AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Przekształcenia automatów skończonych

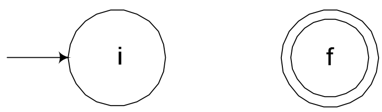
Teoria automatów i języków formalnych

Dr inż. Janusz Majewski
Katedra Informatyki

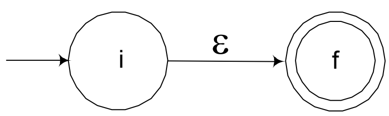
Konstrukcja automatu skończonego na podstawie wyrażenia regularnego (*algorytm Thompsona*)

Wejście: wyrażenie regularne ***r*** nad alfabetem *Σ*
Wyjście: automat skończony akceptujący język *L(**r**)* (język opisany wyrażeniem regularnym ***r***)
Metoda: wyodrębnić z wyrażenia regularnego ***r*** elementy podstawowe. Dla elementów podstawowych skonstruować odpowiadające im automaty, a następnie połączyć je według poniższych zasad:

- Dla ∅ zbudować A(∅)

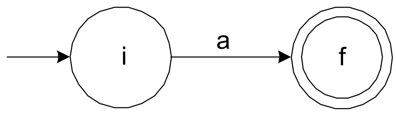


- Dla ε zbudować A(ε)

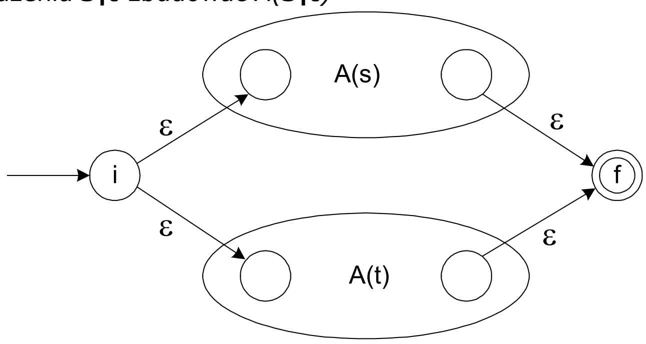


Konstrukcja automatu skończonego na podstawie wyrażenia regularnego (*algorytm Thompsona*)

- Dla a∈Σ zbudować A(**a**)

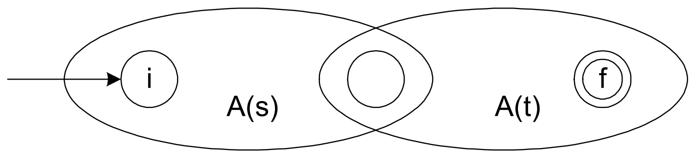


- Gdy *A(**s**)* i *A(**t**)* są automatami dla wyrażeń regularnych ***s*** i ***t*** , to dla wyrażenia ***s|t*** zbudować *A(**s|t**)*

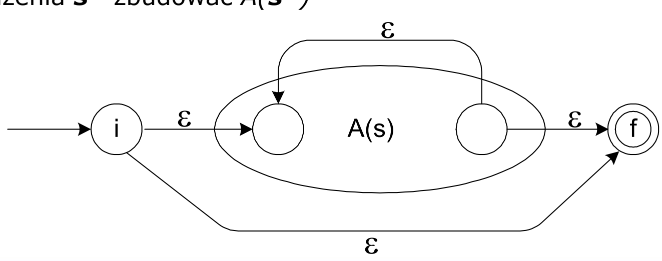


Konstrukcja automatu skończonego na podstawie wyrażenia regularnego (*algorytm Thompsona*)

- Gdy *A(**s**)* i *A(**t**)* są automatami dla wyrażeń regularnych ***s*** i ***t*** , to dla wyrażenia ***st*** zbudować *A(**st**)*

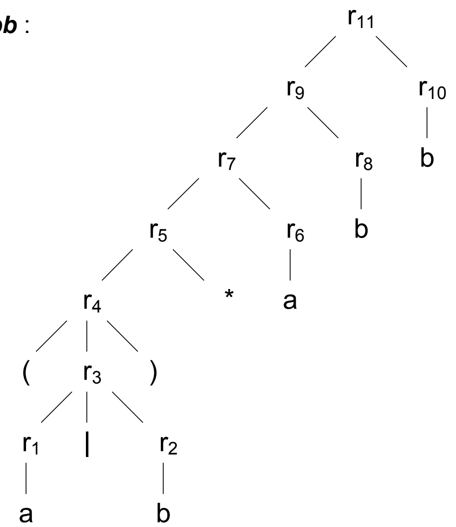


- Gdy *A(**s**)* jest automatami dla wyrażenia regularnego ***s***, to dla wyrażenia ***s**** zbudować *A(**s***)*



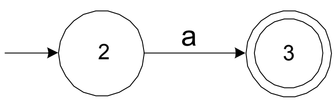
Przykład: konstrukcja automatu skończonego dla wyrażenia regularnego *r* = *(a|b)*abb*

Rozkład wyrażenia ***(a|b)*abb*** :

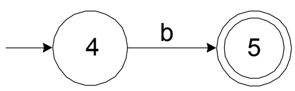


Przykład: konstrukcja automatu skończonego dla wyrażenia regularnego *r* = *(a|b)*abb*

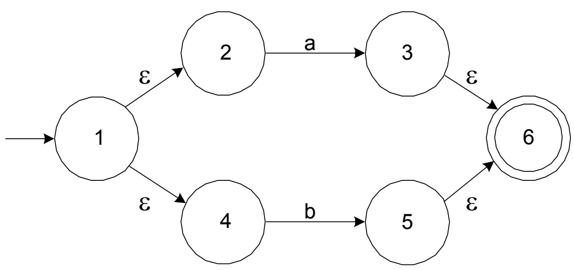
$\mathbf{r_1 = a}$



$\mathbf{r_1 = b}$



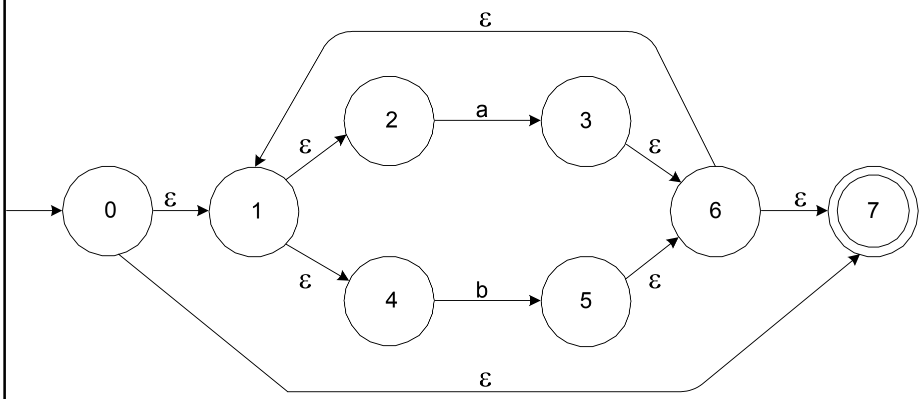
$\mathbf{r_3 = a|b = r_1|r_2}$



Przykład: konstrukcja automatu skończonego dla wyrażenia regularnego *r* = *(a|b)*abb*

$r_4 = (r_3)$

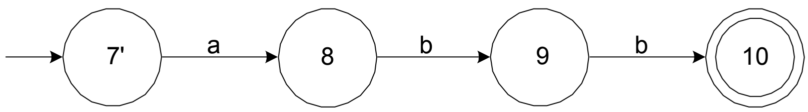
$r_5 = r_4{}^*$



Przykład: konstrukcja automatu skończonego dla wyrażenia regularnego *r* = *(a|b)*abb*

r_6 = **a** ; r_8 = **b** ; r_{10} = **b** - konstrukcje identyczne jak dla r_1 i r_2

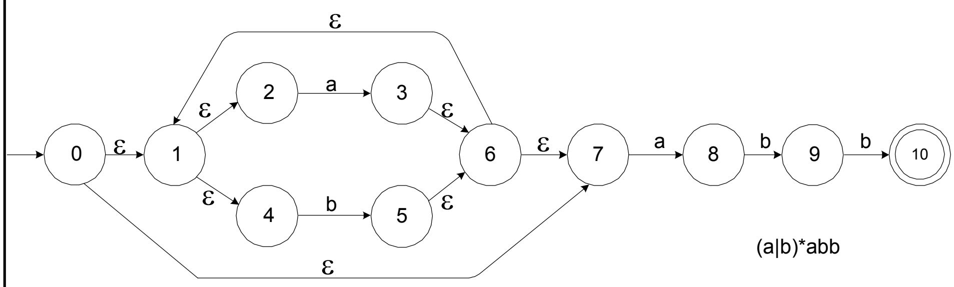
r_x = **abb**



$r = r_5 r_x$ = **(a|b)*abb**

Ostatecznie otrzymujemy:

Przykład: konstrukcja automatu skończonego dla wyrażenia regularnego *r* = *(a|b)*abb*



Konstrukcja automatu deterministycznego na podstawie automatu niedeterministycznego

Dla każdego automatu skończonego istnieje deterministyczny automat skończony akceptujący ten sam język.

Dla $q \in Q$ definiuje się zbiór ε-CLOSURE(q) zawierający te stany $r \in Q$, do których można dojść z q przechodząc tylko przez ε-przejścia, przy czym również $q \in \varepsilon$-CLOSURE(q).

Dla $S \subseteq Q$ definiuje się zbiór ε-CLOSURE(S) zawierający te stany $r \in Q$, do których można dojść ze stanów S przechodząc tylko przez ε-przejścia, przy czym również $S \subseteq \varepsilon$-CLOSURE(S).

Dla $S \subseteq Q$, dla $a \in \Sigma$ rozszerza się definicję funkcji przejścia:

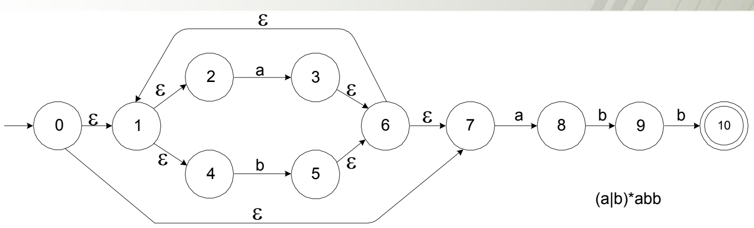
$$\delta(S,a) = \{ r \in Q \mid r \in \delta(s,a),\ s \in S \}$$

Konstrukcja automatu deterministycznego na podstawie automatu niedeterministycznego

Wejście: $A=\langle \Sigma, Q, F, q_0, \delta \rangle$ - automat skończony niedeterministyczny
Wyjście: $A'=\langle \Sigma, Q', F', r_0, \delta' \rangle$ - automat skończony deterministyczny (bez ε-przejść)
Metoda: $Q \supseteq S \mapsto r \in Q'$ *\/* podzbiór zbioru stanów $\mapsto$ pojedynczy stan */*

```
r0 := ε-CLOSURE({q0});   r0 - nieoznaczony;  /* r0 – stan początkowy A' i równocześnie
                                                  podzbiór zbioru stanów Q automatu A */
Q' := {r0};
while ∃ X∈Q' and X – nieoznaczony do          /* X = {q1,...,qk} ⊆ Q*/
  begin
    oznacz X;
    for każde a∈Σ do
      begin
        U := {q∈Q | q∈δ(s,a) ∧ s∈X }           /* U = δ(X,a) */
        Y := ε-CLOSURE(U) ;
        if Y∉Q' then
          begin
            Q' := Q' ∪ {Y};  Y – nieoznaczony; /* dołączenie Y do Q' jako nieoznaczonego*/
          end;
        δ'(X,a) := Y;                           /* ustalenie funkcji przejścia automatu A' */
      end;
  end;
F' := { r ∈ Q' | r ∩ F ≠ ∅ }  /* tutaj r traktowane jako (r⊂Q) podzbiór stanów automatu A */
```

Przykład (1)



$r_0 = \varepsilon\text{-CLOSURE}(\{0\}) = \{ \underline{0,1,2,4,7} \} = r_0$; $Q'=\{ r_0 \}$

---

r_0 – oznaczamy

$U_a = \delta(r_0,a) = \{3,8\}$

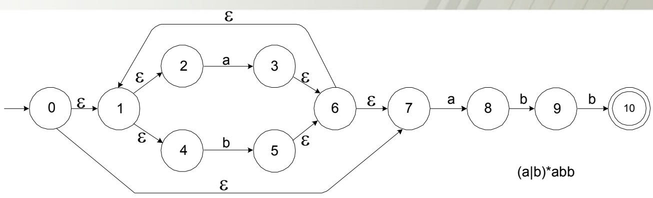
$r_1 = \varepsilon\text{-CLOSURE}(\{3,8\}) = \{ \underline{1,2,3,4,6,7,8} \} = r_1$; $\delta'(r_0,a) = r_1$

$U_b = \delta(r_0,b) = \{5\}$

$r_2 = \varepsilon\text{-CLOSURE}(\{5\}) = \{ \underline{1,2,4,5,6,7} \} = r_2$; $\delta'(r_0,b) = r_2$

$Q' = \{ \underline{r_0} , r_1 , r_2 \}$ */* stan podkreślony jest oznaczony */*

Przykład (2)



(a|b)*abb

r_1 – oznaczamy

$U_a = \delta(r_1,a) = \{3,8\}$

ε-CLOSURE({3,8}) = { $\underline{1,2,3,4,6,7,8}$ }= r_1 ; $\delta'(r_1,a) = r_1$

$U_b = \delta(r_1,b) = \{5,9\}$

ε-CLOSURE({5,9}) = { $\underline{1,2,4,5,6,7,9}$ }= r_3 ; $\delta'(r_1,b) = r_3$

$Q' = \{ \underline{r_0} , \underline{r_1} , r_2 , r_3 \}$

---

r_2 – oznaczamy

$U_a = \delta(r_2,a) = \{3,8\}$

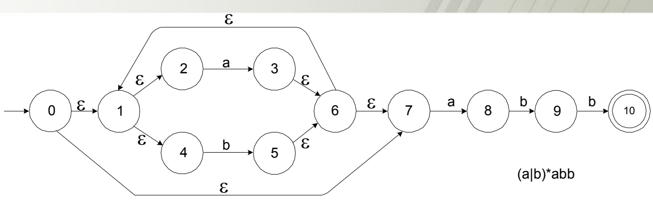
ε-CLOSURE({3,8}) = r_1 ; δ'(r2,a) = r1

$U_b = \delta(r_2,b) = \{5\}$

ε-CLOSURE({5}) = r_2 ; $\delta'(r_2,b) = r_2$

$Q' = \{ \underline{r_0} , \underline{r_1} , \underline{r_2} , r_3 \}$

Przykład (3)



(a|b)*abb

r_3 – oznaczamy

$U_a = \delta(r_3,a) = \{3,8\}$

ε-CLOSURE({3,8}) = r_1 ; $\delta'(r_3,a) = r_1$

$U_b = \delta(r_3,b) = \{5,10\}$

ε-CLOSURE({5,10}) = { $\underline{1,2,4,5,6,7,10}$ } = r_4 ; $\delta'(r_3,b) = r_4$

$Q' = \{ \underline{r_0} , \underline{r_1} , \underline{r_2} , \underline{r_3} , r_4 \}$

---

r_4 – oznaczamy

$U_a = \delta(r_4,a) = \{3,8\}$

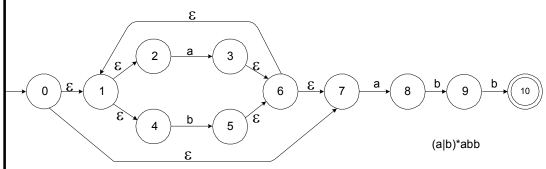
ε-CLOSURE({3,8}) = r_1 ; $\delta'(r_4,a) = r_1$

$U_b = \delta(r_4,b) = \{5\}$

ε-CLOSURE({5,10}) = r_2 ; $\delta'(r_3,b) = r_2$

$Q' = \{ \underline{r_0} , \underline{r_1} , \underline{r_2} , \underline{r_3} , \underline{r_4} \}$

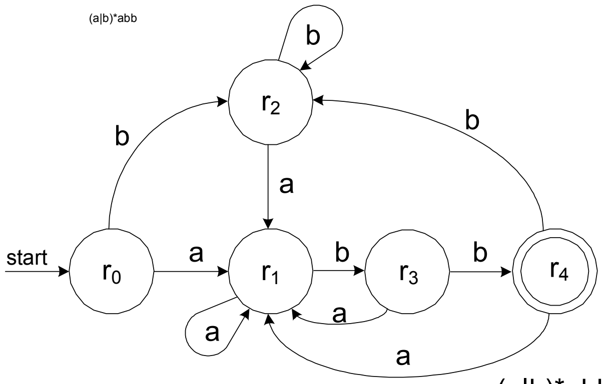
Przykład (4)



Ostatecznie:

Stan	a	b
r_0	r_1	r_2
r_1	r_1	r_3
r_2	r_1	r_2
r_3	r_1	r_4
r_4	r_1	r_2

$F'=\{r_4\}$



(a|b)*abb

Uzupełnienie automatu skończonego

Wejście: $A = < \Sigma, Q, F, q_0, \delta >$ - automat skończony

Wyjście: $A' = < \Sigma, Q', F, q_0, \delta' >$ - automat skończony zupełny

$Q' := Q \cup \{$ <u>err</u> $\}$

<u>for</u> $q \in Q$ <u>do</u>

  <u>for</u> $a \in \Sigma$ <u>do</u>

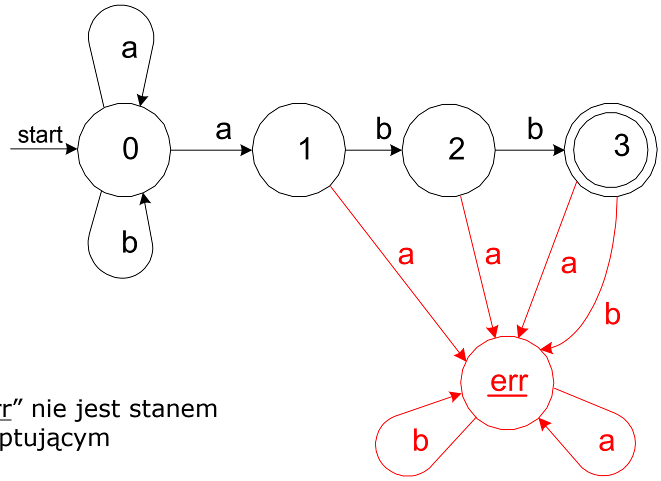
    <u>if</u> $\delta(q,a) = \varnothing$ <u>then</u> $\delta'(q,a) := \{$ <u>err</u> $\}$

    <u>else</u> $\delta'(q,a) := \delta(q,a)$;

<u>for</u> $a \in \Sigma$ <u>do</u>

  $\delta'($<u>err</u>$,a) := \{$ <u>err</u> $\}$

Przykład



Stan pułapki „err” nie jest stanem końcowym akceptującym

Redukcja automatu skończonego

$A = < \Sigma, Q, F, q_0, \delta >$ - deterministyczny, zupełny automat skończony

$x \in \Sigma^*$ - słowo nad alfabetem Σ

$q_1, q_2, q_3, q_4 \in Q$ – stany automatu A

- $x \in \Sigma^*$ rozróżnia stany q_1 i $q_2 \Leftrightarrow$
  - (1) $(q_1, x) \vdash_A^* (q_3, \varepsilon)$
  - (2) $(q_2, x) \vdash_A^* (q_4, \varepsilon)$
  - (3) $(q_3 \in F \wedge q_4 \notin F) \vee (q_3 \notin F \wedge q_4 \in F)$
- q_1 i q_2 są k–nierozróżnialne, co oznaczamy $q_1 \equiv^k q_2 \Leftrightarrow \neg(\exists x \in \Sigma^*)$ takie że: x rozróżnia q_1 i q_2 oraz $|x| \leq k$
- q_1 i q_2 są nierozróżnialne, co oznaczamy $q_1 \equiv q_2 \Leftrightarrow (\forall k \geq 0)\ (q_1 \equiv^k q_2)$
- $q \in Q - \{q_0\}$ jest nieosiągalny $\Leftrightarrow \neg(\exists x \in \Sigma^*)\ ((q_0, x) \vdash_A^+ (q, y) \wedge y \in \Sigma^*)$
- Automat skończony (deterministyczny, zupełny) nazywamy zredukowanym $\Leftrightarrow$
  - (1) $\neg(\exists\, q \in Q)$ (q jest nieosiągalny)
  - (2) $(\forall q_1, q_2 \in Q)$ (q_1 i q_2 <u>nie</u> są nierozróżnialne)

Usuwanie stanów nieosiągalnych

$A = < \Sigma, Q, F, q_0, \delta >$ - deterministyczny automat skończony

Niech R będzie relacją ($R \subseteq Q \times Q$) zdefiniowaną następująco:

$q_1 \, R \, q_2 \Leftrightarrow (\exists a \in \Sigma) \, (\delta(q_1,a) = q_2)$

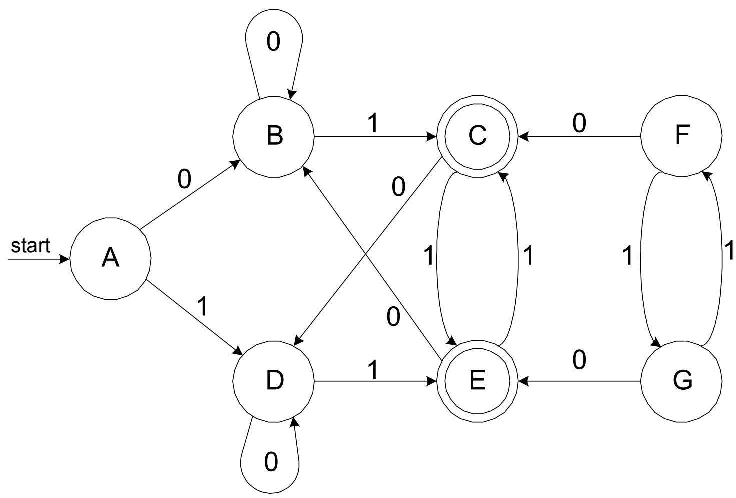
Trzeba znaleźć automat $A' = < \Sigma, Q', F', q_0, \delta' >$ bez stanów nieosiągalnych, to znaczy trzeba wyznaczyć

$Q' = \{ q \in Q \mid q_0 \, R^* \, q \}$

Zakładamy, że elementy Q są nieoznaczone.

```
L := {q0};
while L ≠ ∅ do
    begin
        b := pierwszy element z L;
        oznacz b w Q;
        L := L – {b};
        L := L ∪ { c∈Q | b R c  ∧  c – nieoznaczone w Q };
    end;
stop;      /* elementy nieoznaczone w Q są nieosiągalne */
```

Przykład usuwania stanów nieosiągalnych (1)

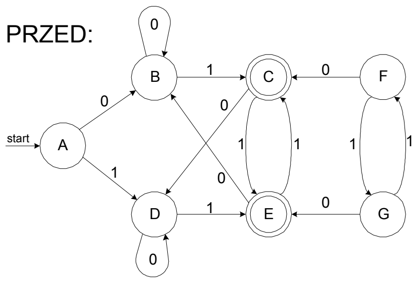


Przykład usuwania stanów nieosiągalnych (2)

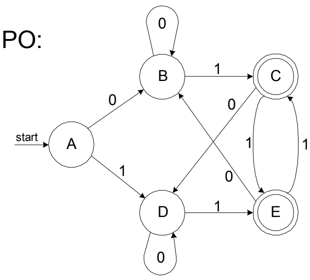
L = { A };	$\underline{A}$; Q = { $\underline{A}$, B, C, D, E, F, G };	L = ∅
L = { B, D };	$\underline{B}$; Q = { $\underline{A}$, $\underline{B}$, C, D, E, F, G };	L = { D }
L = { D, C };	$\underline{D}$; Q = { $\underline{A}$, $\underline{B}$, C, $\underline{D}$, E, F, G };	L = { C }
L = { C, E};	$\underline{C}$; Q = { $\underline{A}$, $\underline{B}$, $\underline{C}$, $\underline{D}$, E, F, G };	L = { E }
L = { E };	$\underline{E}$; Q = { $\underline{A}$, $\underline{B}$, $\underline{C}$, $\underline{D}$, $\underline{E}$, F, G };	L = ∅

nieoznaczone = nieosiągalne

PRZED:



PO:



Łączenie stanów nierozróżnialnych

A = < Σ, Q, F, q_0, δ > - automat skończony, bez stanów nieosiągalnych, deterministyczny, zupełny.

Twierdzenie 1. Niech n = #Q

$(\equiv) \subseteq (\equiv^{n-2}) \subseteq (\equiv^{n-3}) \subseteq \ldots \subseteq (\equiv^{1}) \subseteq (\equiv^{0})$

przy czym: $(\equiv) \subseteq Q{\times}Q$; $(\equiv^k) \subseteq Q{\times}Q$

$q_1 \equiv^0 q_2 \Leftrightarrow (q_1{\in}F \wedge q_2{\in}F) \vee (q_1{\notin}F \wedge q_2{\notin}F)$

$q_1 \equiv^k q_2 \Leftrightarrow q_1 \equiv^{k-1} q_2 \wedge (\forall a{\in}\Sigma)\ (\delta(q_1,a) \equiv^{k-1} \delta(q_2,a))$

Twierdzenie 2. Relacja nierozróżnialności $(\equiv) \subseteq Q{\times}Q$ jest zwrotna, symetryczna i przechodnia, jest więc relacją równoważności.

Algorytm łączenia stanów nierozróżnialnych polega na wyznaczeniu relacji nierozróżnialności $(\equiv) \subseteq Q{\times}Q$, a następnie przypisaniu każdej klasie równoważności relacji $(\equiv)$ stanu tworzonego automatu zredukowanego. Relację $(\equiv)$ wyznaczamy zgodnie z tw1. poczynając od $(\equiv^0)$ i dokonując kolejnych podziałów Q na klasy równoważności.

Algorytm łączenia stanów nierozróżnialnych

Wejście: A = < Σ, Q, F, q_0, δ > - deterministyczny, zupełny, bez stanów nieosiągalnych

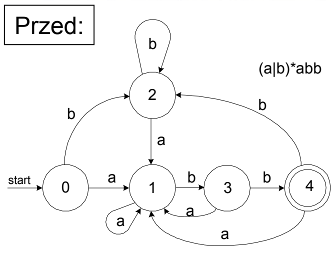
Wyjście: A′ = < Σ, Q′, F′, q_0′, δ′ > - automat posiadający najmniejszą liczbę stanów spośród wszystkich automatów deterministycznych i zupełnych akceptujących język L(A)

1) Podzielić Q na klasy równoważności dla relacji ($\equiv^0$) , ($\equiv^1$) , ... Postępować tak długo, aż podziały : dla ($\equiv^k$) i dla ($\equiv^{k+1}$) będą identyczne. Jako podział względem relacji (≡) przyjąć ten dla ($\equiv^k$)
2) Oznaczamy $[q]_\equiv$ klasę równoważności relacji (≡) w Q, do której należy q∈Q
3) Q′ := { $[p]_\equiv$ | p ∈ Q }
4) δ′($[p]_\equiv$, a) := $[q]_\equiv$ ⇔ δ(p,a) = q
5) q_0′ := $[q_0]_\equiv$
6) F′ := { $[q]_\equiv$ | q∈F }

Przykład

Relacja	Klasy równoważności	Przejścia		
$\equiv^0$	{4} {0, 1, 2, 3}	δ(0,a)=1 δ(1,a)=1 δ(2,a)=1 δ(3,a)=1	δ(0,b)=2 δ(1,b)=3 δ(2,b)=2 δ(3,b)=4	$2 \equiv^0 3$
$\equiv^1$	{4} {0, 1, 2} {3}		δ(0,b)=2 δ(2,b)=2 δ(1,b)=3	$2 \equiv^1 2$
$\equiv^2$	{4} {0, 2} {1} {3}		δ(0,b)=2 δ(2,b)=2	$2 \equiv^2 2$
$\equiv^3$	{4} {0,2} {1} {3} akceptujący początkowy			

Przed:



Po: