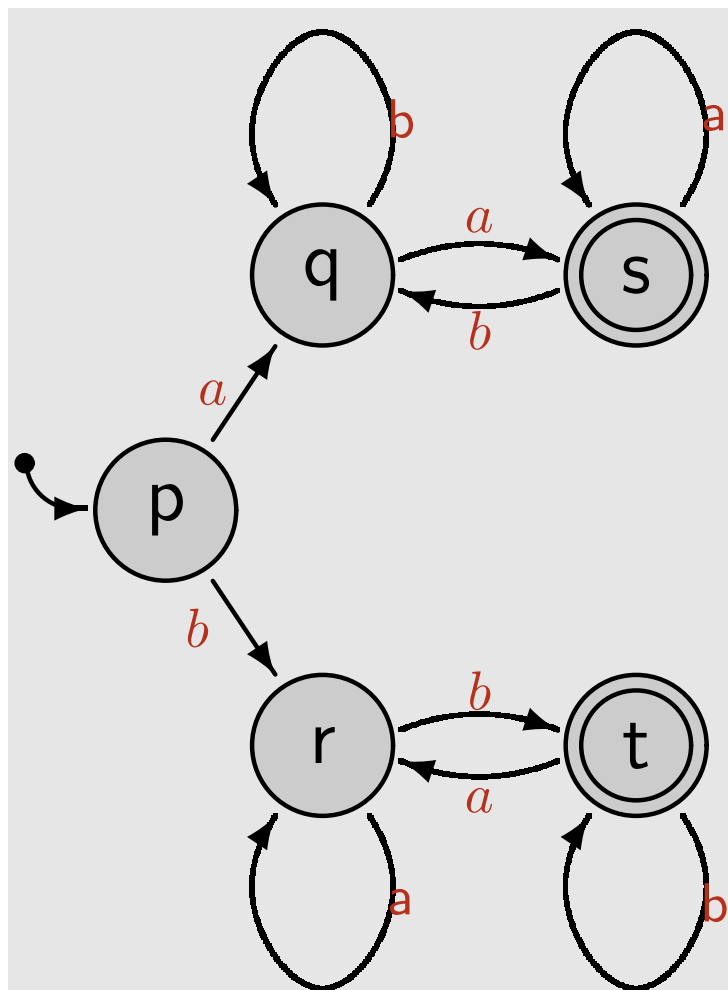


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\delta^* \langle p, baab \rangle = \delta^* \langle \delta \langle p, b \rangle, aab \rangle$$

Akceptory: automaty skończone

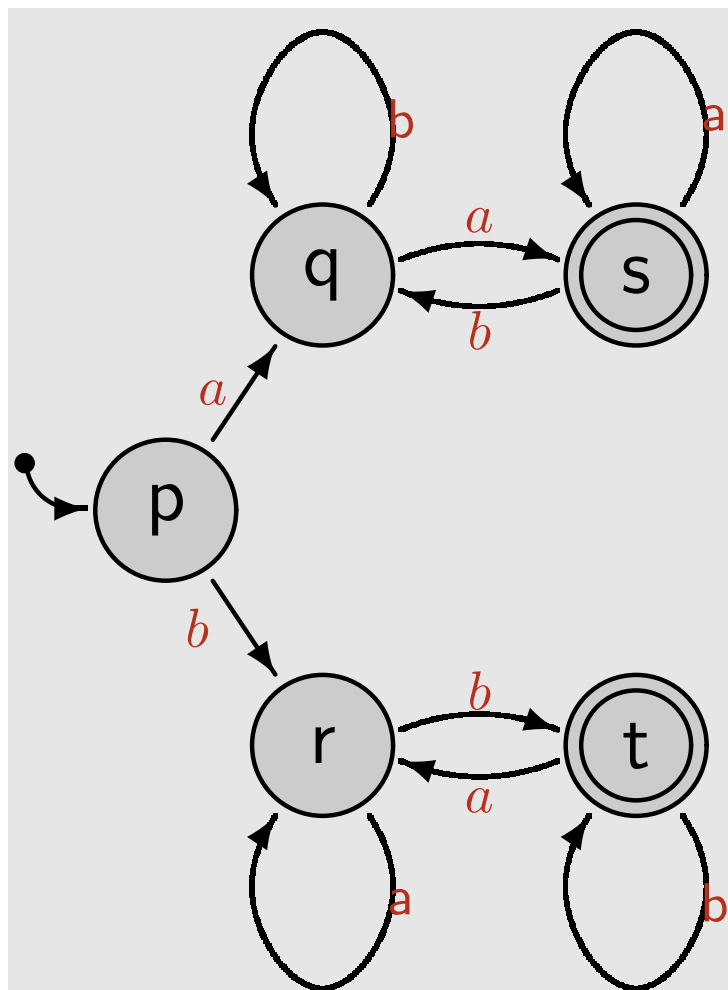


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\begin{aligned}
 \delta^* \langle p, baab \rangle &= \delta^* \langle \delta \langle p, b \rangle, aab \rangle \\
 &= \delta^* \langle r, aab \rangle
 \end{aligned}$$

Akceptory: automaty skończone

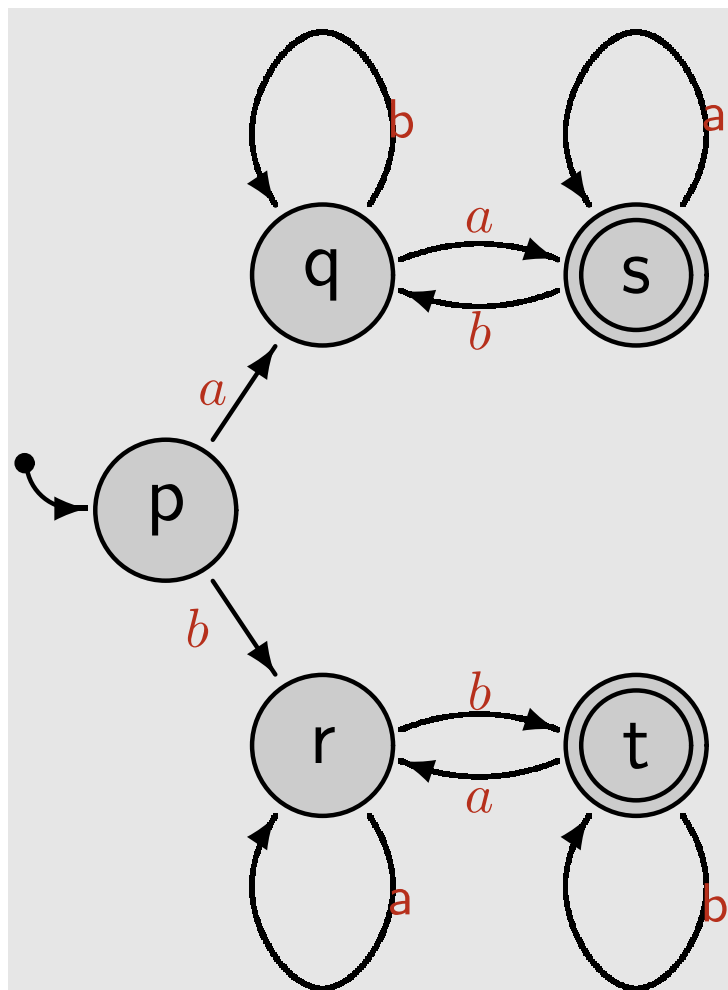


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\begin{aligned}
 \delta^* \langle p, baab \rangle &= \delta^* \langle \delta \langle p, b \rangle, aab \rangle \\
 &= \delta^* \langle r, aab \rangle = \delta^* \langle \delta \langle r, a \rangle, ab \rangle
 \end{aligned}$$

Akceptory: automaty skończone

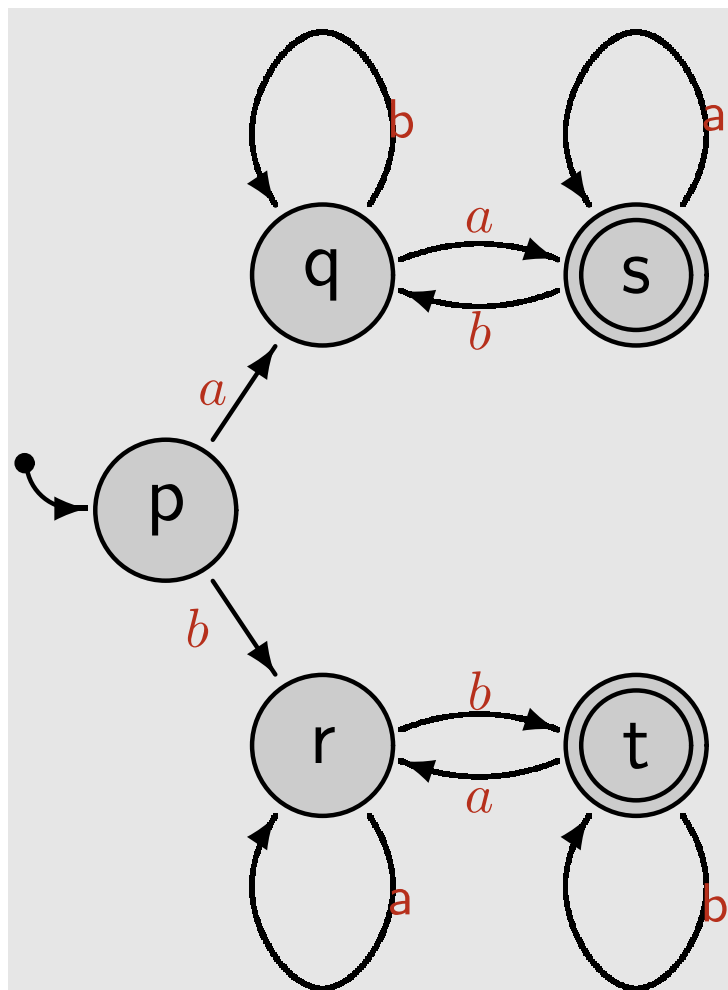


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\begin{aligned}
 \delta^* \langle p, baab \rangle &= \delta^* \langle \delta \langle p, b \rangle, aab \rangle \\
 &= \delta^* \langle r, aab \rangle = \delta^* \langle \delta \langle r, a \rangle, ab \rangle
 \end{aligned}$$

Akceptory: automaty skończone

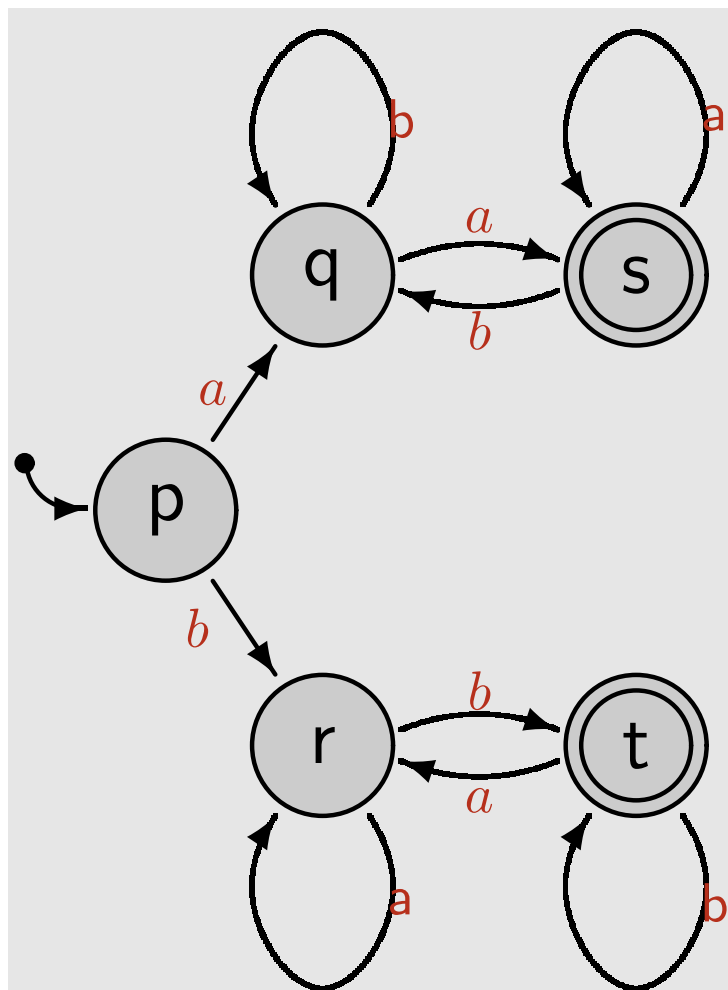


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\begin{aligned}
 \delta^* \langle p, baab \rangle &= \delta^* \langle \delta \langle p, b \rangle, aab \rangle \\
 &= \delta^* \langle r, aab \rangle = \delta^* \langle \delta \langle r, a \rangle, ab \rangle \\
 &= \delta^* \langle r, ab \rangle
 \end{aligned}$$

Akceptory: automaty skończone

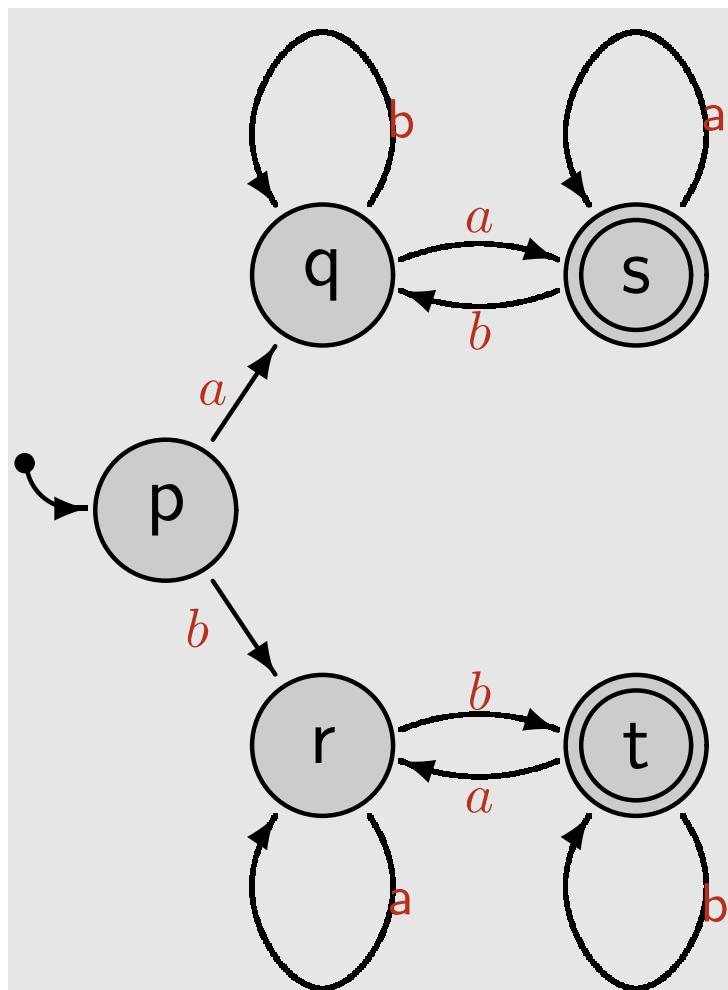


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\begin{aligned}
 \delta^* \langle p, baab \rangle &= \delta^* \langle \delta \langle p, b \rangle, aab \rangle \\
 &= \delta^* \langle r, aab \rangle = \delta^* \langle \delta \langle r, a \rangle, ab \rangle \\
 &= \delta^* \langle r, ab \rangle = \delta^* \langle \delta \langle r, a \rangle, b \rangle
 \end{aligned}$$

Akceptory: automaty skończone

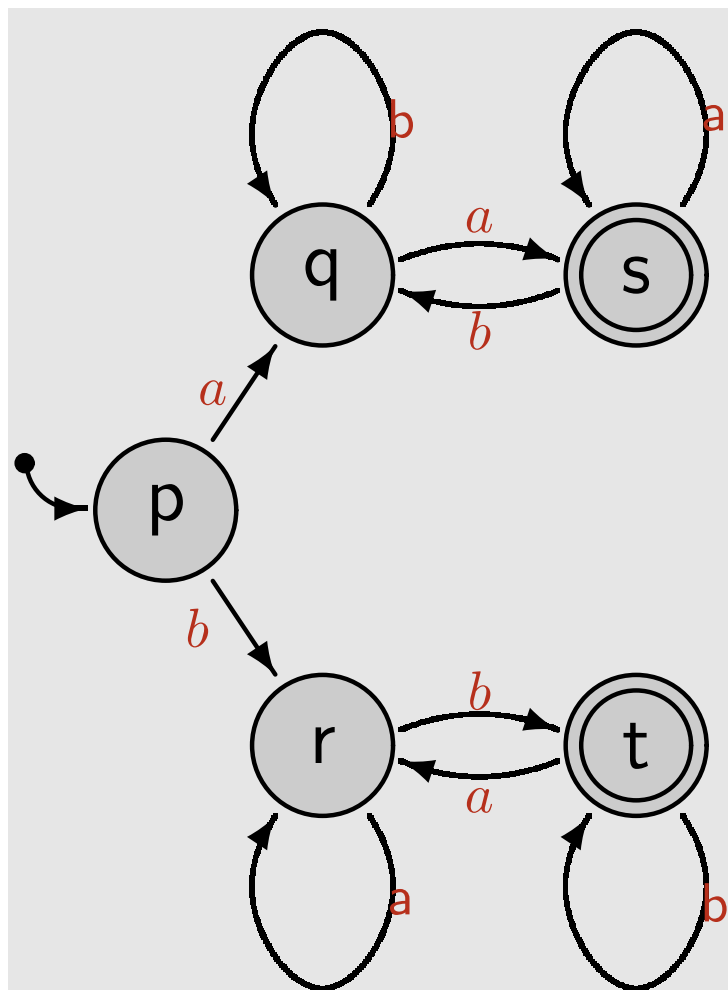


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\begin{aligned}
 \delta^* \langle p, baab \rangle &= \delta^* \langle \delta \langle p, b \rangle, aab \rangle \\
 &= \delta^* \langle r, aab \rangle = \delta^* \langle \delta \langle r, a \rangle, ab \rangle \\
 &= \delta^* \langle r, ab \rangle = \delta^* \langle \delta \langle r, a \rangle, b \rangle \\
 &= \delta^* \langle r, b \rangle
 \end{aligned}$$

Akceptory: automaty skończone

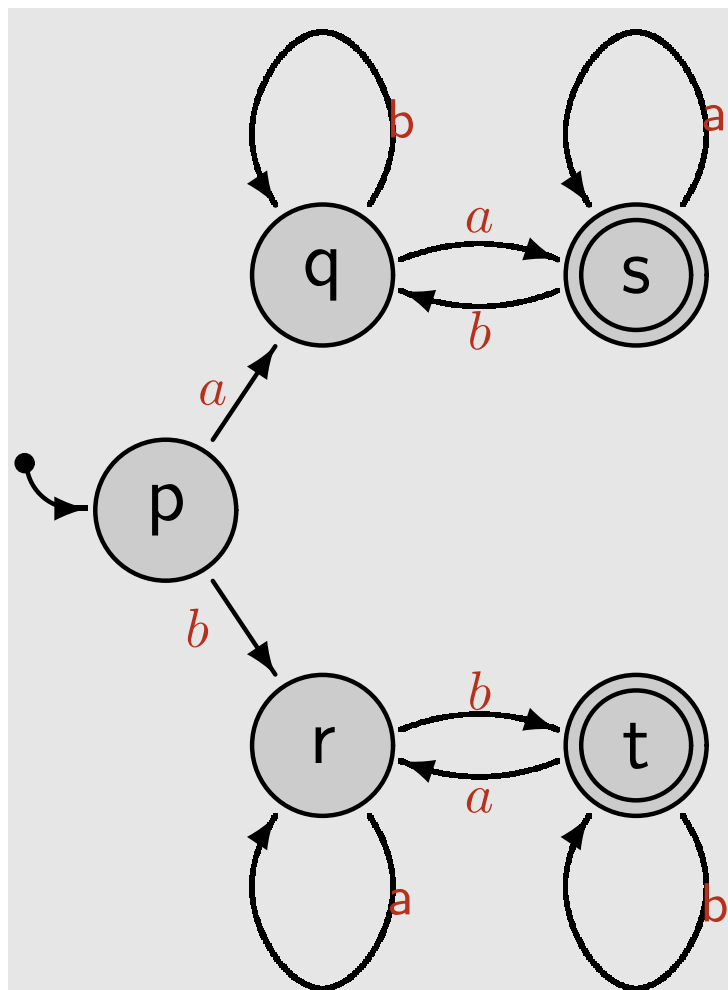


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\begin{aligned}
 \delta^* \langle p, baab \rangle &= \delta^* \langle \delta \langle p, b \rangle, aab \rangle \\
 &= \delta^* \langle r, aab \rangle = \delta^* \langle \delta \langle r, a \rangle, ab \rangle \\
 &= \delta^* \langle r, ab \rangle = \delta^* \langle \delta \langle r, a \rangle, b \rangle \\
 &= \delta^* \langle r, b \rangle = \delta^* \langle \delta \langle r, b \rangle, \lambda \rangle
 \end{aligned}$$

Akceptory: automaty skończone

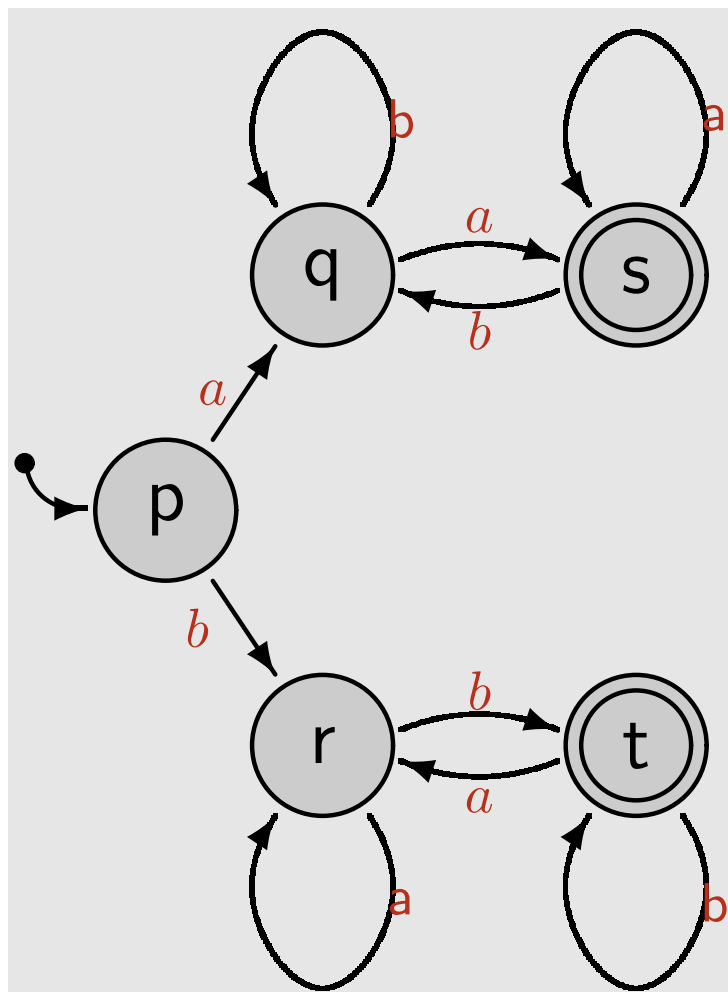


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\begin{aligned}
 \delta^* \langle p, baab \rangle &= \delta^* \langle \delta \langle p, b \rangle, aab \rangle \\
 &= \delta^* \langle r, aab \rangle = \delta^* \langle \delta \langle r, a \rangle, ab \rangle \\
 &= \delta^* \langle r, ab \rangle = \delta^* \langle \delta \langle r, a \rangle, b \rangle \\
 &= \delta^* \langle r, b \rangle = \delta^* \langle \delta \langle r, b \rangle, \lambda \rangle \\
 &= \delta^* \langle t, \lambda \rangle
 \end{aligned}$$

Akceptory: automaty skończone

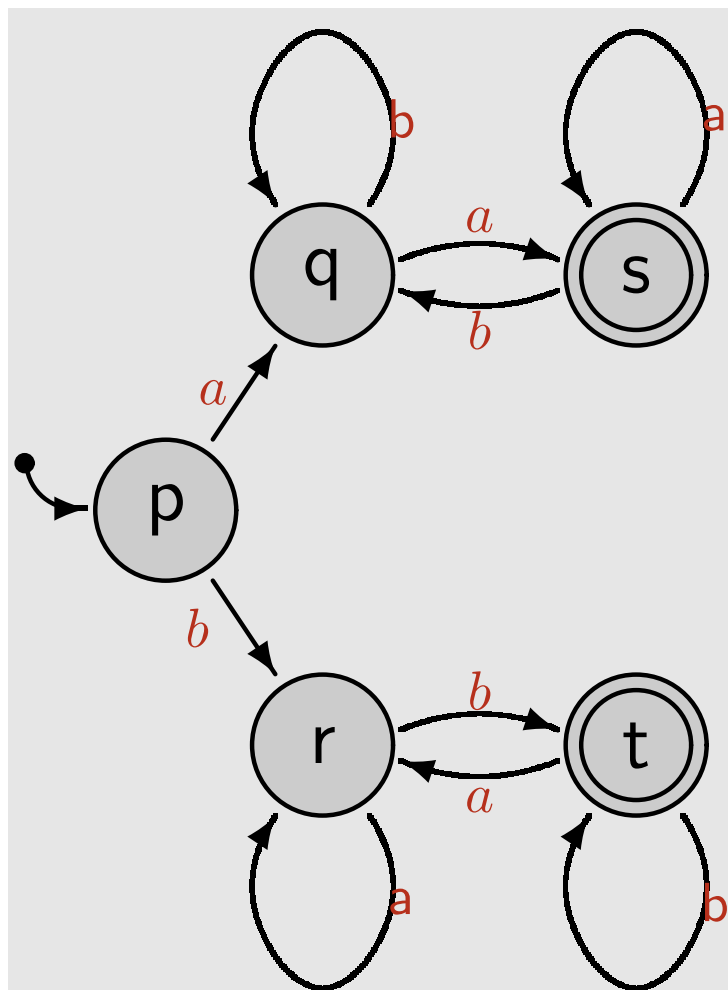


Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\begin{aligned}
 \delta^* \langle p, baab \rangle &= \delta^* \langle \delta \langle p, b \rangle, aab \rangle \\
 &= \delta^* \langle r, aab \rangle = \delta^* \langle \delta \langle r, a \rangle, ab \rangle \\
 &= \delta^* \langle r, ab \rangle = \delta^* \langle \delta \langle r, a \rangle, b \rangle \\
 &= \delta^* \langle r, b \rangle = \delta^* \langle \delta \langle r, b \rangle, \lambda \rangle \\
 &= \delta^* \langle t, \lambda \rangle = t
 \end{aligned}$$

Akceptory: automaty skończone



Funkcja przejścia automatu:

δ	a	b
p	q	r
q	s	q
r	r	t
s	s	q
t	r	t

$$\begin{aligned}
 \delta^* \langle p, baab \rangle &= \delta^* \langle \delta \langle p, b \rangle, aab \rangle \\
 &= \delta^* \langle r, aab \rangle = \delta^* \langle \delta \langle r, a \rangle, ab \rangle \\
 &= \delta^* \langle r, ab \rangle = \delta^* \langle \delta \langle r, a \rangle, b \rangle \\
 &= \delta^* \langle r, b \rangle = \delta^* \langle \delta \langle r, b \rangle, \lambda \rangle \\
 &= \delta^* \langle t, \lambda \rangle = t
 \end{aligned}$$

Akceptacja słowa w : jeśli $\delta^* \langle p, w \rangle$ jest stanem akceptującym.

Automaty skończone

DEFINICJA: Deterministyczny automat skończony M —

Automaty skończone

DEFINICJA: Deterministyczny automat skończony M —

- skończony zbiór Q *stanów*

Automaty skończone

DEFINICJA: Deterministyczny automat skończony M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter*

Automaty skończone

DEFINICJA: Deterministyczny automat skończony M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter*,
- funkcja *przejścia* $\delta : Q \times \Sigma \rightarrow Q$

Automaty skończone

DEFINICJA: Deterministyczny automat skończony M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter*,
- funkcja *przejścia* $\delta : Q \times \Sigma \rightarrow Q$,
- wyróżniony stan *początkowy* $q_0 \in Q$

Automaty skończone

DEFINICJA: Deterministyczny automat skończony M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter*,
- funkcja *przejścia* $\delta : Q \times \Sigma \rightarrow Q$,
- wyróżniony stan *początkowy* $q_0 \in Q$,
- wyróżniony zbiór stanów *końcowych* (*akceptujących*) $F \subseteq Q$.

Automaty skończone

DEFINICJA: Deterministyczny automat skończony M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter*,
- funkcja *przejścia* $\delta : Q \times \Sigma \rightarrow Q$,
- wyróżniony stan *początkowy* $q_0 \in Q$,
- wyróżniony zbiór stanów *końcowych* (*akceptujących*) $F \subseteq Q$.

DEFINICJA: Uogólniona funkcja przejścia — $\delta^* : Q \times \Sigma^* \rightarrow Q$:

$$\delta^* \langle q, w \rangle \stackrel{\text{def}}{=} \begin{cases} q & \text{jeśli } w = \lambda \\ \delta^* \langle \delta \langle q, \sigma \rangle, w' \rangle & \text{jeśli } w = \sigma w' \\ & \text{dla pewnych } \sigma \in \Sigma \text{ i } w' \in \Sigma^* \end{cases}$$

Automaty skończone

DEFINICJA: Deterministyczny automat skończony M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter*,
- funkcja *przejścia* $\delta : Q \times \Sigma \rightarrow Q$,
- wyróżniony stan *początkowy* $q_0 \in Q$,
- wyróżniony zbiór stanów *końcowych* (*akceptujących*) $F \subseteq Q$.

DEFINICJA: Uogólniona funkcja przejścia — $\delta^* : Q \times \Sigma^* \rightarrow Q$:

$$\delta^*\langle q, w \rangle \stackrel{\text{def}}{=} \begin{cases} q & \text{jeśli } w = \lambda \\ \delta^*\langle \delta\langle q, \sigma \rangle, w' \rangle & \text{jeśli } w = \sigma w' \\ & \text{dla pewnych } \sigma \in \Sigma \text{ i } w' \in \Sigma^* \end{cases}$$

DEFINICJA: Automat M akceptuje słowo w jeśli $\delta^*\langle q_0, w \rangle \in F$

Automaty skończone

DEFINICJA: Deterministyczny automat skończony M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter*,
- funkcja *przejścia* $\delta : Q \times \Sigma \rightarrow Q$,
- wyróżniony stan *początkowy* $q_0 \in Q$,
- wyróżniony zbiór stanów *końcowych* (*akceptujących*) $F \subseteq Q$.

DEFINICJA: Uogólniona funkcja przejścia — $\delta^* : Q \times \Sigma^* \rightarrow Q$:

$$\delta^*\langle q, w \rangle \stackrel{\text{def}}{=} \begin{cases} q & \text{jeśli } w = \lambda \\ \delta^*\langle \delta\langle q, \sigma \rangle, w' \rangle & \text{jeśli } w = \sigma w' \\ & \text{dla pewnych } \sigma \in \Sigma \text{ i } w' \in \Sigma^* \end{cases}$$

DEFINICJA: Automat M akceptuje słowo w jeśli $\delta^*\langle q_0, w \rangle \in F$

DEFINICJA: Język akceptowany przez automat $M = \langle Q, \Sigma, \delta, q_0, F \rangle$:

$$L(M) \stackrel{\text{def}}{=} \{w \in \Sigma^* \mid M \text{ akceptuje } w\} = \{w \in \Sigma^* \mid \delta^*\langle q_0, w \rangle \in F\}$$

Wyrażenia regularne a automaty skończone

Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$0 \mid 1 \{0 \mid 1\}^* \longleftarrow \text{wyr.reg.}$$

Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(0 \mid 1 \{0 \mid 1\}^*) \longleftarrow \text{wyr.reg.}$$

Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(\overbrace{0 \mid 1 \{0 \mid 1\}^*}) = \{0\} \cup \{1w \mid w \in \{0, 1\}^*\}$$

wyr.reg. ←

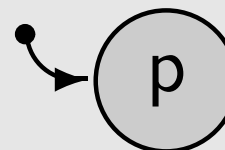
Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(\overbrace{0 \mid 1 \{0 \mid 1\}^*}) = \{0\} \cup \{1w \mid w \in \{0, 1\}^*\}$$

wyr.reg.



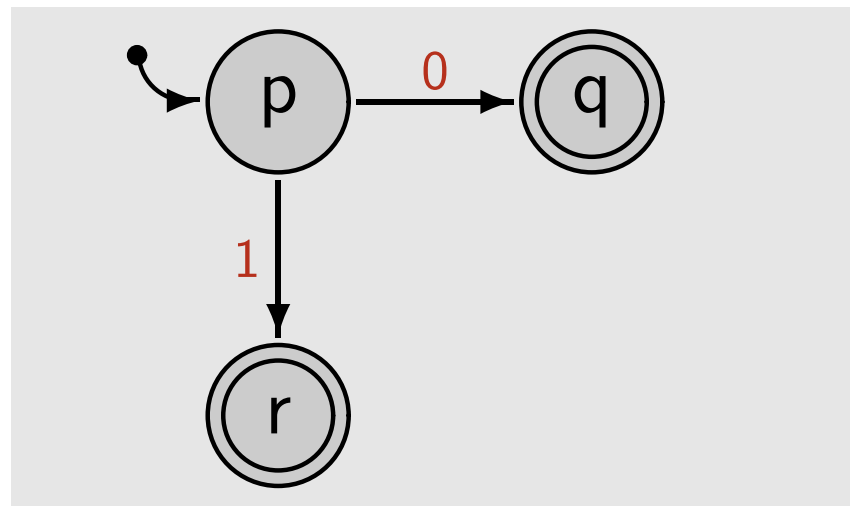
Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(\overbrace{0 \mid 1 \{0 \mid 1\}^*}) = \{0\} \cup \{1w \mid w \in \{0, 1\}^*\}$$

wyr.reg.



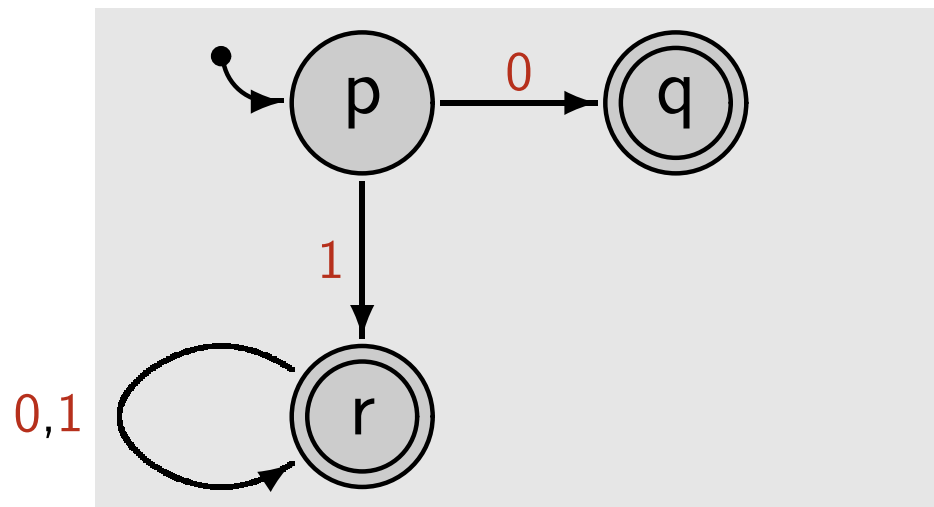
Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(\overbrace{0 \mid 1 \{0 \mid 1\}^*}) = \{0\} \cup \{1w \mid w \in \{0,1\}^*\}$$

wyr.reg.



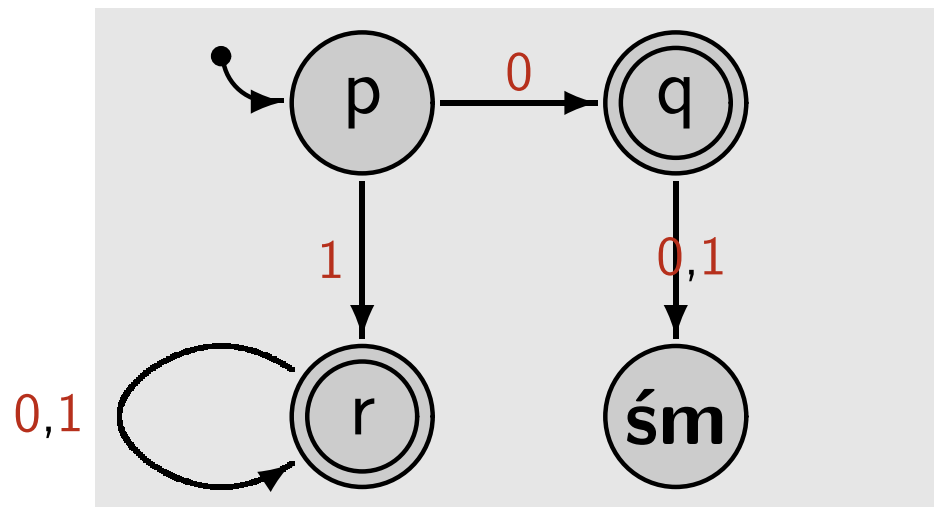
Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(\overbrace{0 \mid 1 \{0 \mid 1\}^*}) = \{0\} \cup \{1w \mid w \in \{0,1\}^*\}$$

wyr.reg.



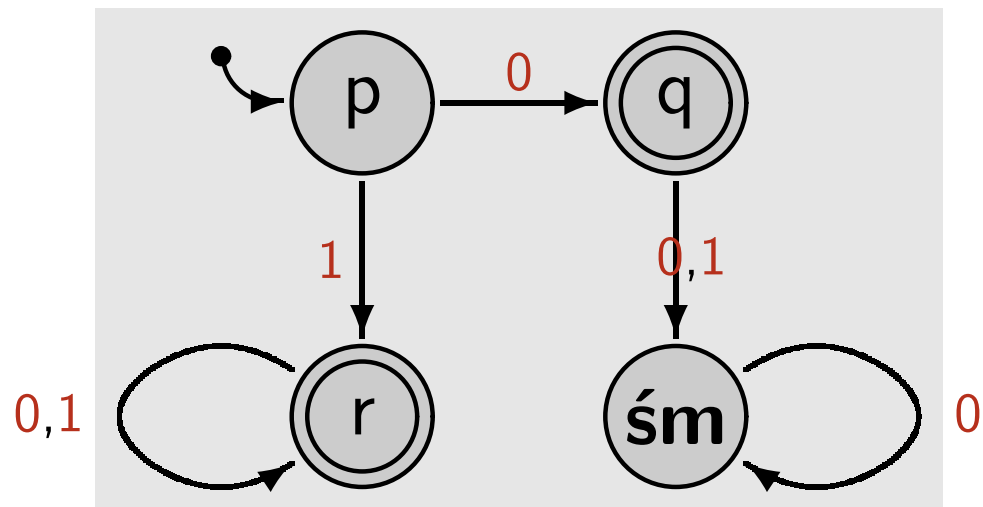
Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(\overbrace{0 \mid 1 \{0 \mid 1\}^*}) = \{0\} \cup \{1w \mid w \in \{0, 1\}^*\}$$

wyr.reg.



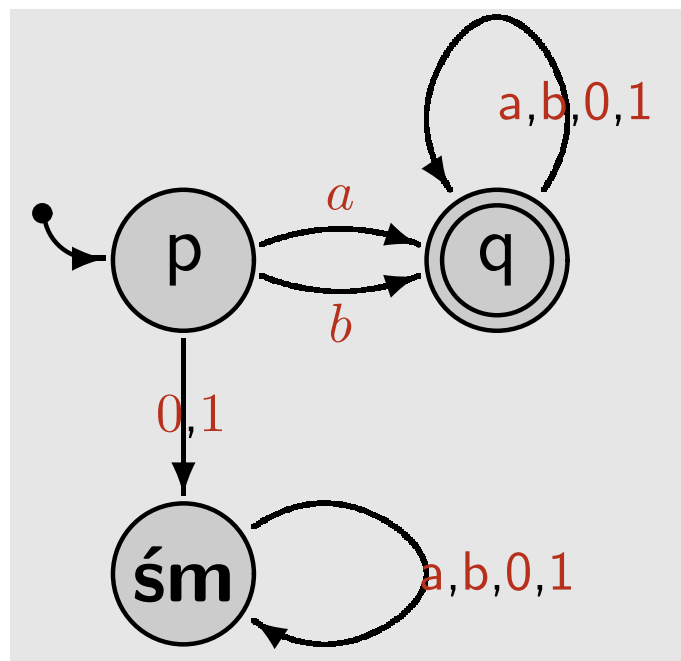
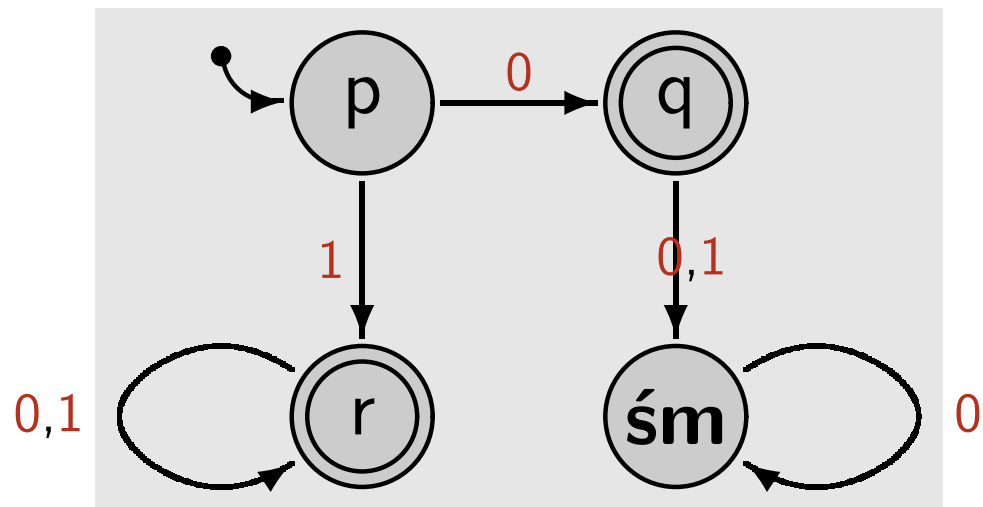
Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(\overbrace{0 \mid 1 \{0 \mid 1\}^*}) = \{0\} \cup \{1w \mid w \in \{0,1\}^*\}$$

wyr.reg.



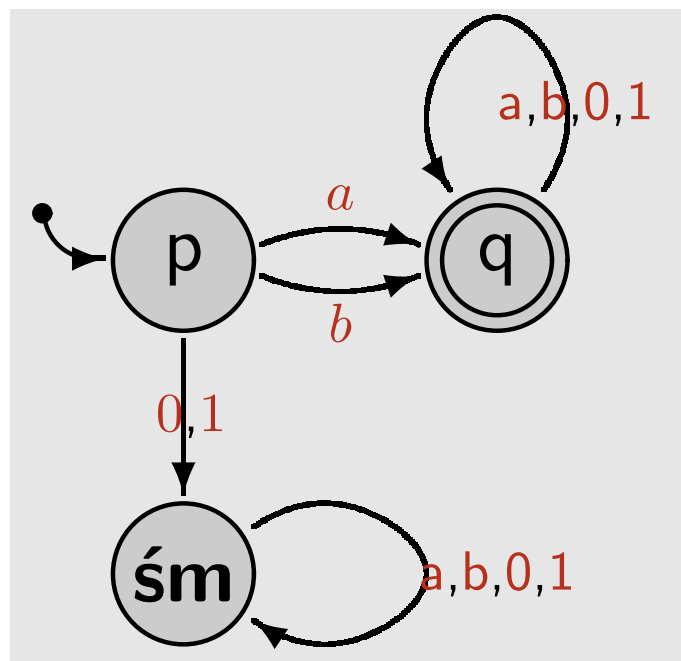
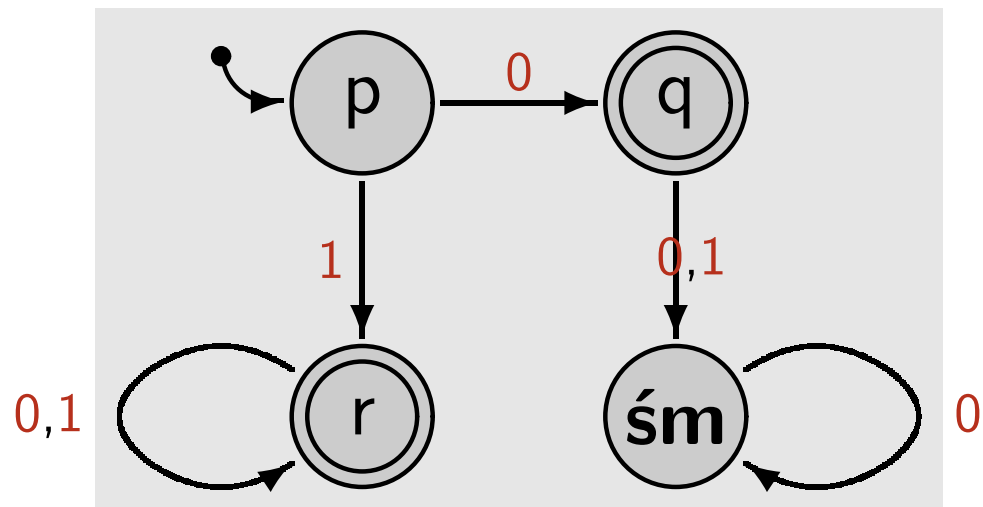
Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(\overbrace{0 \mid 1 \{0 \mid 1\}^*}) = \{0\} \cup \{1w \mid w \in \{0,1\}^*\}$$

wyr.reg.



PRZYKŁAD:

Język identyfikatorów

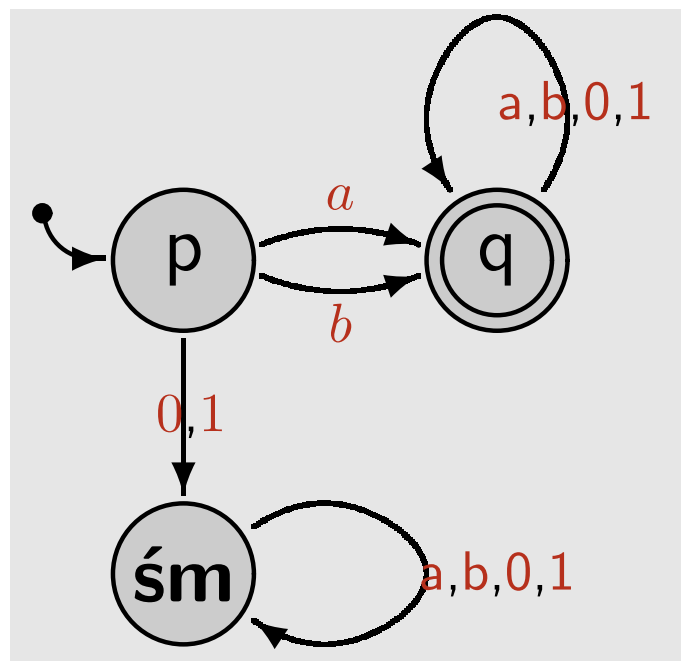
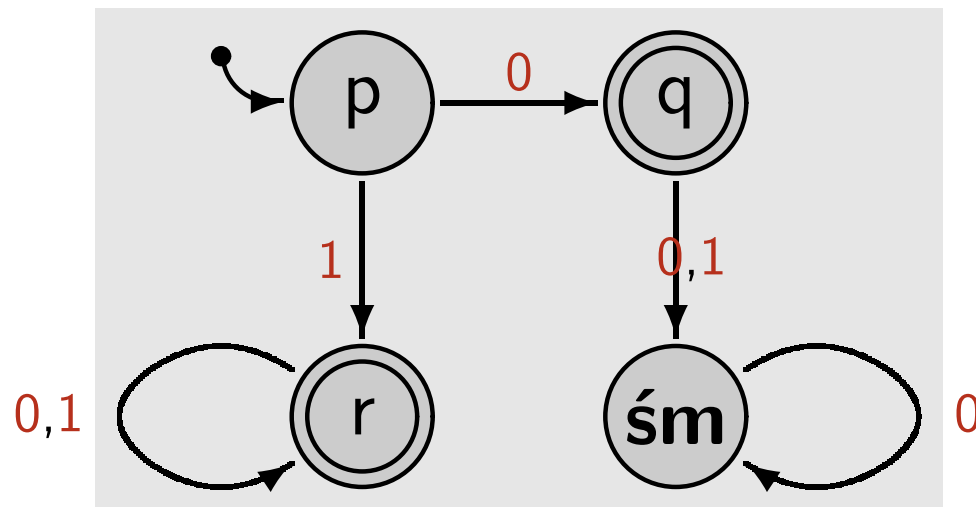
Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(\overbrace{0 \mid 1 \{0 \mid 1\}^*}) = \{0\} \cup \{1w \mid w \in \{0, 1\}^*\}$$

wyr.reg.



PRZYKŁAD:

Język identyfikatorów:

$$\{wv \mid w \in \{a, b\} \wedge v \in \{a, b, 0, 1\}^*\}$$

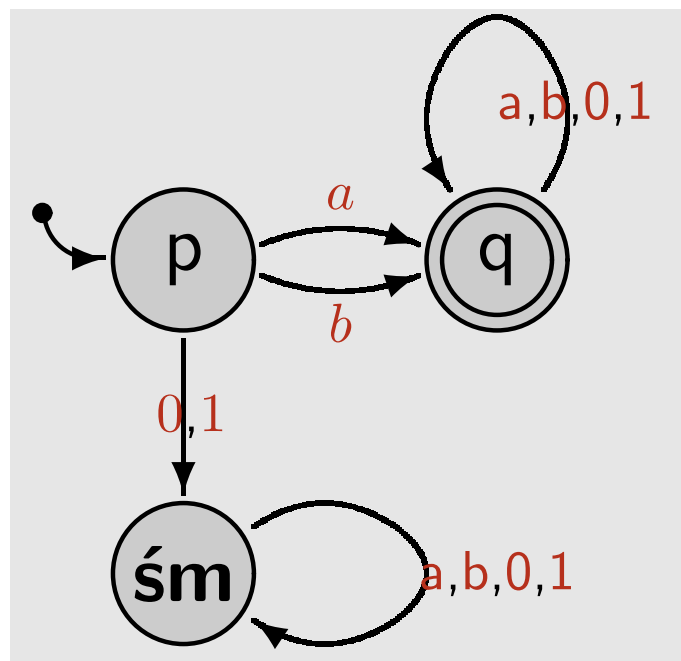
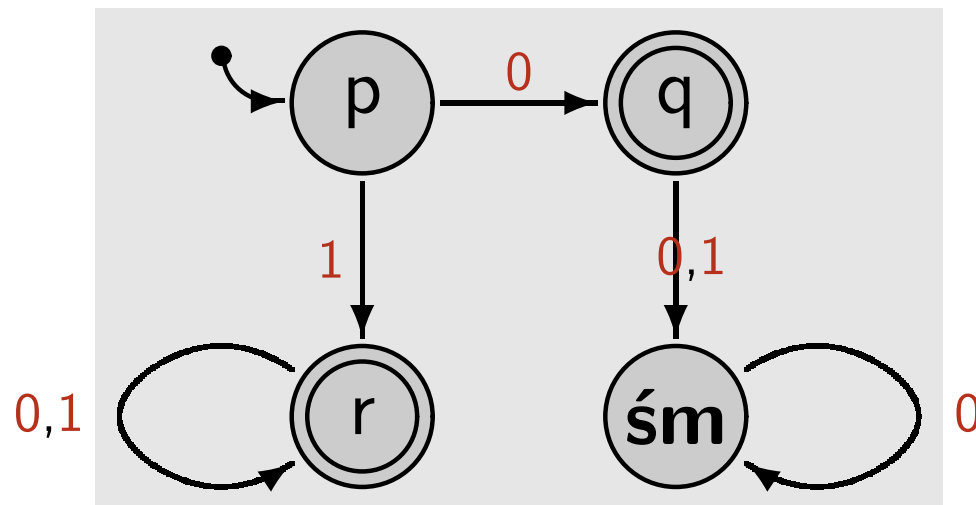
Wyrażenia regularne a automaty skończone

PRZYKŁAD:

Język liczb naturalnych:

$$L(\overbrace{0 \mid 1 \{0 \mid 1\}^*}) = \{0\} \cup \{1w \mid w \in \{0, 1\}^*\}$$

wyr.reg.



PRZYKŁAD:

Język identyfikatorów:

$$\{wv \mid w \in \{a, b\} \wedge v \in \{a, b, 0, 1\}^*\} = L(\underbrace{\{a \mid b\} \{a \mid b \mid 0 \mid 1\}^*}_{\text{wyr.reg.}})$$

Automaty skończone a gramatyki prawoliniowe

Automaty skończone a gramatyki prawoliniowe

TWIERDZENIE:

Dla każdego automatu skończonego M język $L(M)$ jest regularny.

Automaty skończone a gramatyki prawoliniowe

TWIERDZENIE:

*Dla każdego automatu skończonego M język $L(M)$ jest **regularny**.*

Dla każdego języka regularnego L istnieje automat skończony M taki że $L = L(M)$.

Automaty skończone a gramatyki prawoliniowe

TWIERDZENIE:

*Dla każdego automatu skończonego M język $L(M)$ jest **regularny**.*

Dla każdego języka regularnego L istnieje automat skończony M taki że $L = L(M)$.

PRZYKŁAD:

gramatyka prawoliniowa	wyrażenie regularne	automat skończony
$\langle S \rangle ::= \lambda$ $ \text{abb} \langle S \rangle$		

Automaty skończone a gramatyki prawoliniowe

TWIERDZENIE:

Dla każdego automatu skończonego M język $L(M)$ jest *regularny*.

Dla każdego języka regularnego L istnieje automat skończony M taki że $L = L(M)$.

PRZYKŁAD:

gramatyka prawoliniowa	wyrażenie regularne	automat skończony
$\langle S \rangle \begin{array}{l} ::= \lambda \\ \text{abb} \langle S \rangle \end{array}$	$\{abb\}^*$	

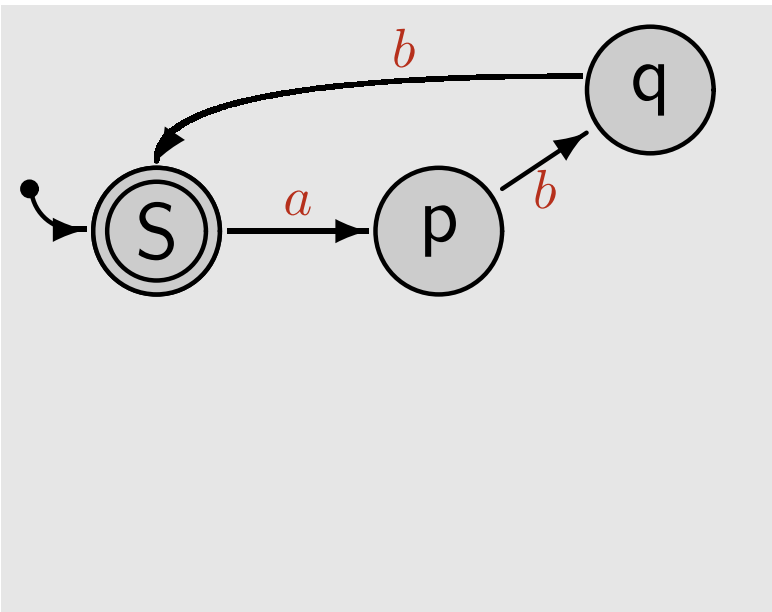
Automaty skończone a gramatyki prawoliniowe

TWIERDZENIE:

Dla każdego automatu skończonego M język $L(M)$ jest *regularny*.

Dla każdego języka regularnego L istnieje automat skończony M taki że $L = L(M)$.

PRZYKŁAD:

gramatyka prawoliniowa	wyrażenie regularne	automat skończony
$\langle S \rangle ::= \lambda$ $ abb \langle S \rangle$	$\{abb\}^*$	 <pre> graph LR start(()) --> S(((S))) S -- a --> p((p)) p -- b --> q((q)) q -- b --> S style start fill:none,stroke:none style S fill:none,stroke:none style p fill:none,stroke:none style q fill:none,stroke:none </pre>

Automaty skończone a gramatyki prawoliniowe

TWIERDZENIE:

Dla każdego automatu skończonego M język $L(M)$ jest *regularny*.

Dla każdego języka regularnego L istnieje automat skończony M taki że $L = L(M)$.

PRZYKŁAD:

gramatyka prawoliniowa	wyrażenie regularne	automat skończony
$\langle S \rangle ::= \lambda$ $ abb \langle S \rangle$	$\{abb\}^*$	<pre> graph LR S((S)) -- a --> p((p)) p -- b --> q((q)) q -- b --> S q -- a --> sm((śm)) p -- a --> sm sm -- "a,b" --> sm style S fill:none,stroke:#000,stroke-width:2px style sm fill:none,stroke:#000,stroke-width:2px </pre>

Automaty skończone a gramatyki prawoliniowe

PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a \langle A \rangle \mid b \langle B \rangle$$

$$\langle A \rangle ::= a \mid b \langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa \langle A \rangle$$

Automaty skończone a gramatyki prawoliniowe

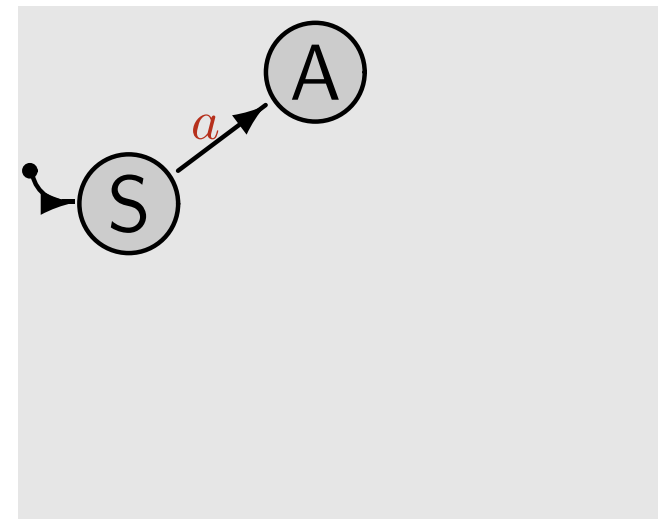
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a \langle A \rangle \mid b \langle B \rangle$$

$$\langle A \rangle ::= a \mid b \langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa \langle A \rangle$$



Automaty skończone a gramatyki prawoliniowe

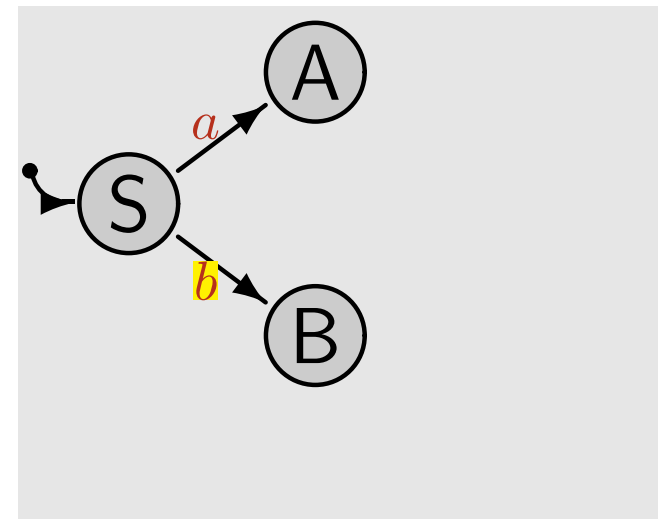
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a \langle A \rangle \mid b \langle B \rangle$$

$$\langle A \rangle ::= a \mid b \langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa \langle A \rangle$$



Automaty skończone a gramatyki prawoliniowe

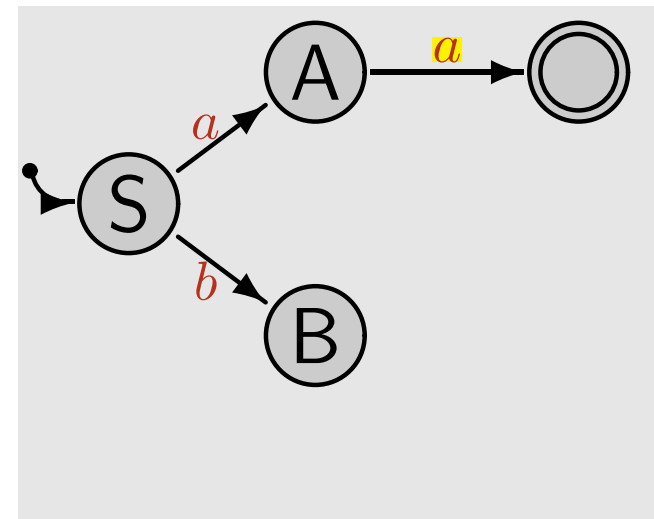
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a \langle A \rangle \mid b \langle B \rangle$$

$$\langle A \rangle ::= a \mid b \langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa \langle A \rangle$$



Automaty skończone a gramatyki prawoliniowe

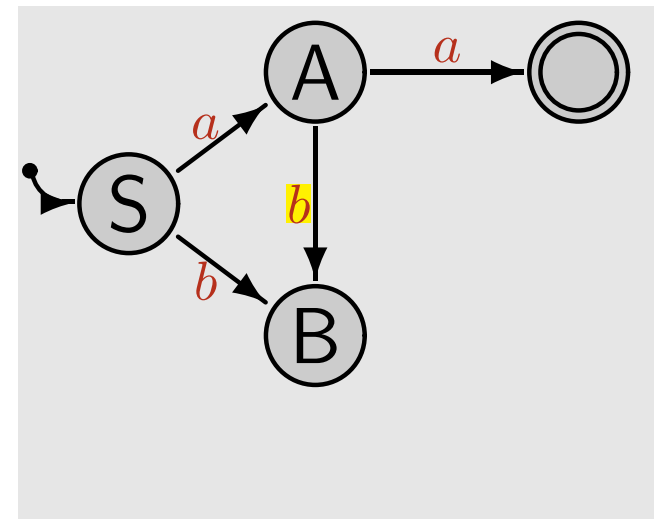
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a \langle A \rangle \mid b \langle B \rangle$$

$$\langle A \rangle ::= a \mid b \langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa \langle A \rangle$$



Automaty skończone a gramatyki prawoliniowe

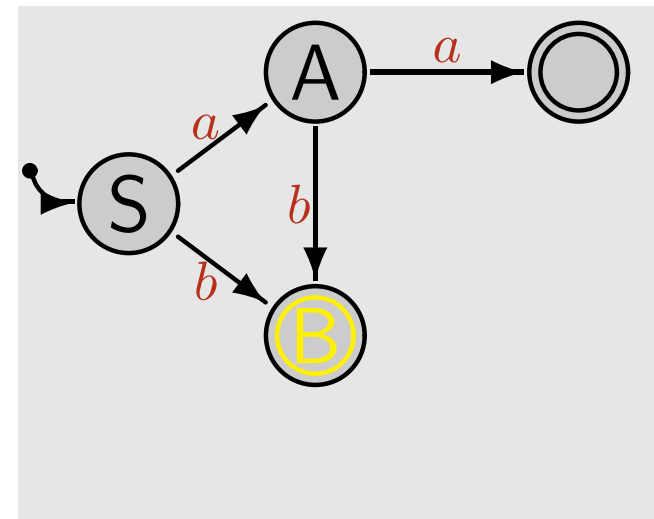
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a\langle A \rangle \mid b\langle B \rangle$$

$$\langle A \rangle ::= a \mid b\langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa\langle A \rangle$$



Automaty skończone a gramatyki prawoliniowe

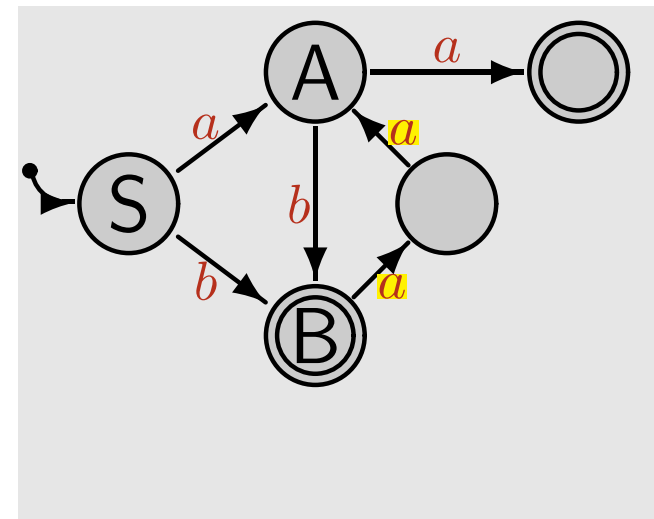
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a\langle A \rangle \mid b\langle B \rangle$$

$$\langle A \rangle ::= a \mid b\langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa\langle A \rangle$$



Automaty skończone a gramatyki prawoliniowe

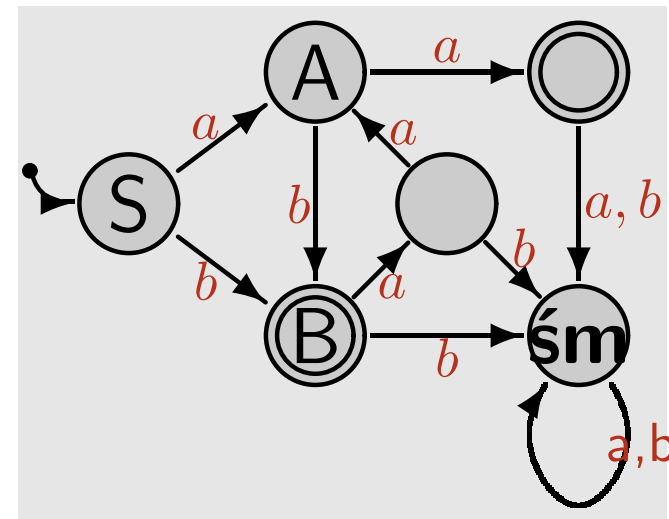
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a\langle A \rangle \mid b\langle B \rangle$$

$$\langle A \rangle ::= a \mid b\langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa\langle A \rangle$$



Automaty skończone a gramatyki prawoliniowe

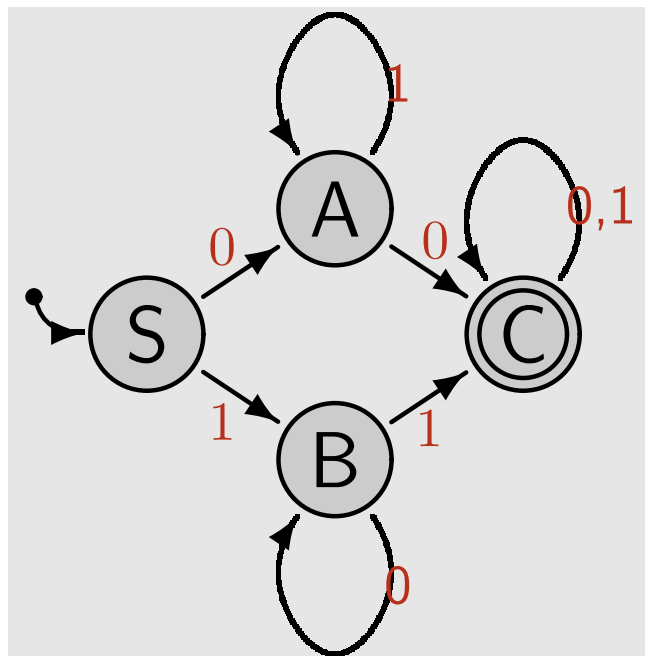
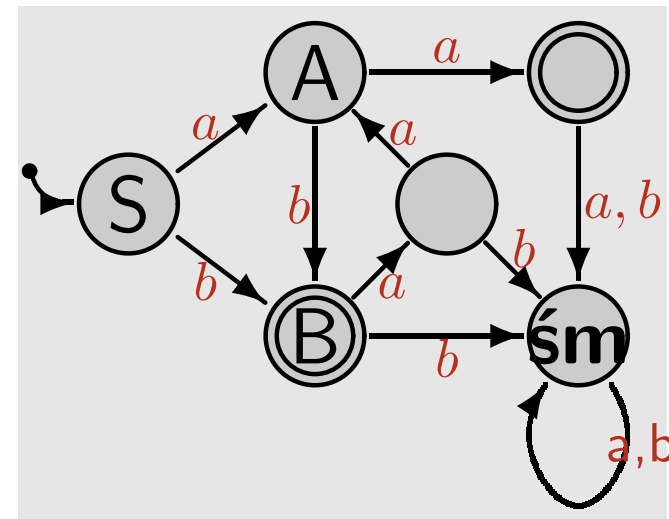
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a\langle A \rangle \mid b\langle B \rangle$$

$$\langle A \rangle ::= a \mid b\langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa\langle A \rangle$$



Automaty skończone a gramatyki prawoliniowe

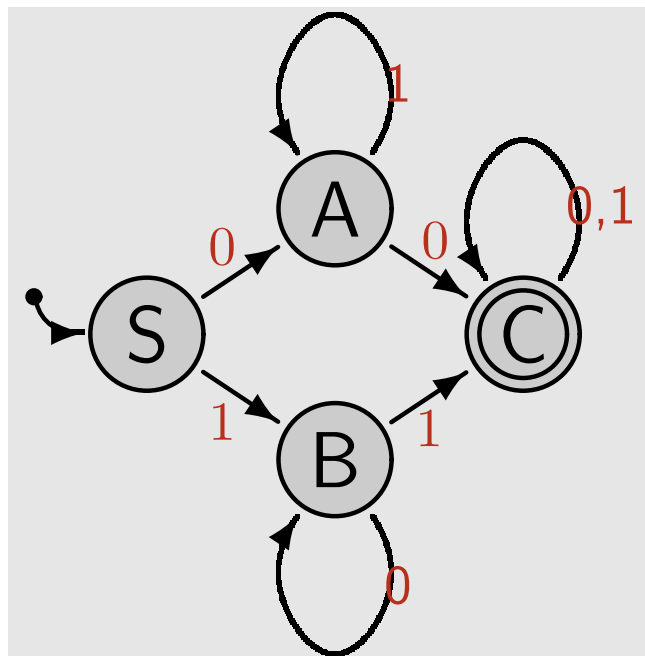
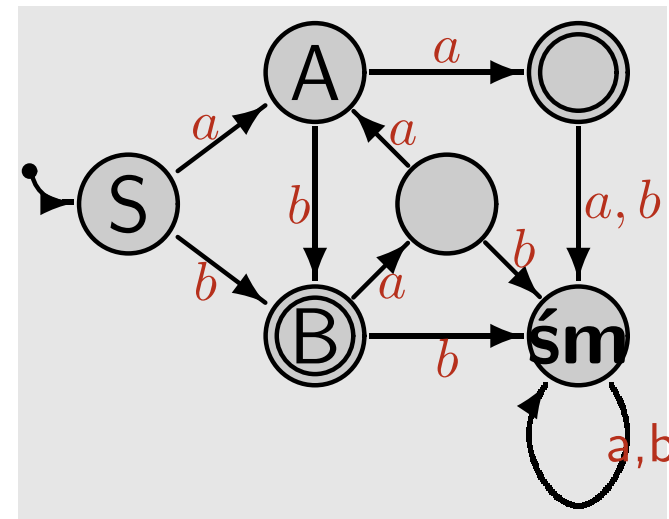
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a\langle A \rangle \mid b\langle B \rangle$$

$$\langle A \rangle ::= a \mid b\langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa\langle A \rangle$$



PRZYKŁAD:

Gramatyka prawoliniowa

Automaty skończone a gramatyki prawoliniowe

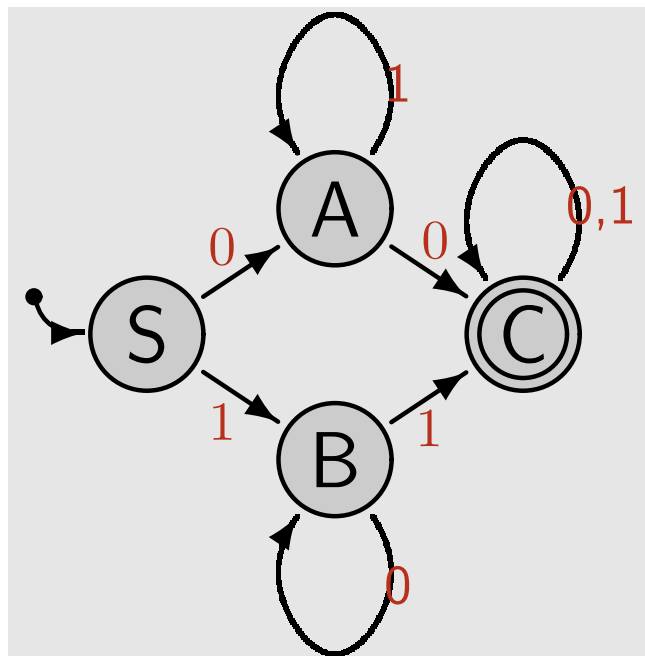
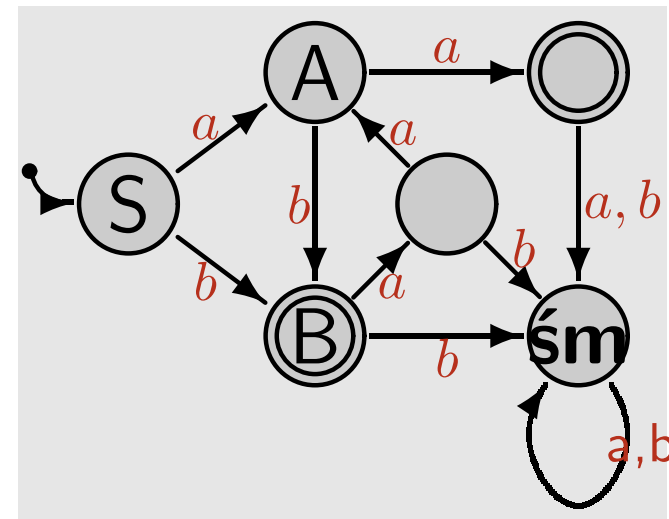
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a\langle A \rangle \mid b\langle B \rangle$$

$$\langle A \rangle ::= a \mid b\langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa\langle A \rangle$$



PRZYKŁAD:

Gramatyka prawoliniowa:

$$\langle S \rangle ::= 0\langle A \rangle \mid 1\langle B \rangle$$

Automaty skończone a gramatyki prawoliniowe

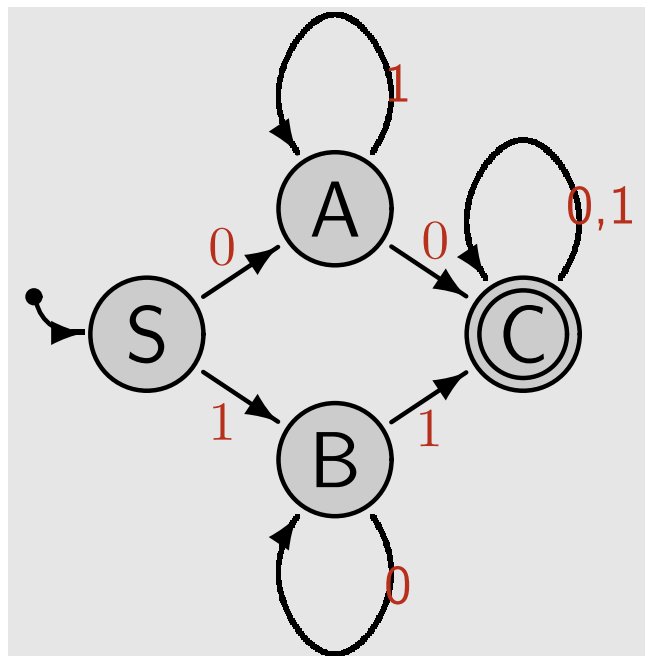
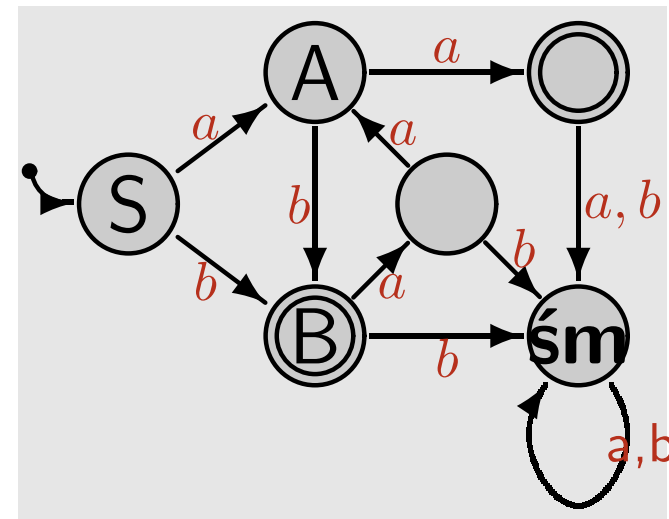
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a\langle A \rangle \mid b\langle B \rangle$$

$$\langle A \rangle ::= a \mid b\langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa\langle A \rangle$$



PRZYKŁAD:

Gramatyka prawoliniowa:

$$\langle S \rangle ::= 0\langle A \rangle \mid 1\langle B \rangle$$

$$\langle A \rangle ::= 0\langle C \rangle \mid 1\langle A \rangle$$

Automaty skończone a gramatyki prawoliniowe

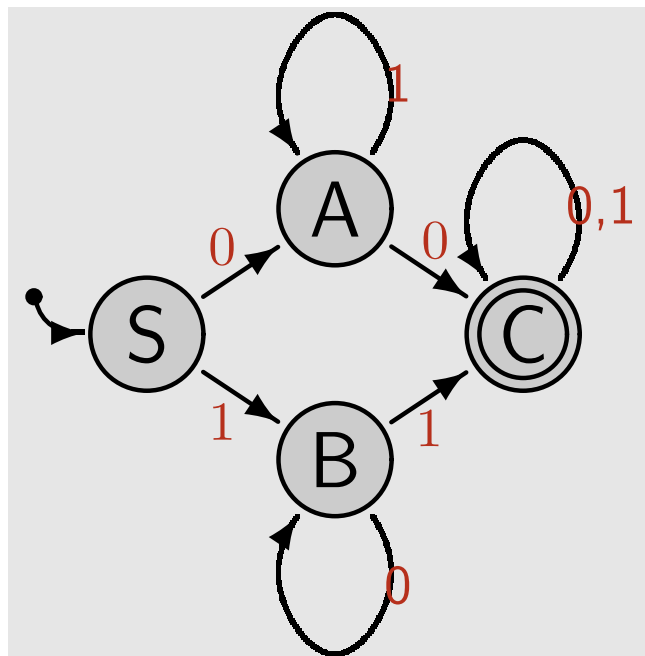
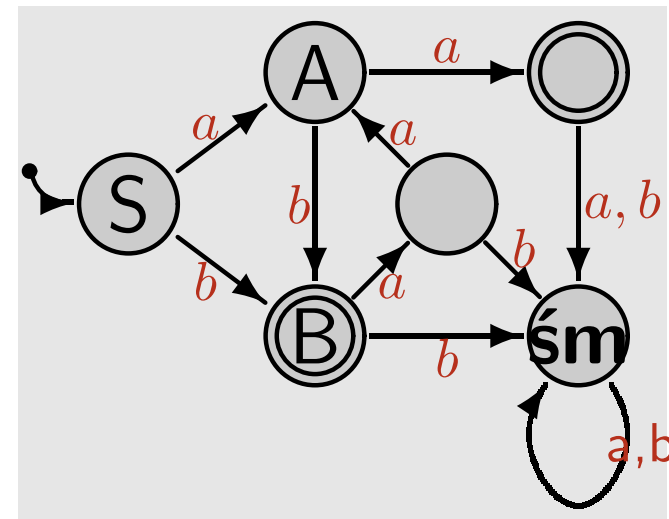
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a\langle A \rangle \mid b\langle B \rangle$$

$$\langle A \rangle ::= a \mid b\langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa\langle A \rangle$$



PRZYKŁAD:

Gramatyka prawoliniowa:

$$\langle S \rangle ::= 0\langle A \rangle \mid 1\langle B \rangle$$

$$\langle A \rangle ::= 0\langle C \rangle \mid 1\langle A \rangle$$

$$\langle B \rangle ::= 0\langle B \rangle \mid 1\langle C \rangle$$

Automaty skończone a gramatyki prawoliniowe

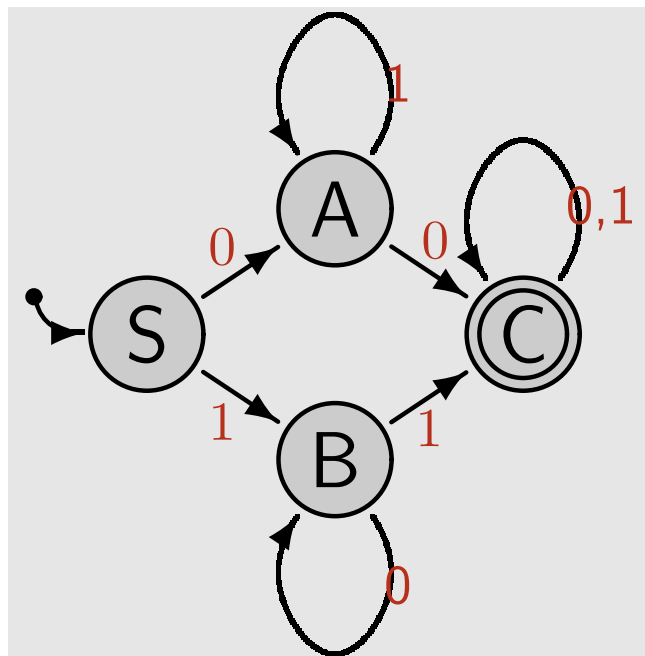
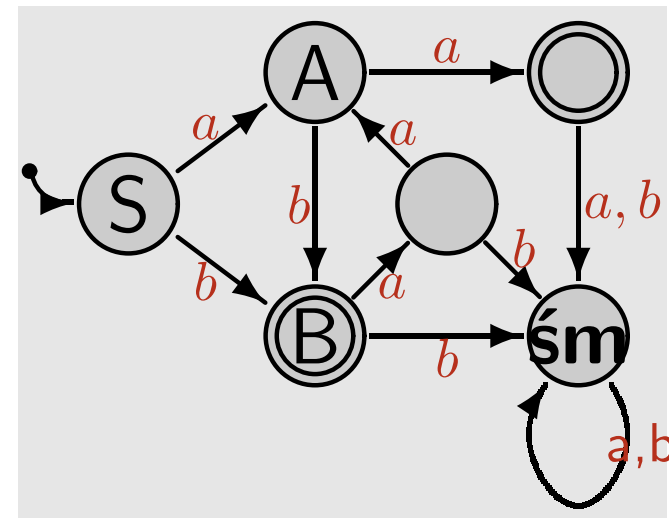
PRZYKŁAD:

Gramatyka prawoliniowa G :

$$\langle S \rangle ::= a\langle A \rangle \mid b\langle B \rangle$$

$$\langle A \rangle ::= a \mid b\langle B \rangle$$

$$\langle B \rangle ::= \lambda \mid aa\langle A \rangle$$



PRZYKŁAD:

Gramatyka prawoliniowa:

$$\langle S \rangle ::= 0\langle A \rangle \mid 1\langle B \rangle$$

$$\langle A \rangle ::= 0\langle C \rangle \mid 1\langle A \rangle$$

$$\langle B \rangle ::= 0\langle B \rangle \mid 1\langle C \rangle$$

$$\langle C \rangle ::= \lambda \mid 0\langle C \rangle \mid 1\langle C \rangle$$

Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

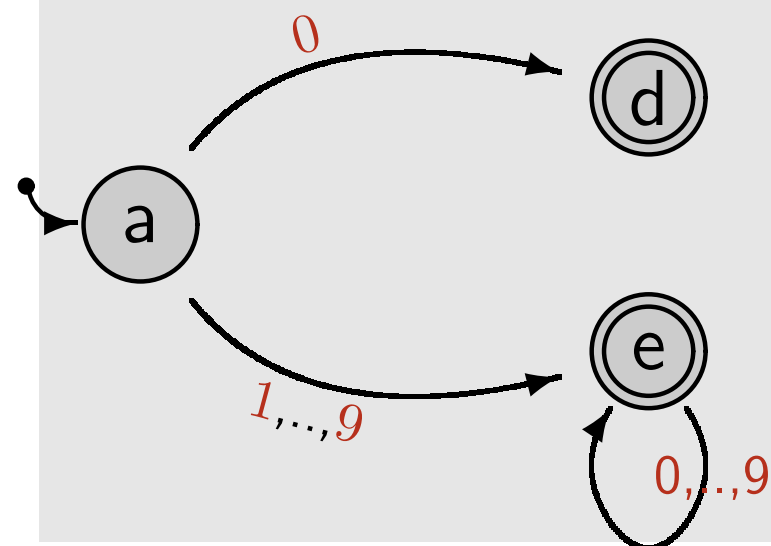
1234 − 0.001 6.02*E* − 23

Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

1234 − 0.001 6.02*E* − 23

CZĘŚĆ
CAŁKOWITA

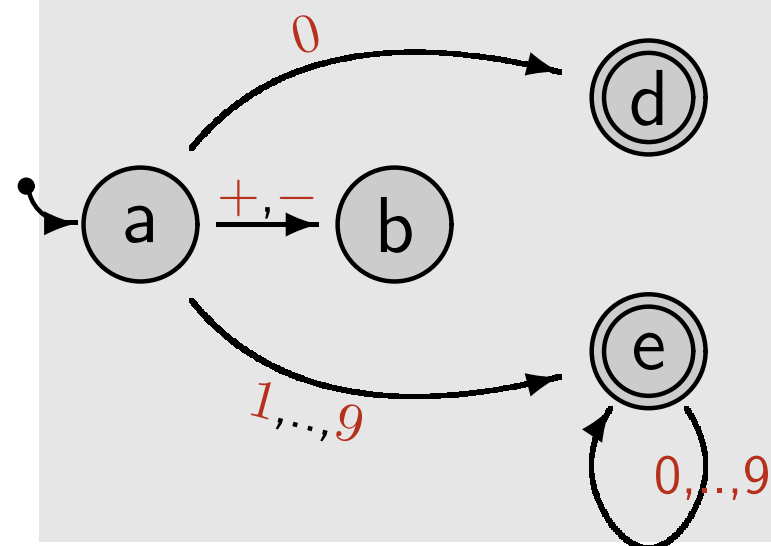


Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

1234 − 0.001 6.02*E* − 23

CZĘŚĆ
CAŁKOWITA

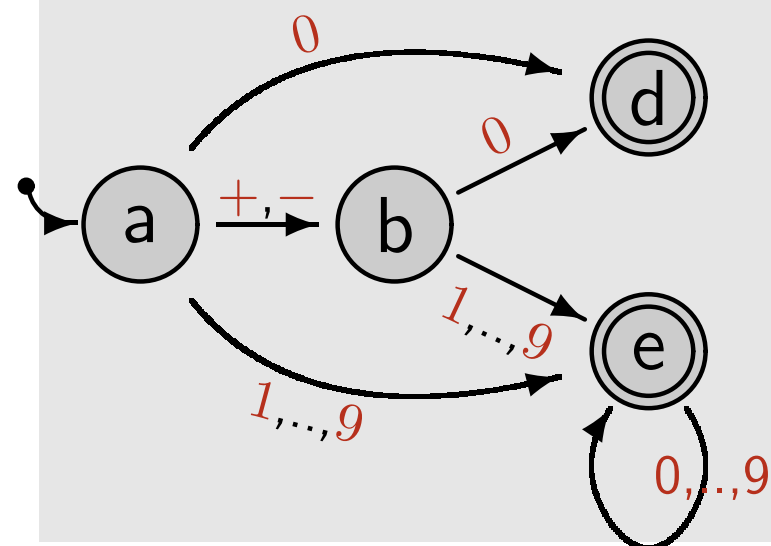


Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

1234 − 0.001 6.02E − 23

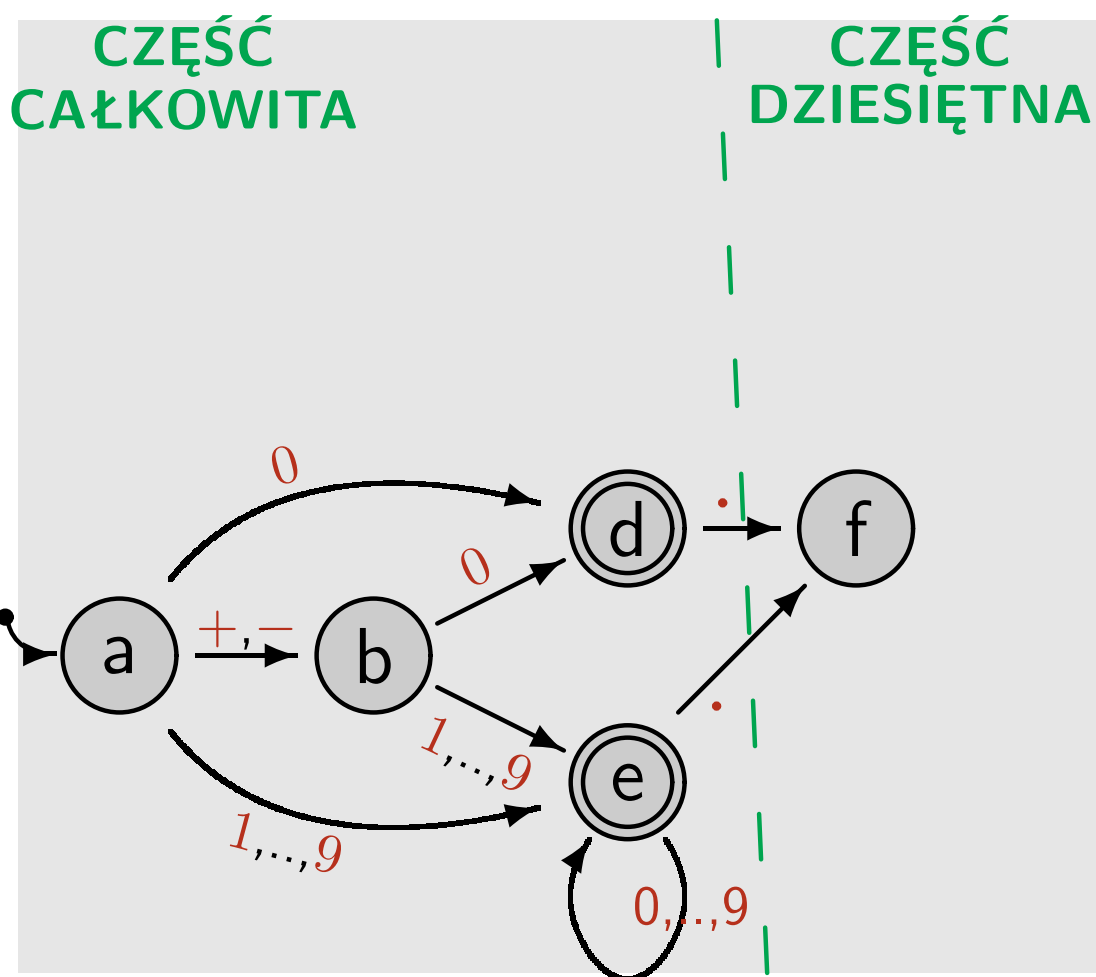
CZĘŚĆ
CAŁKOWITA



Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

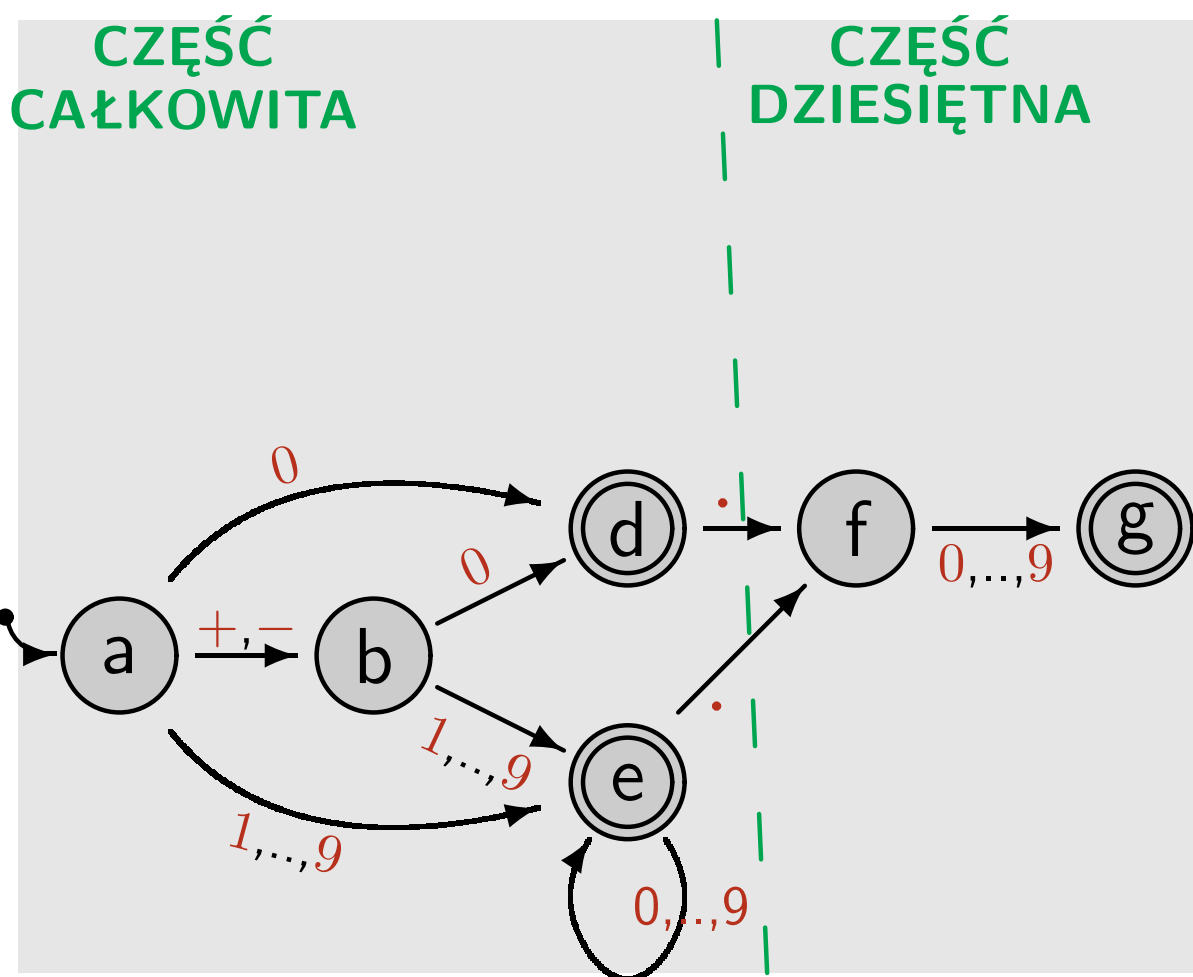
1234 − 0.001 6.02E − 23



Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

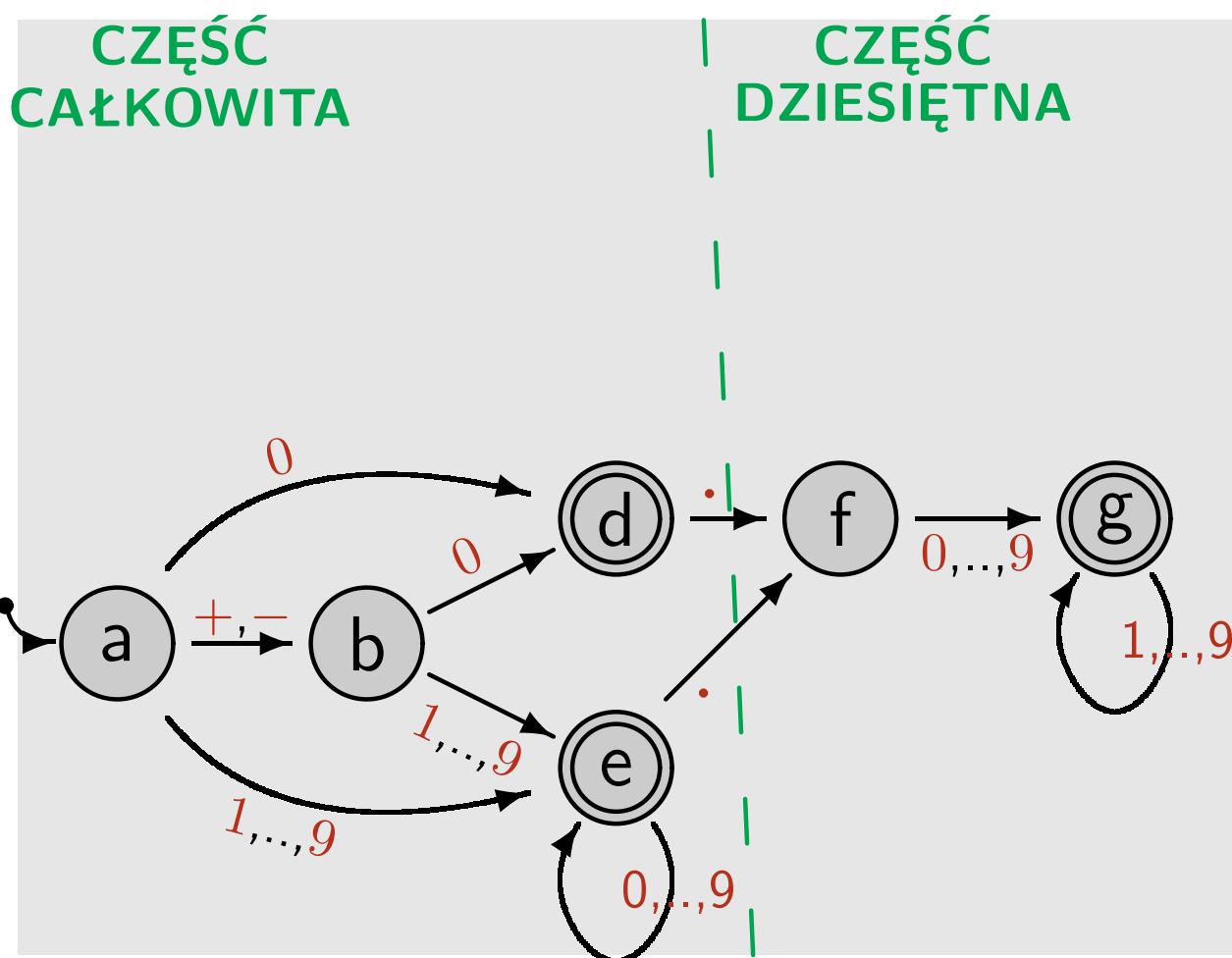
1234 − 0.001 6.02E − 23



Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

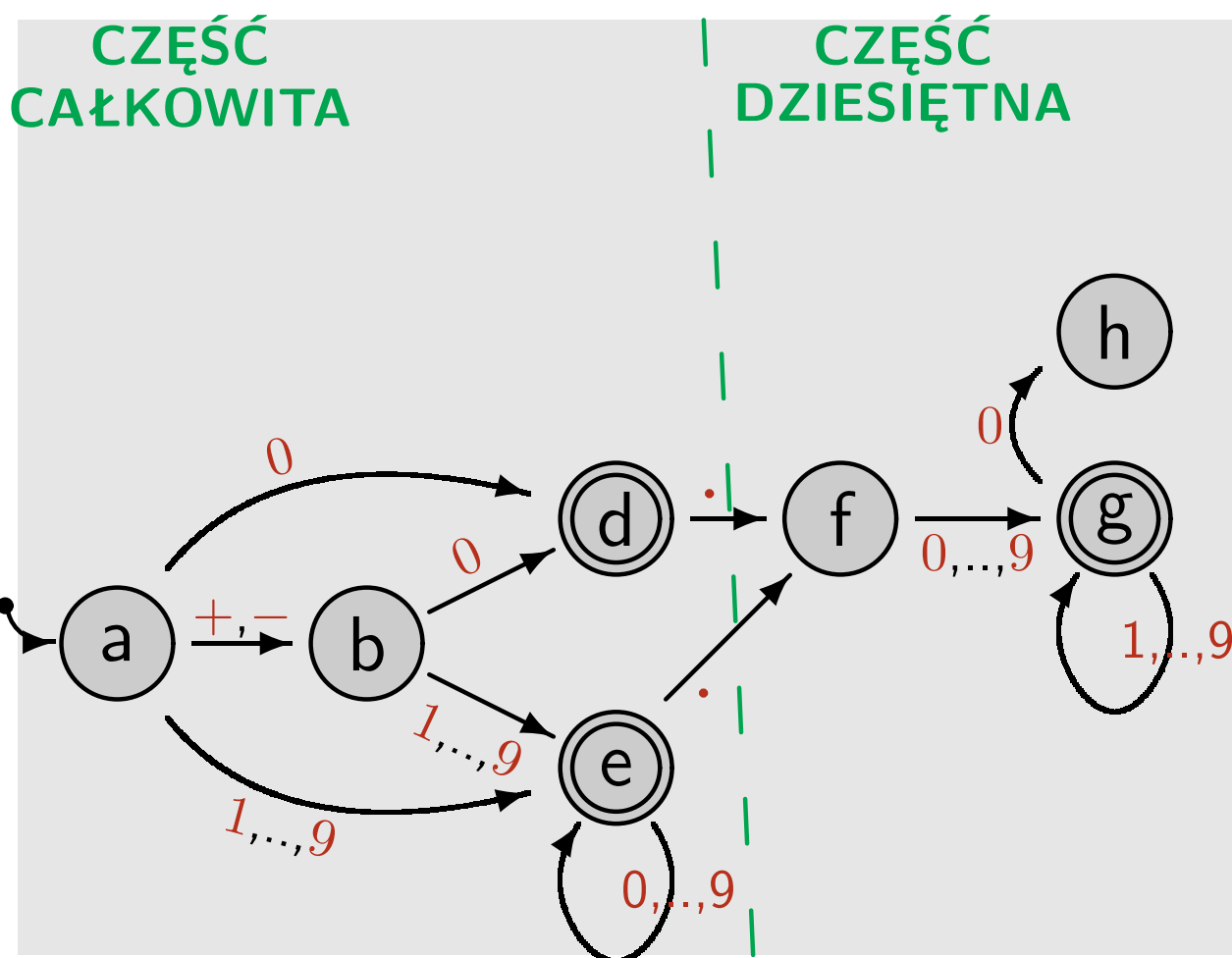
1234 − 0.001 6.02E − 23



Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

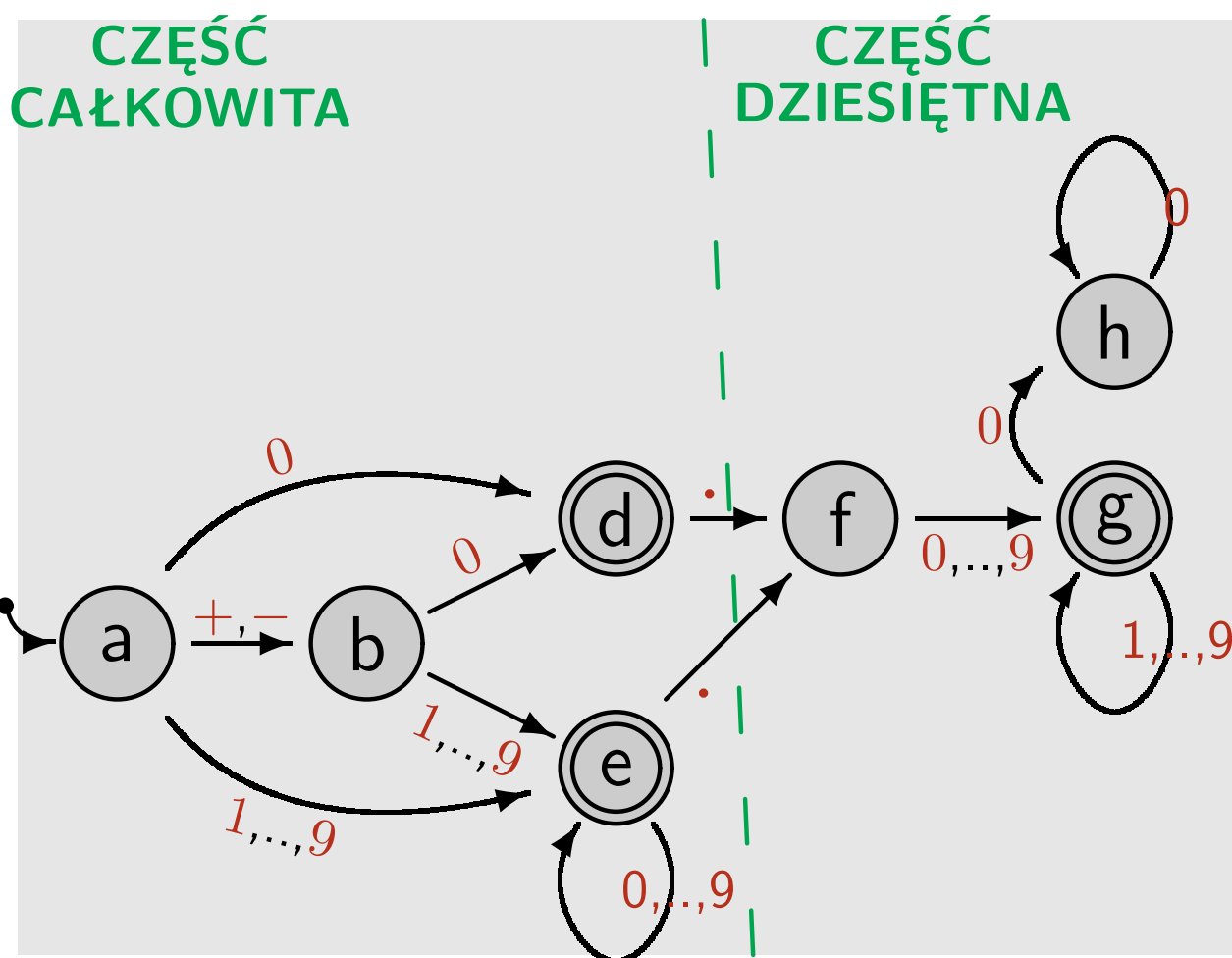
1234 − 0.001 6.02E − 23



Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

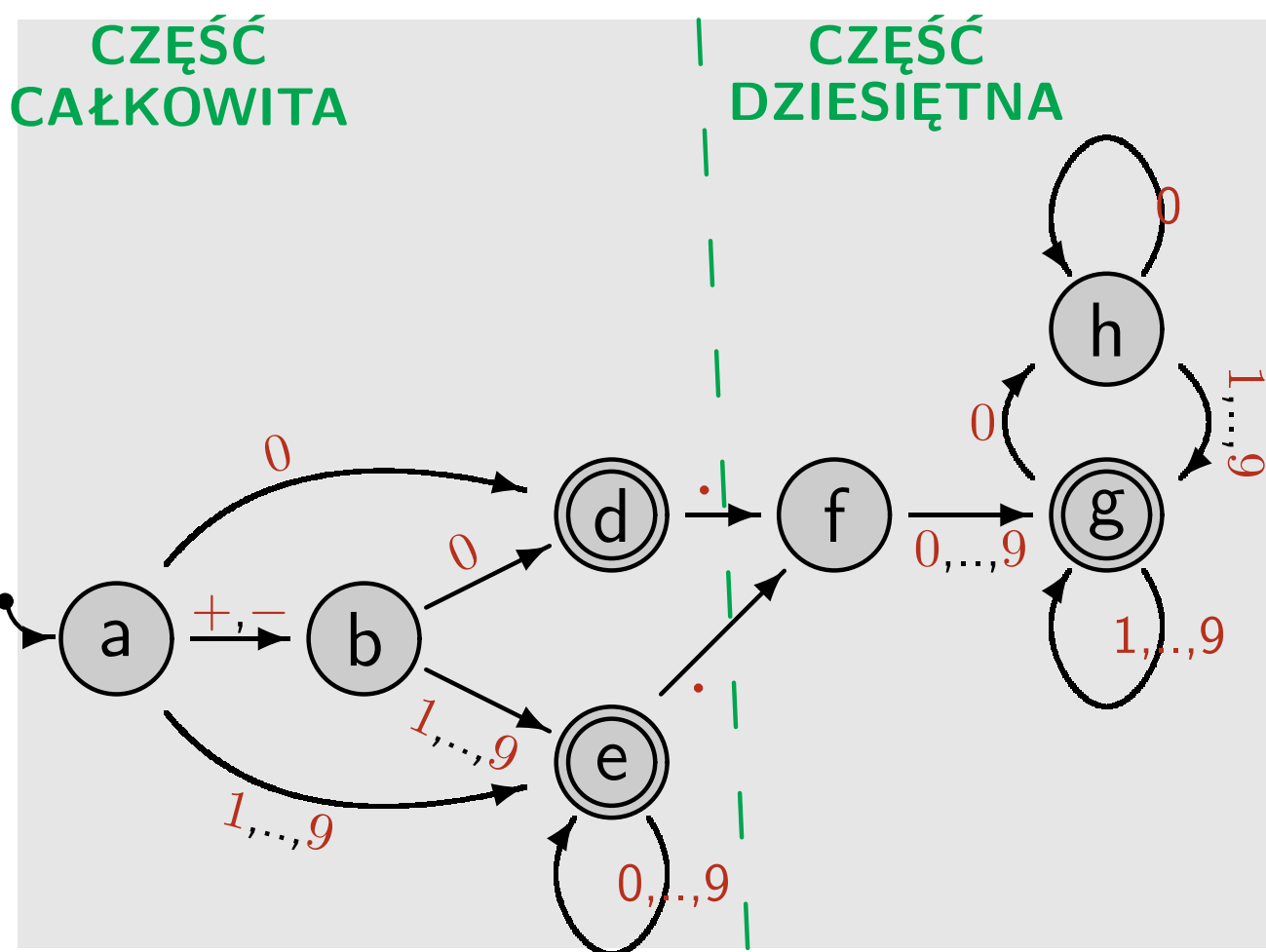
1234 − 0.001 6.02E − 23



Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

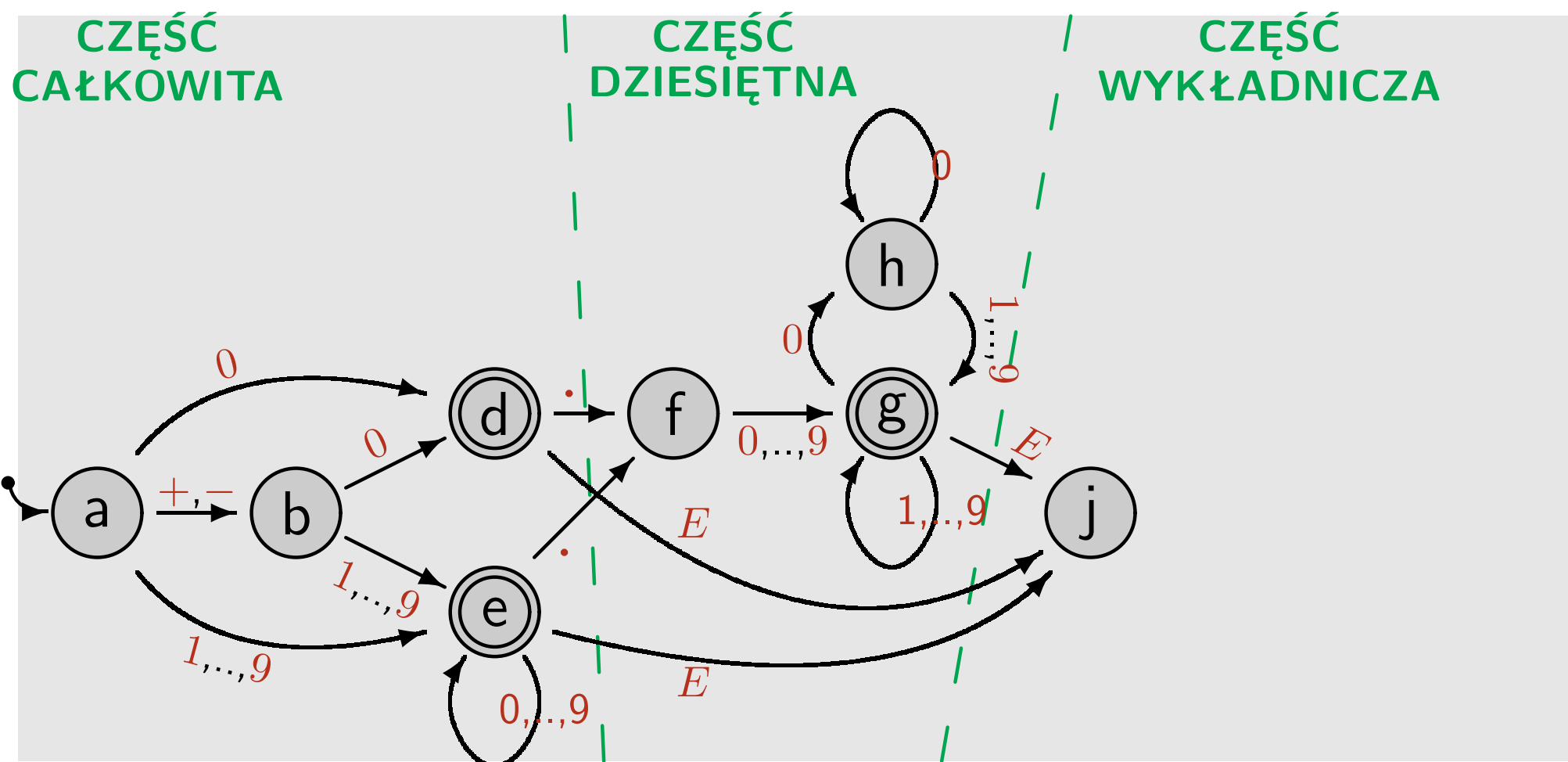
1234 − 0.001 6.02E − 23



Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

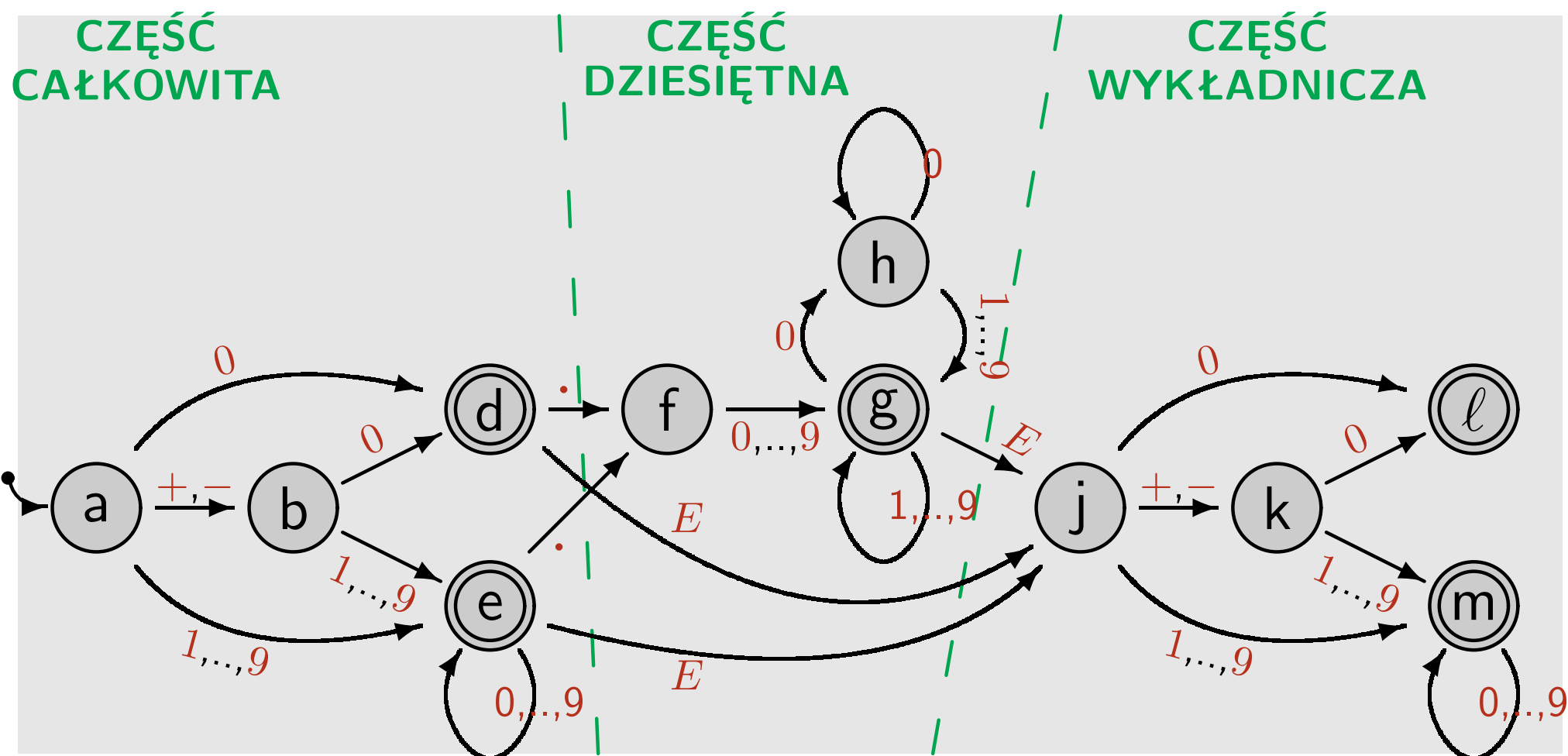
1234 − 0.001 6.02E − 23



Automaty skończone

Automat akceptujący liczby rzeczywiste takie jak na przykład:

1234 − 0.001 6.02E − 23



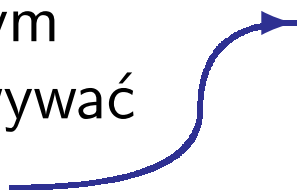
Maszyna ze stosem

Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery

Maszyna ze stosem

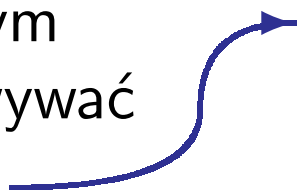
— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady



L
I
F
O

Maszyna ze stosem

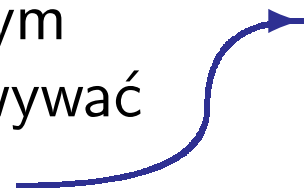
— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady



Last
In
First
Out

Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady



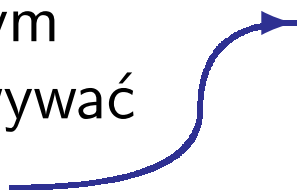
Last
In
First
Out

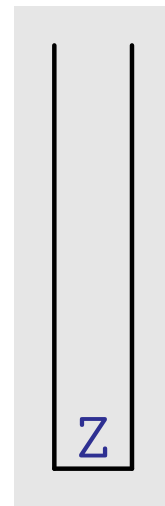
=

{ kto ostatni
przyszedł,
pierwszy
wychodzi }

Maszyna ze stosem

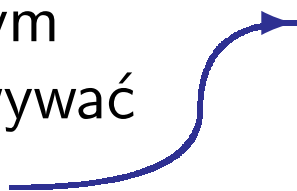
— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady

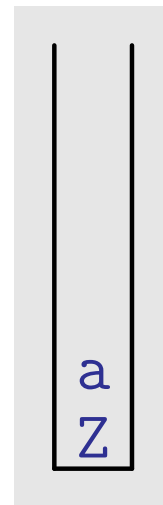
 $\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$



Maszyna ze stosem

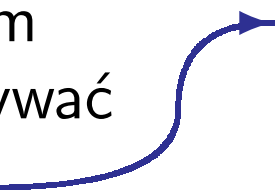
— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady

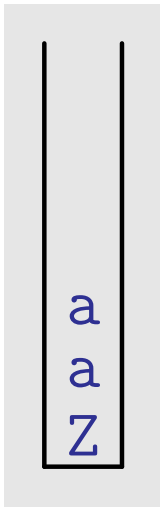

$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$



Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady

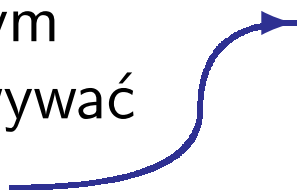

$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$

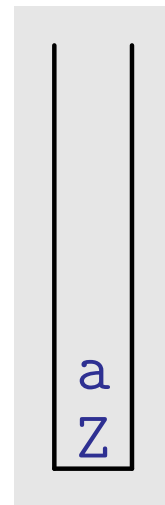


a
a
Z

Maszyna ze stosem

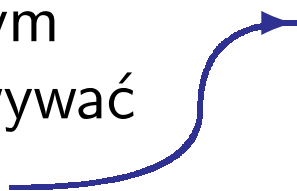
— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady


$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$



Maszyna ze stosem

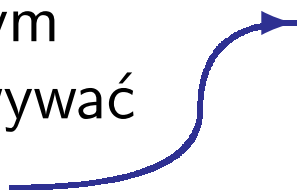
— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady


$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$

b
a
z

Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady



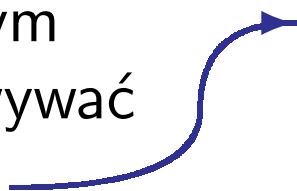
A blue curved arrow points from the text 'wg zasady' to the LIFO stack representation.

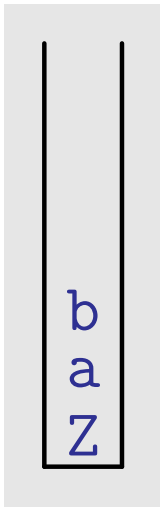
$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$

a
b
a
Z

Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady

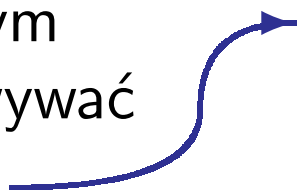

$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$



b
a
z

Maszyna ze stosem

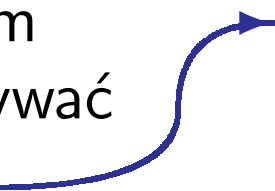
— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady

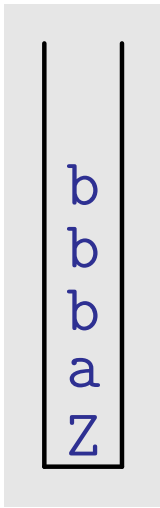

$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$

b
b
a
z

Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady


$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$



b
b
b
a
Z

Maszyna ze stosem

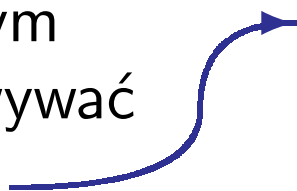
— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady

$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$

b
b
a
z

Maszyna ze stosem

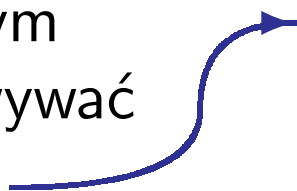
— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady


$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$

a
b
b
a
Z

Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady



A blue curved arrow points from the text 'wg zasady' to the LIFO stack representation.

$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$

b
a
b
b
a
Z

Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady

$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$

b
a
b
b
a
Z

Strzałki mają postać:

litera , litera $\longrightarrow$ słowo

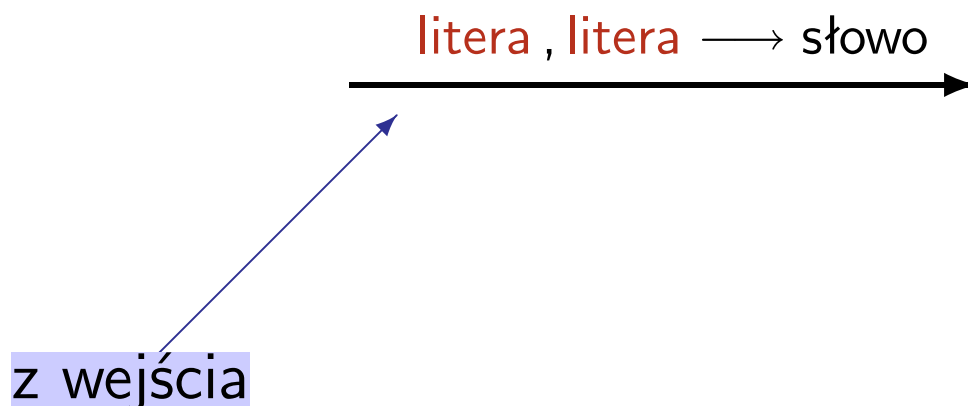
Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady

$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$

b
a
b
b
a
Z

Strzałki mają postać:



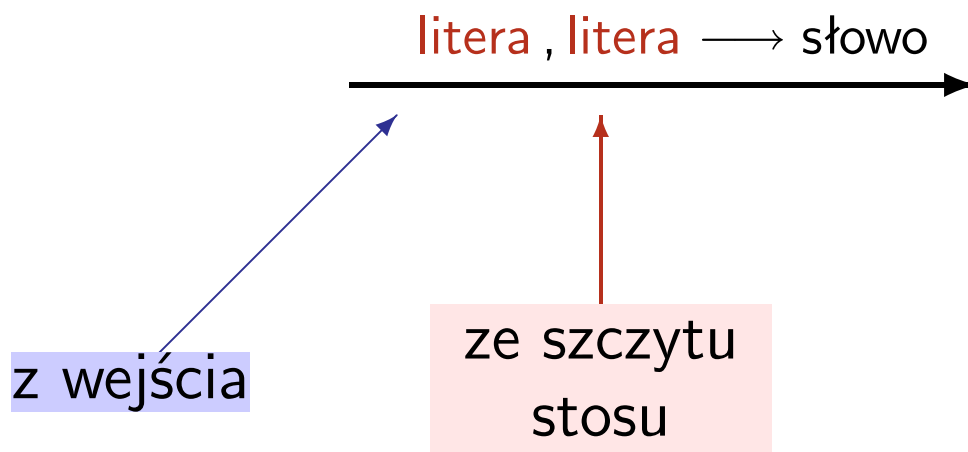
Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady

$$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$$

b
a
b
b
a
z

Strzałki mają postać:



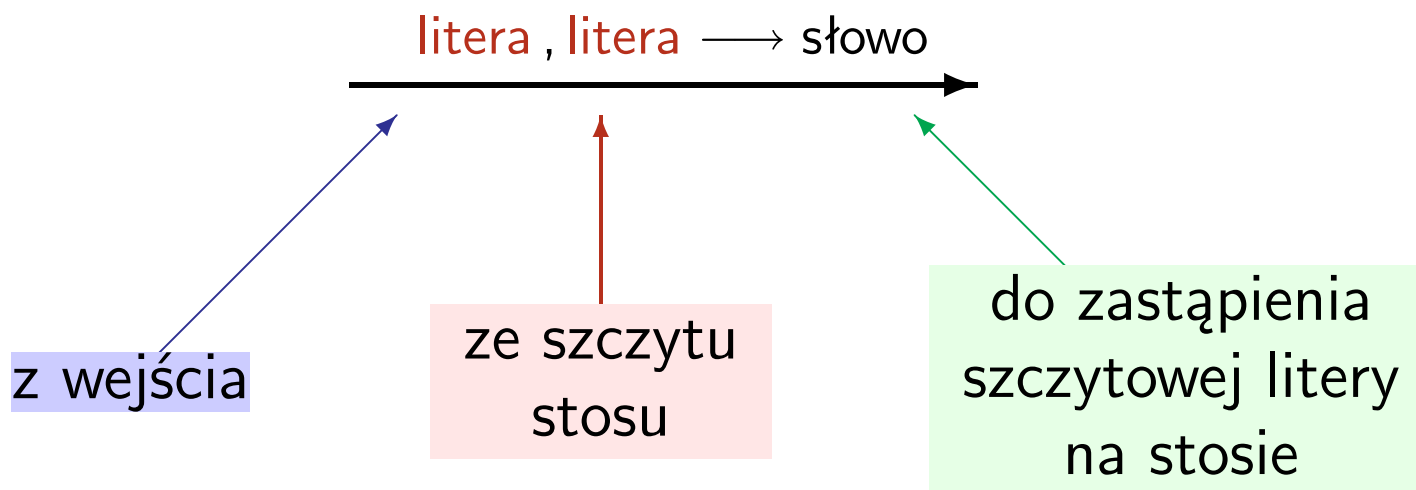
Maszyna ze stosem

— jakby automat, ale
z **nieskończonym**
stosem, na którym
można przechowywać
litery wg zasady

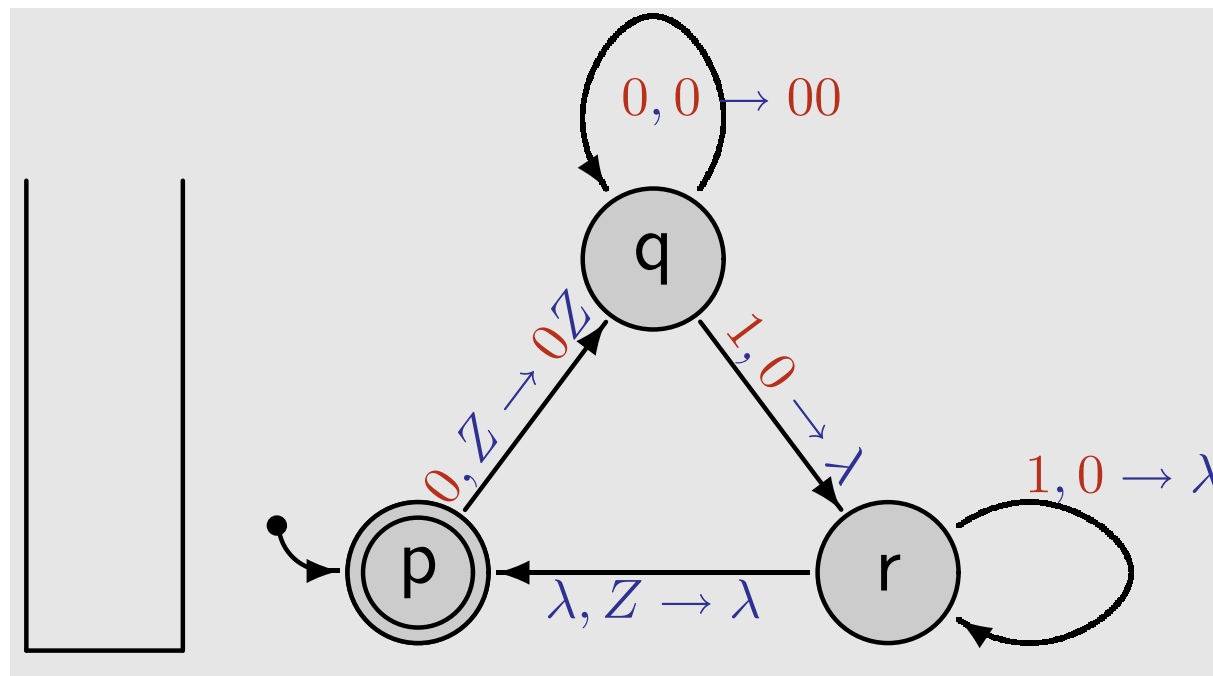
$\left\{ \begin{array}{c} \text{Last} \\ \text{In} \\ \text{First} \\ \text{Out} \end{array} \right\} = \left\{ \begin{array}{c} \text{kto ostatni} \\ \text{przyszedł,} \\ \text{pierwszy} \\ \text{wychodzi} \end{array} \right\}$

b
a
b
b
a
Z

Strzałki mają postać:



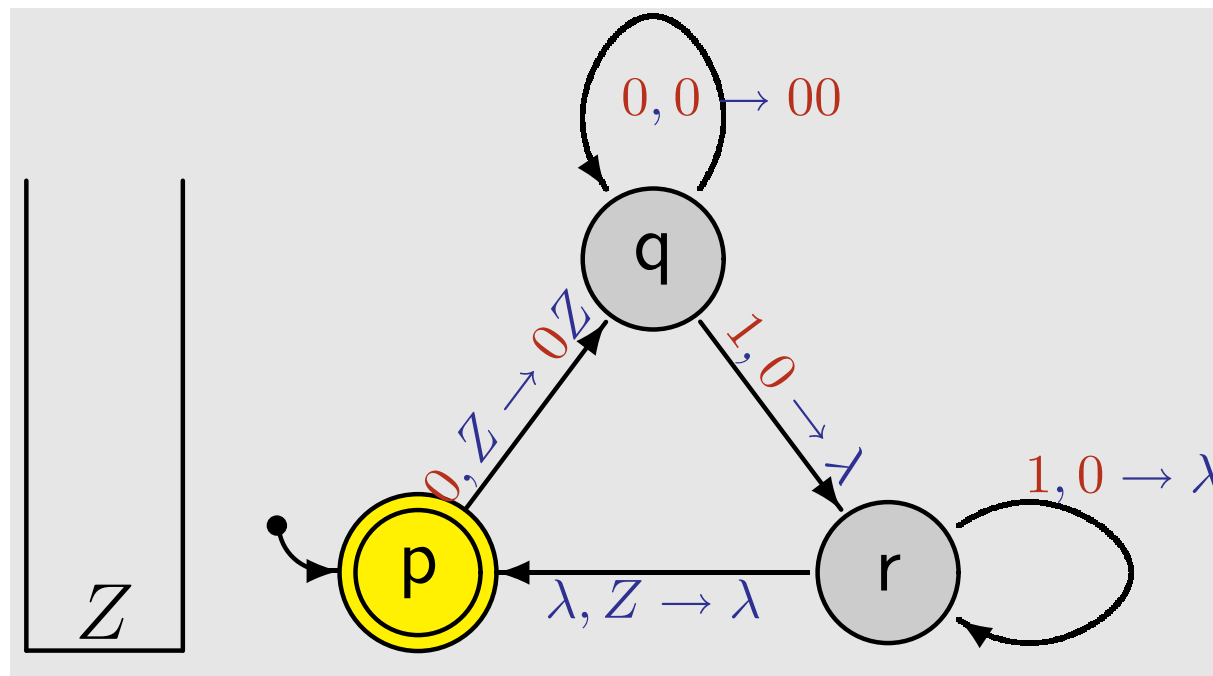
Maszyna ze stosem



Maszyna ze stosem

Słowo na wejściu:

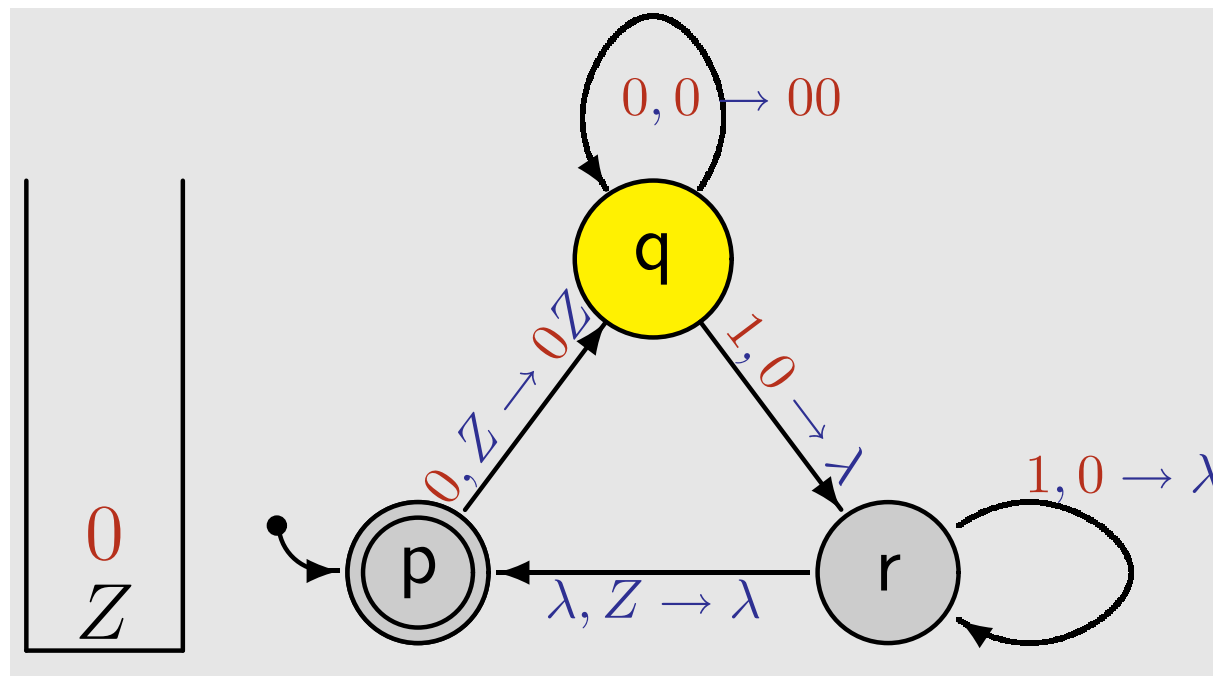
$0 \ 0 \ 1 \ 1$
 część
 nieprzeczytana



Maszyna ze stosem

Słowo na wejściu:

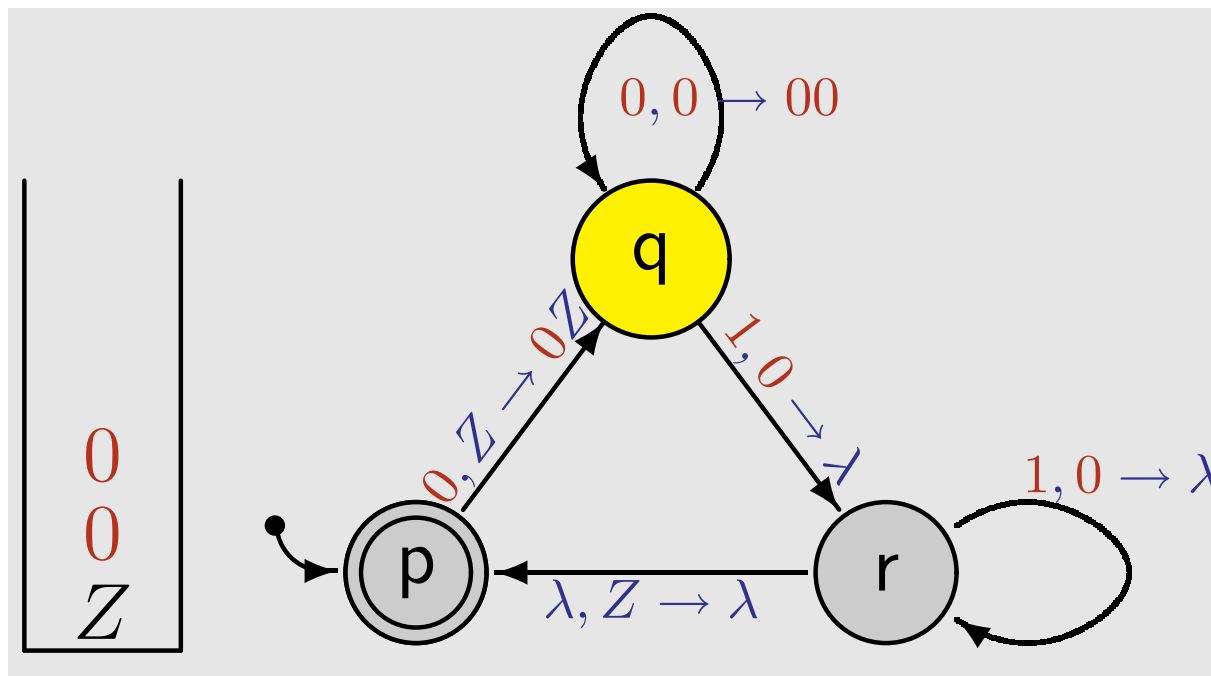
0 0 1 1
 └──────────┘
 część
 nieprzeczytana



Maszyna ze stosem

Słowo na wejściu:

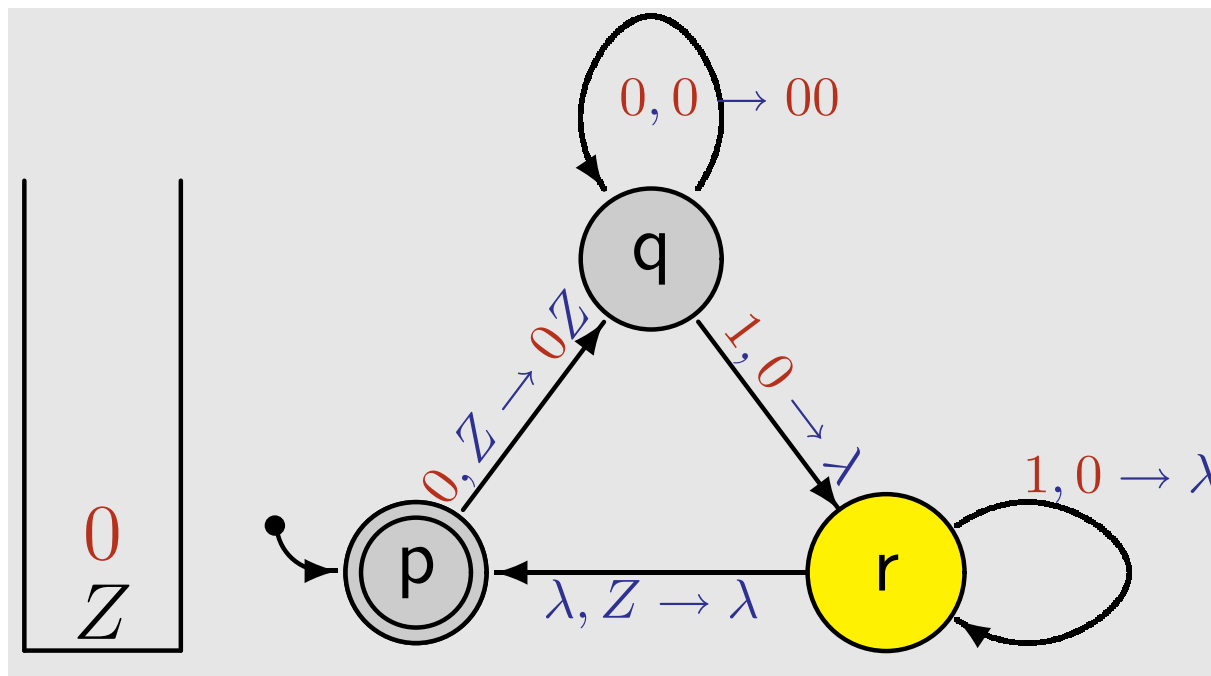
0 0 1 1
 część
 nieprzeczytana



Maszyna ze stosem

Słowo na wejściu:

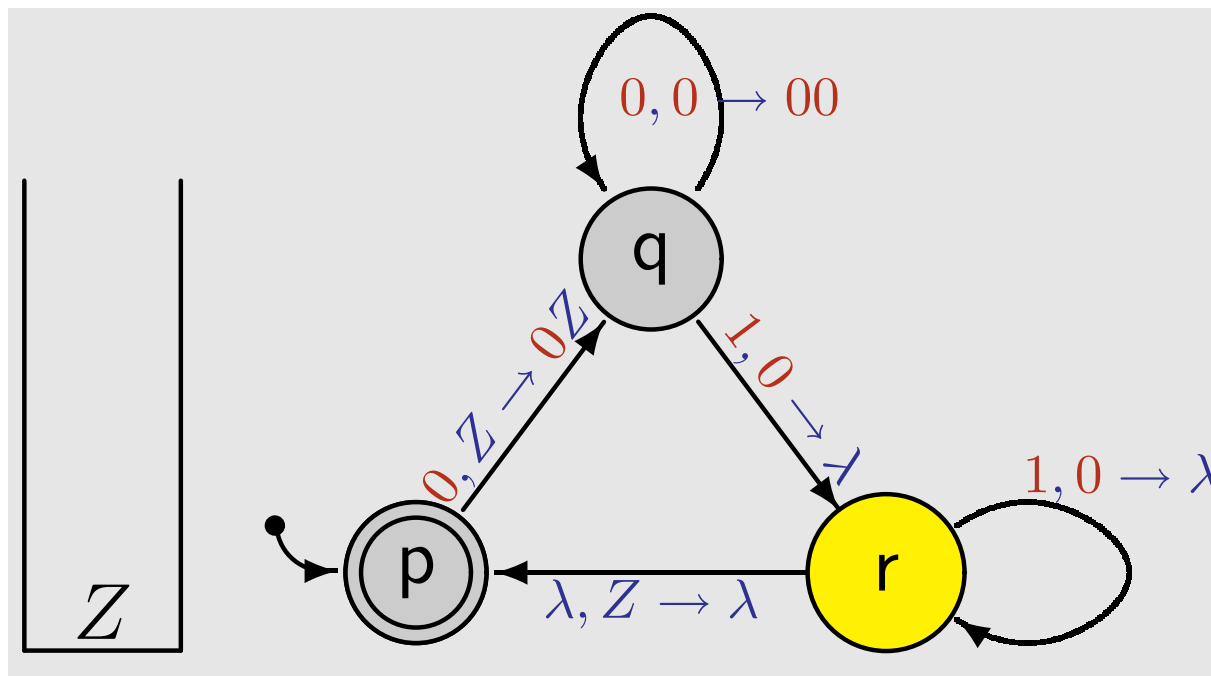
0 0 1 1
 └─┬─┘
 część
 nieprzeczytana



Maszyna ze stosem

Słowo na wejściu:

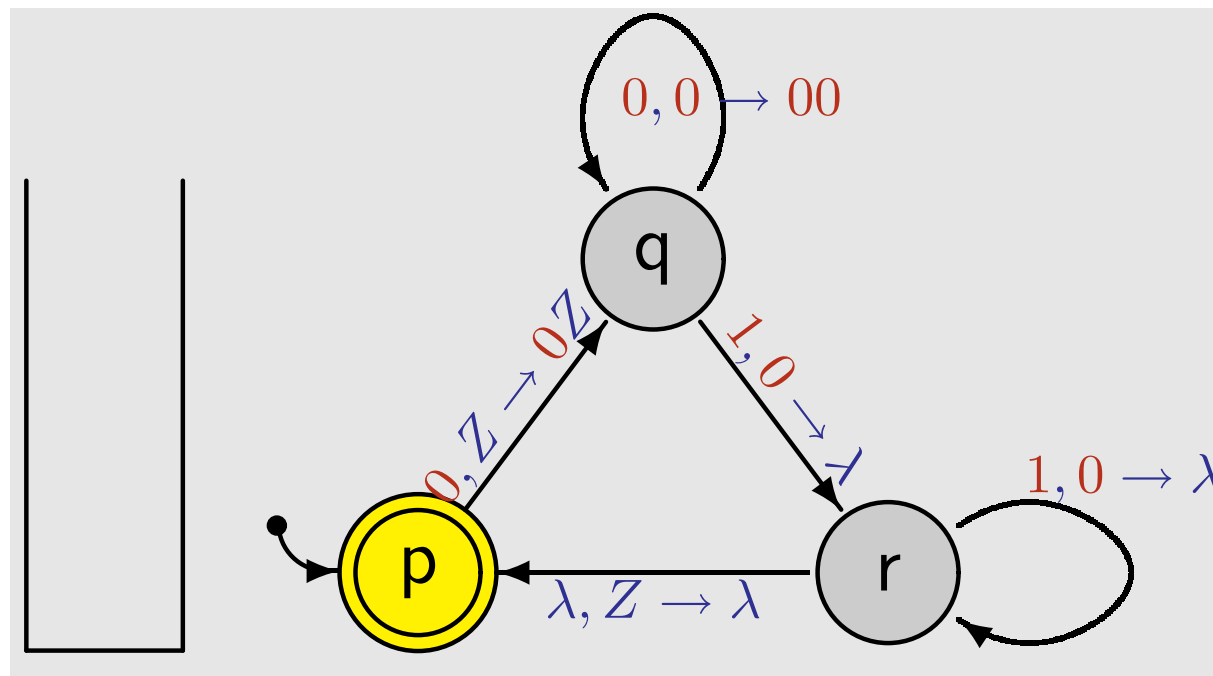
0 0 1 1 $\square$
 część
 nieprzeczytana



Maszyna ze stosem

Słowo na wejściu:

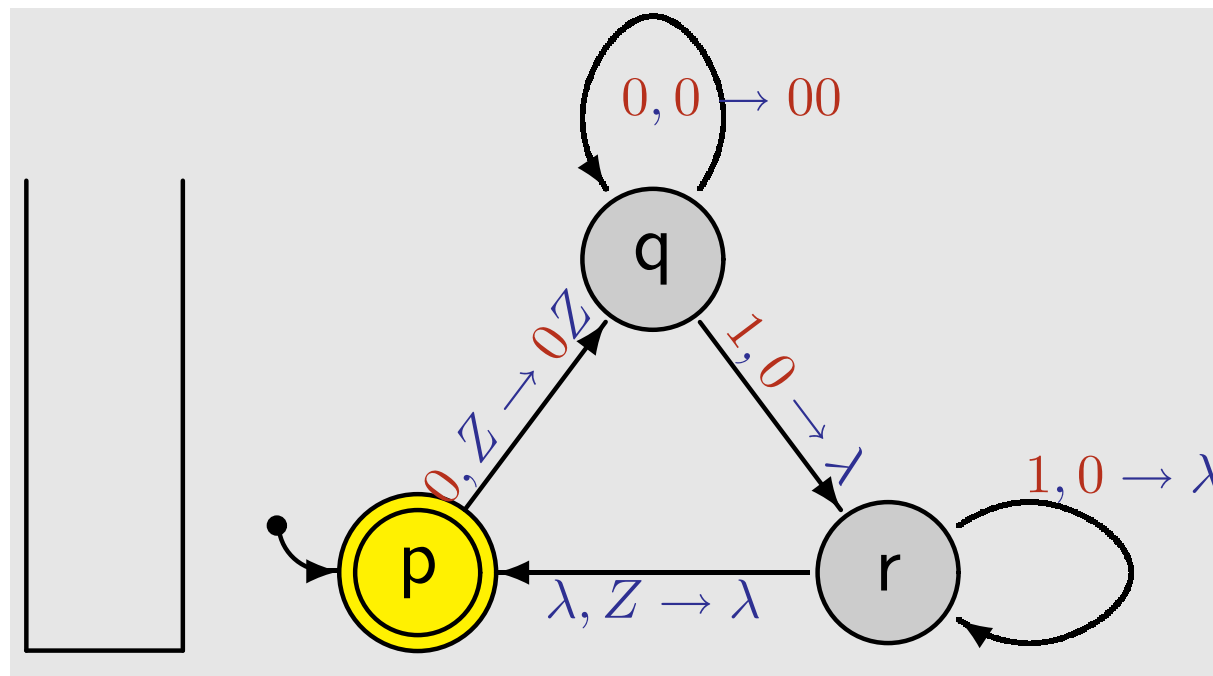
0 0 1 1 $\square$
 część
 nieprzeczytana



Maszyna ze stosem

Słowo na wejściu:

0 0 1 1 $\square$
 część
 nieprzeczytana

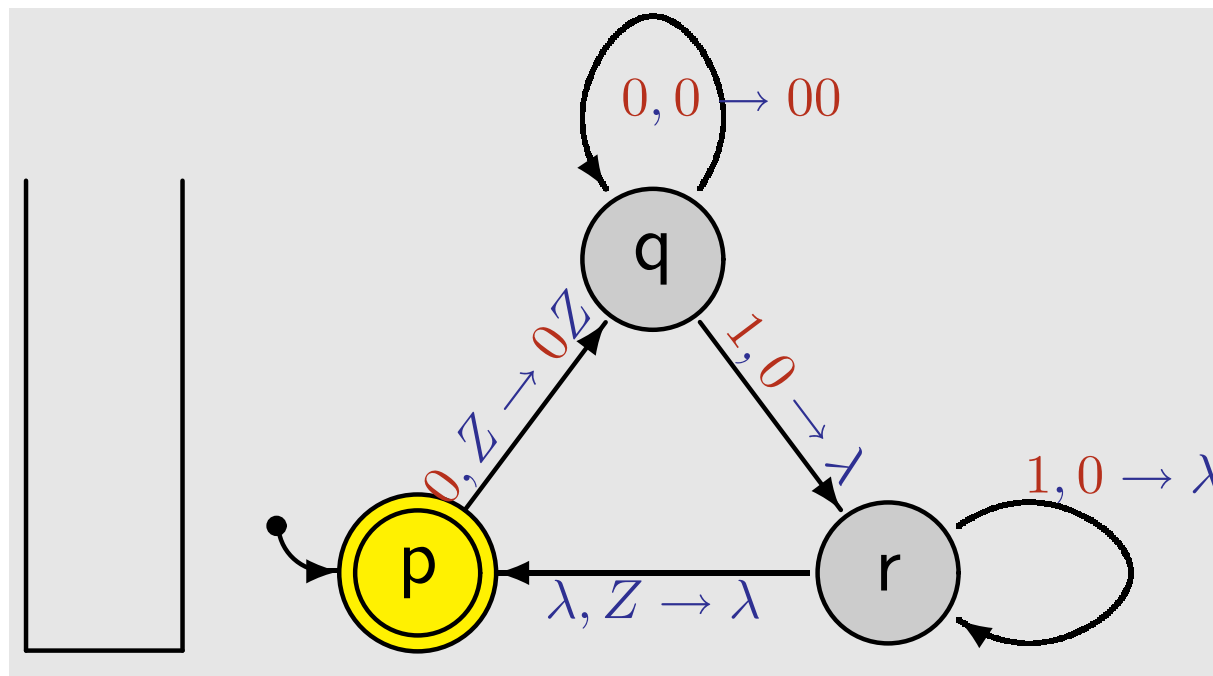


Kiedy wędrówka po maszynie się „zatnie”, czyli nie ma już żadnej możliwości pójścia dalej, sprawdzamy, czy słowo zostało **zaakceptowane**

Maszyna ze stosem

Słowo na wejściu:

0 0 1 1 $\square$
 część
 nieprzeczytana



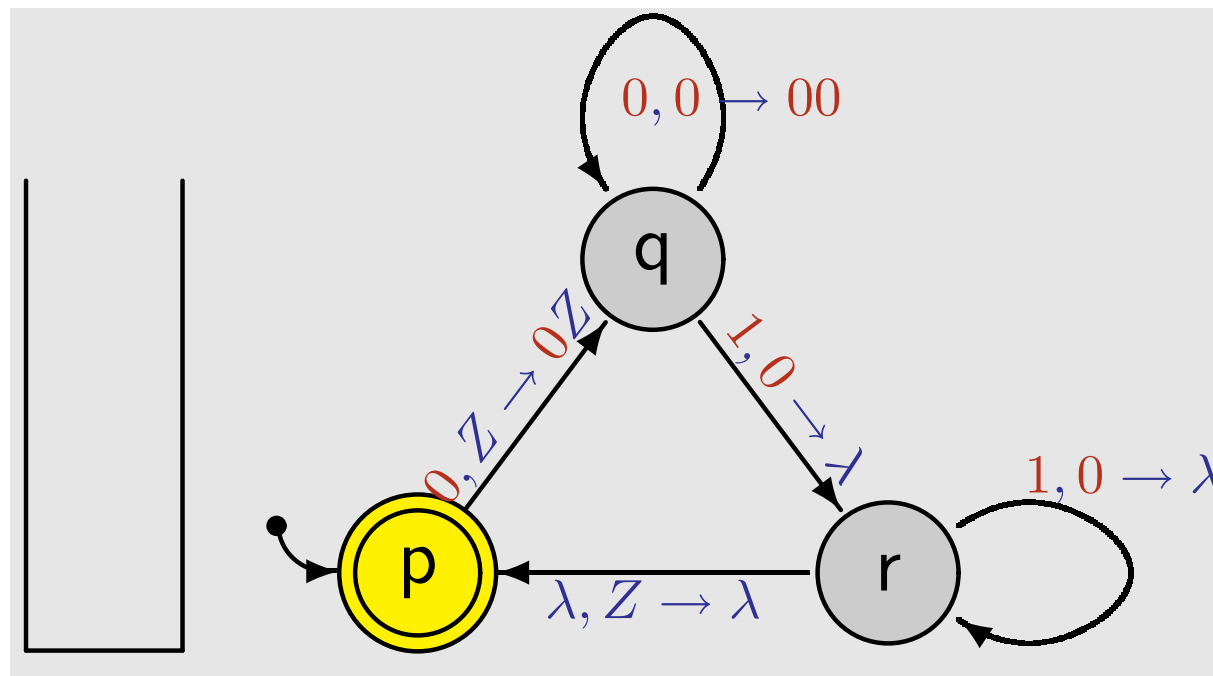
Kiedy wędrówka po maszynie się „zatnie”, czyli nie ma już żadnej możliwości pójścia dalej, sprawdzamy, czy słowo zostało **zaakceptowane**; t.zn., czy spełnione są następujące warunki:

- na **wejściu** jest słowo puste (wszystko już przeczytane)

Maszyna ze stosem

Słowo na wejściu:

0 0 1 1 $\square$
 część
 nieprzeczytana



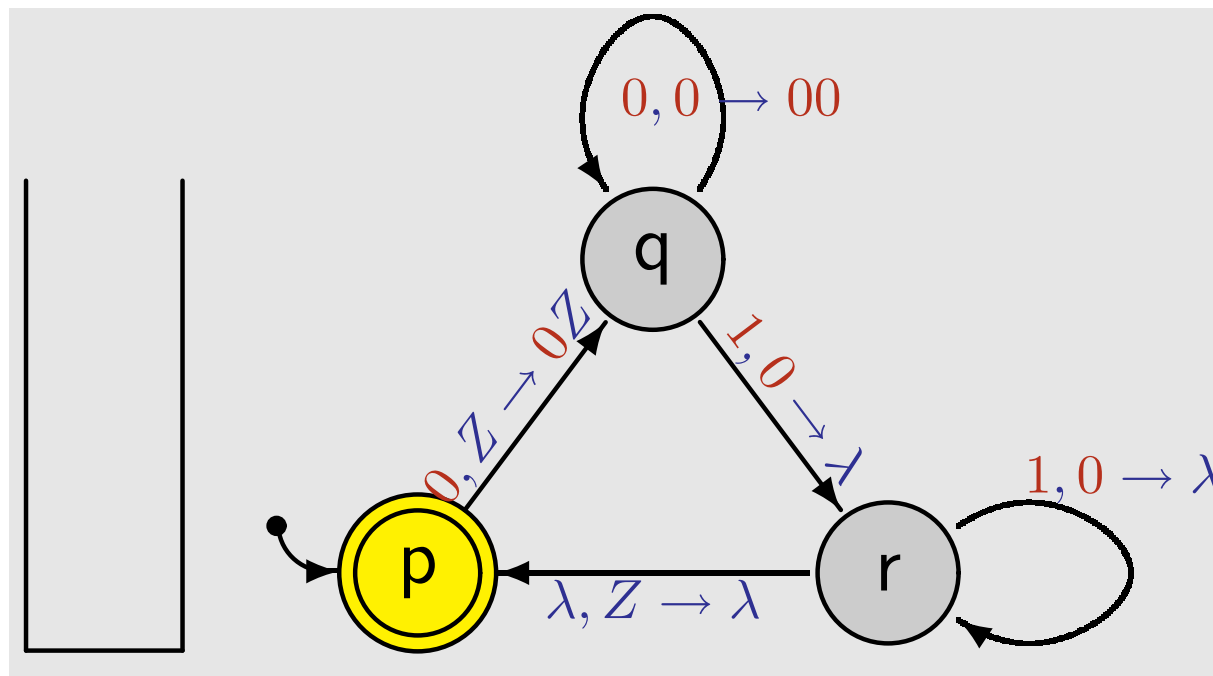
Kiedy wędrówka po maszynie się „zatnie”, czyli nie ma już żadnej możliwości pójścia dalej, sprawdzamy, czy słowo zostało **zaakceptowane**; t.zn., czy spełnione są następujące warunki:

- na **wejściu** jest słowo puste (wszystko już przeczytane);
- na **stosie** jest słowo puste

Maszyna ze stosem

Słowo na wejściu:

0 0 1 1 □
 część
 nieprzeczytana



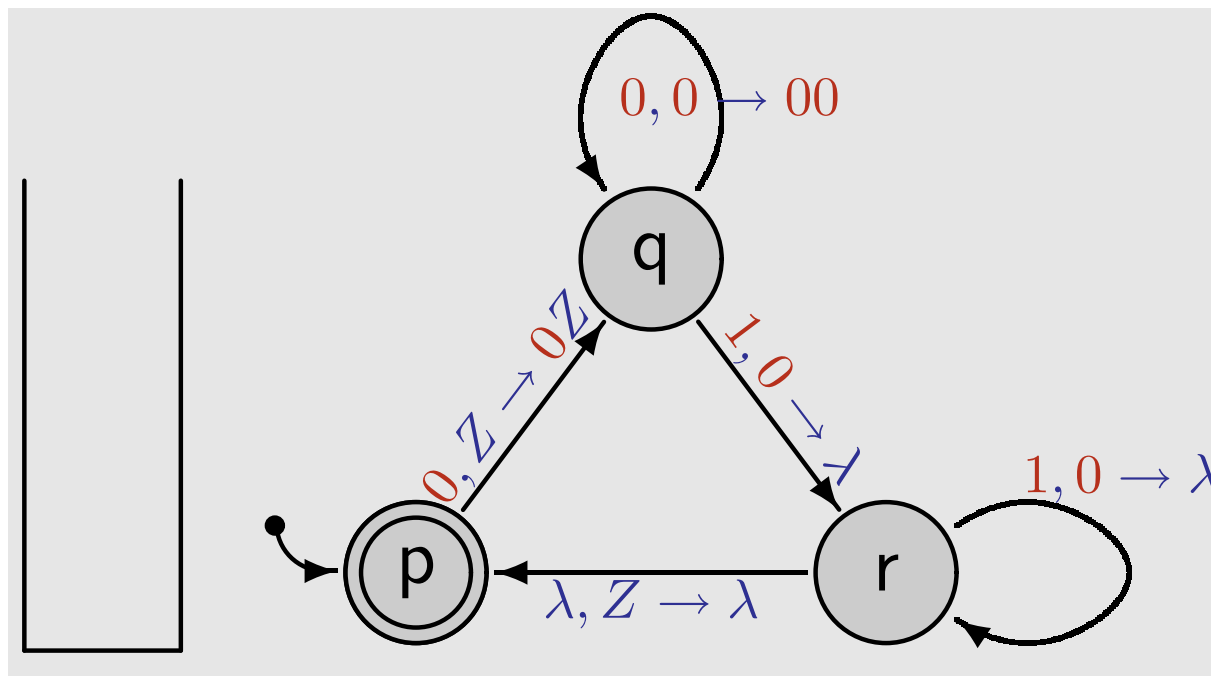
Kiedy wędrówka po maszynie się „zatnie”, czyli nie ma już żadnej możliwości pójścia dalej, sprawdzamy, czy słowo zostało **zaakceptowane**; t.zn., czy spełnione są następujące warunki:

- na **wejściu** jest słowo puste (wszystko już przeczytane);
- na **stosie** jest słowo puste;
- sprawdzanie zakończyło się w **stanie** końcowym (akceptującym).

Maszyna ze stosem

Język akceptowany:

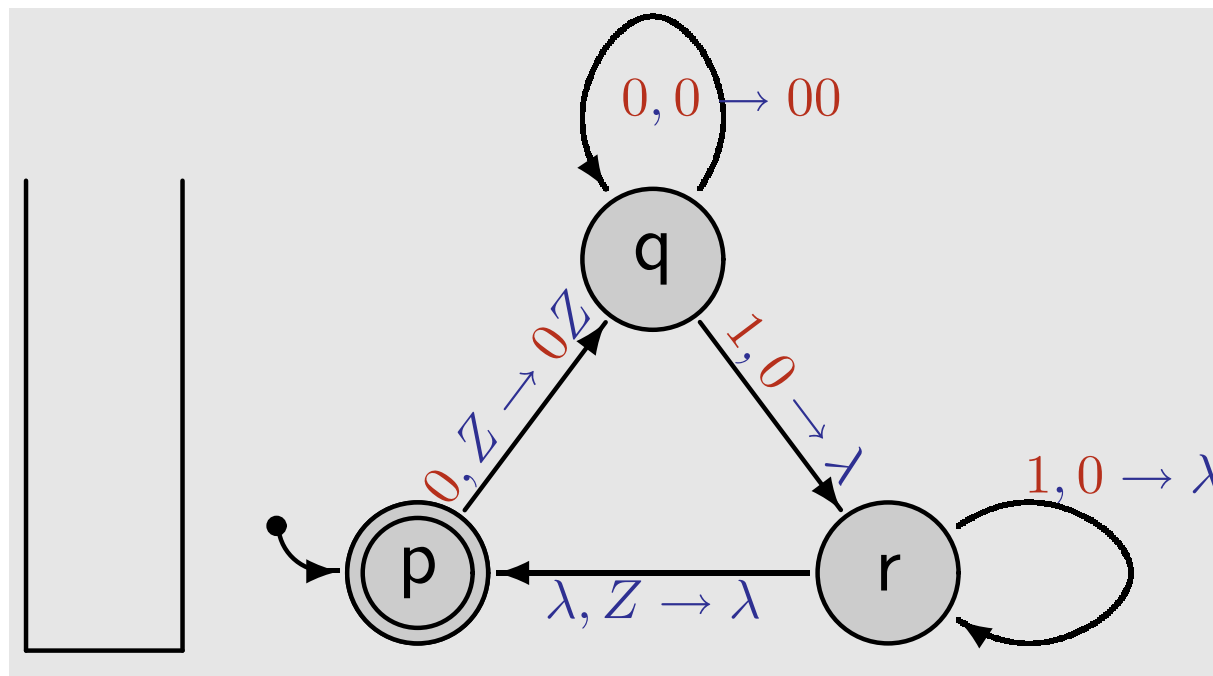
$$\{0^n 1^n \mid n \geq 1\}$$



Maszyna ze stosem

Język akceptowany:

$$\{0^n 1^n \mid n \geq 1\}$$



wejście		0		1		λ	
stos		Z	0	Z	0	Z	0
stan	p	$\langle q, 0Z \rangle$					
	q		$\langle q, 00 \rangle$		$\langle r, \lambda \rangle$		
	r				$\langle r, \lambda \rangle$	$\langle p, \lambda \rangle$	

Maszyna ze stosem

DEFINICJA: Maszyna ze stosem M —

Maszyna ze stosem

DEFINICJA: Maszyna ze stosem M —

- skończony zbiór Q *stanów*

Maszyna ze stosem

DEFINICJA: Maszyna ze stosem M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter wejściowych*

Maszyna ze stosem

DEFINICJA: Maszyna ze stosem M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter wejściowych*,
- skończony zbiór Γ *symboli stosowych*

Maszyna ze stosem

DEFINICJA: Maszyna ze stosem M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter wejściowych*,
- skończony zbiór Γ *symboli stosowych*,
- $Z \in \Gamma$ — wyróżniony symbol stosowy oznaczający dno stosu

Maszyna ze stosem

DEFINICJA: Maszyna ze stosem M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter wejściowych*,
- skończony zbiór Γ *symboli stosowych*,
- $Z \in \Gamma$ — wyróżniony symbol stosowy oznaczający dno stosu,
- częściowa funkcja *przejścia* $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightrightarrows Q \times \Gamma^*$
(częściowa — t.zn. nie musi być określona dla wszystkich argumentów)

Maszyna ze stosem

DEFINICJA: Maszyna ze stosem M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter wejściowych*,
- skończony zbiór Γ *symboli stosowych*,
- $Z \in \Gamma$ — wyróżniony symbol stosowy oznaczający dno stosu,
- częściowa funkcja *przejścia* $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightrightarrows Q \times \Gamma^*$
(częściowa — t.zn. nie musi być określona dla wszystkich argumentów),
- wyróżniony stan *początkowy* $q_0 \in Q$

Maszyna ze stosem

DEFINICJA: Maszyna ze stosem M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter wejściowych*,
- skończony zbiór Γ *symboli stosowych*,
- $Z \in \Gamma$ — wyróżniony symbol stosowy oznaczający dno stosu,
- częściowa funkcja *przejścia* $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightrightarrows Q \times \Gamma^*$
(częściowa — t.zn. nie musi być określona dla wszystkich argumentów),
- wyróżniony stan *początkowy* $q_0 \in Q$,
- wyróżniony zbiór stanów *końcowych* (*akceptujących*) $F \subseteq Q$.

Maszyna ze stosem

DEFINICJA: Maszyna ze stosem M —

- skończony zbiór Q *stanów*,
- skończony zbiór Σ *liter wejściowych*,
- skończony zbiór Γ *symboli stosowych*,
- $Z \in \Gamma$ — wyróżniony symbol stosowy oznaczający dno stosu,
- częściowa funkcja *przejścia* $\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightrightarrows Q \times \Gamma^*$
(częściowa — t.zn. nie musi być określona dla wszystkich argumentów),
- wyróżniony stan *początkowy* $q_0 \in Q$,
- wyróżniony zbiór stanów *końcowych* (*akceptujących*) $F \subseteq Q$.

DEFINICJA: Język akceptowany przez M —

$$L(M) \stackrel{\text{def}}{=} \left\{ w \in \Sigma^* \mid \delta^* \langle q_0, w, Z \rangle = \langle q', \lambda, \lambda \rangle \text{ dla pewnego } q' \in F \right\}$$

gdzie δ^* jest uogólnioną funkcją przejścia.

Maszyna ze stosem a języki bezkontekstowe

TWIERDZENIE:

Język akceptowany przez dowolną maszynę stosową jest bezkontekstowy.

Maszyna ze stosem a języki bezkontekstowe

TWIERDZENIE:

Język akceptowany przez dowolną maszynę stosową jest bezkontekstowy.

Uwaga:

Nie każdy język bezkontekstowy jest akceptowany przez jakąś **maszynę stosową**; ale każdy jest akceptowany przez jakąś **niedeterministyczną maszynę stosową** (nierozpatrywane na tym wykładzie).

Maszyna ze stosem a języki bezkontekstowe

TWIERDZENIE:

Język akceptowany przez dowolną maszynę stosową jest bezkontekstowy.

Uwaga:

Nie każdy język bezkontekstowy jest akceptowany przez jakąś **maszynę stosową**; ale każdy jest akceptowany przez jakąś **niedeterministyczną maszynę stosową** (nierozpatrywane na tym wykładzie).

„Prawie” każdy „praktyczny” język bezkontekstowy jest akceptowany przez jakąś **maszynę stosową**

Maszyna ze stosem a języki bezkontekstowe

TWIERDZENIE:

Język akceptowany przez dowolną maszynę stosową jest bezkontekstowy.

Uwaga:

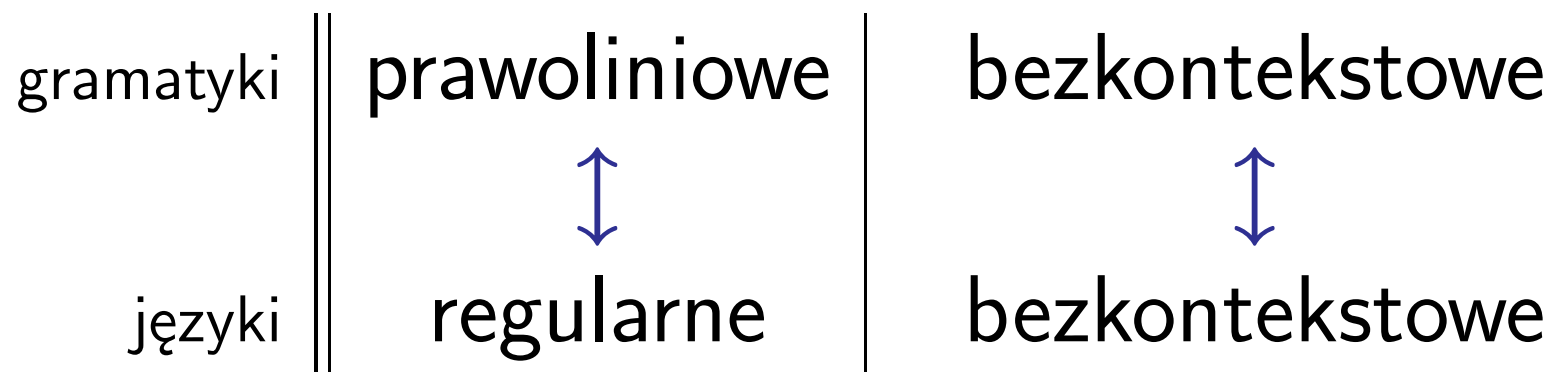
Nie każdy język bezkontekstowy jest akceptowany przez jakąś **maszynę stosową**; ale każdy jest akceptowany przez jakąś **niedeterministyczną maszynę stosową** (nierozpatrywane na tym wykładzie).

„Prawie” każdy „praktyczny” język bezkontekstowy jest akceptowany przez jakąś **maszynę stosową**; to umożliwia kompilowanie programów komputerowych.

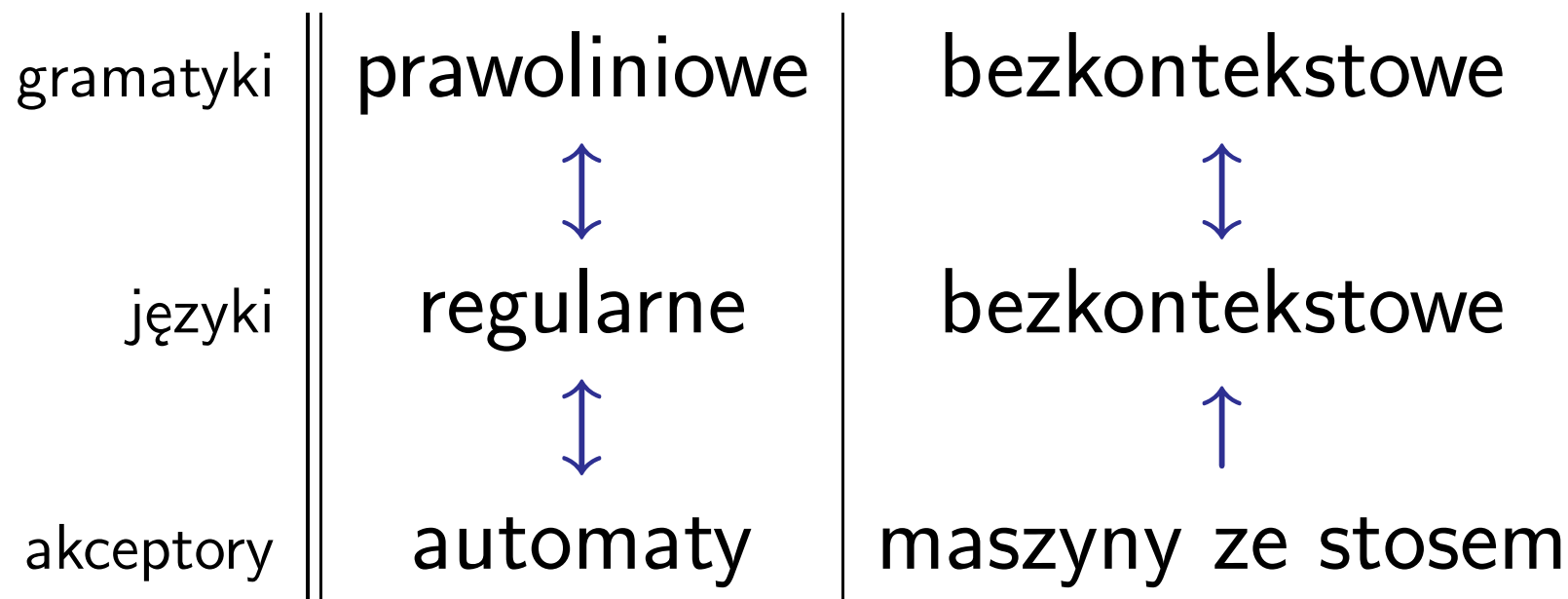
Gramatyki, języki, maszyny

gramatyki || prawoliniowe | bezkontekstowe

Gramatyki, języki, maszyny



Gramatyki, języki, maszyny



Gramatyki, języki, maszyny

