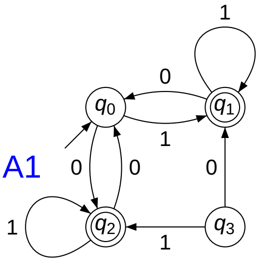
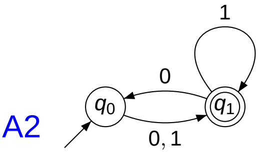
Minimalizace KA - Úvod

- Tyto dva KA **A1,A2** jsou jazykově ekvivalentní, tzn. že rozpoznávají tentýž jazyk. $L(A1) = L(A2)$
- Názorně lze vidět, že automat A2 má menší počet stavů než A1, tudíž našim cílem bude ukázat jakými způsoby lze zmenšit počet stavů KA a tak dospět k automatu s nejmenším počtem stavů.





Minimalizace KA - Úvod

Zredukování počtu stavů KA sestává ze dvou kroků:

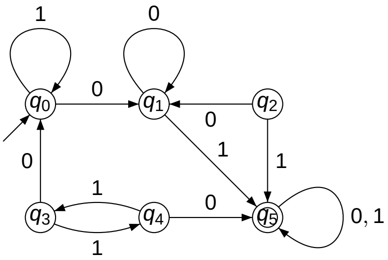
- Eliminace nedosažitelných stavů

Minimalizace KA - Úvod

Zredukování počtu stavů KA sestává ze dvou kroků:

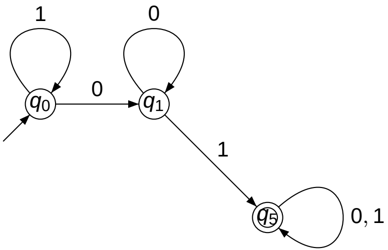
- Eliminace nedosažitelných stavů
- Sjednocení ekvivalentních stavů

Eliminace nedosažitelných stavů



- Automat přijímá jazyk $L = \{w \in 0, 1^* | w \text{ obsahuje podslovo } 01\}$
- Pro žádnou posloupnost vstupních symbolů se automat nedostane do stavů q_2, q_3 nebo q_4 .

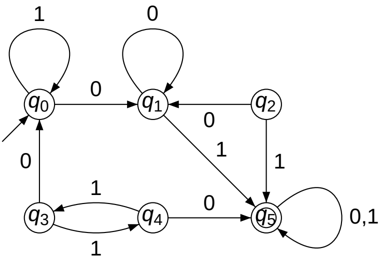
Eliminace nedosažitelných stavů



- Automat přijímá jazyk $L = \{w \in 0, 1^* | w \text{ obsahuje podslovo } 01\}$
- Pro žádnou posloupnost vstupních symbolů se automat nedostane do stavů q_2, q_3 nebo q_4 .
- Pokud tyto stavy odstraníme, automat pořád přijímá stejný jazyk $L = \{w \in 0, 1^* | w \text{ obsahuje podslovo } 01\}$

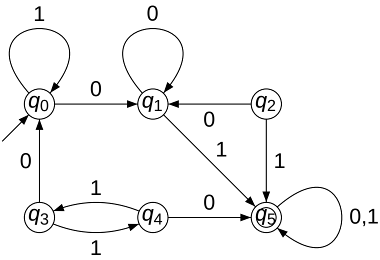
Eliminace nedosažitelných stavů

Nyní si ukážeme na příkladu, jak lze odstranit z automatu nedosažitelné stavy.



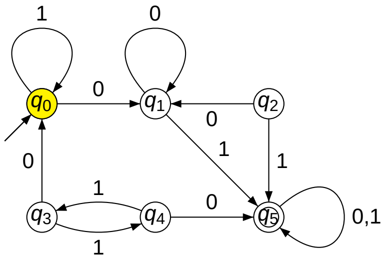
Eliminace nedosažitelných stavů

- Princip spočívá v tom, že procházíme graf a tak určujeme dosažitelné stavy.
- Z toho plyne, že které nejsou dosažitelné, jsou nedosažitelné a můžeme je vynechat, aniž by se změnil jazyk přijímaný automatem.



Eliminace nedosažitelných stavů

Nyní začneme procházet automat. Počáteční stav q_0 je označen jako dosažitelný. Jakmile probereme všechny jeho výstupní šipky, kterými se dostáváme do dalších stavů, označíme jej jako "vyřízený" (zpracovaný).



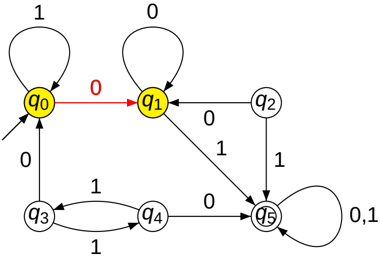
Dosažitelné stavy = [q_0,

Vyřízené stavy = [

Nedosažitelné stavy = [

Eliminace nedosažitelných stavů

Nyní se z počátečního stavu q_0 dostaneme slovem 0 do stavu q_1.
Označíme jej tedy jako dosažitelný.



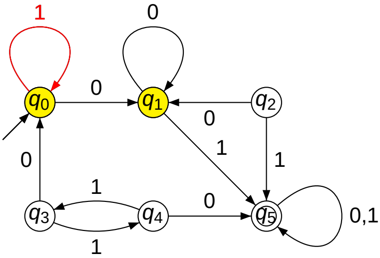
Dosažitelné stavy = [q_0, q_1,

Vyřízené stavy = [

Nedosažitelné stavy = [

Eliminace nedosažitelných stavů

Nyní ze stavu q_0 přejdeme slovem 1 do dalšího stavu.
Vidíme, že tímto slovem se dostáváme do stavu q_0. Ten je již označen jako dosažitelný.
Stav q_0 označíme jako vyřízený, protože jsme probrali všechny jeho výstupní šipky.



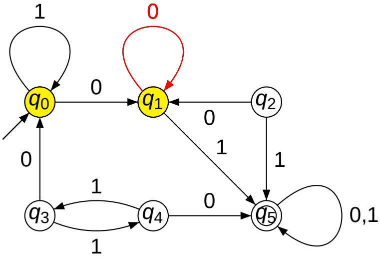
Dosažitelné stavy = [q_0, q_1,

Vyřízené stavy = [q_0,

Nedosažitelné stavy = [

Eliminace nedosažitelných stavů

Nyní budeme přecházet ze stavu, který je označen jako dosažitelný, ale není přitom označen jako vyřízený, a to je stav q_1.
Přejdeme tedy z tohoto stavu slovem 0 do dalšího stavu.
Vidíme, že tímto slovem se dostáváme do stavu q_1.

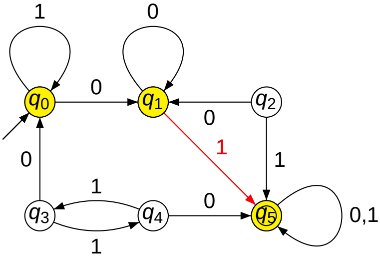


Dosažitelné stavy = [q_0, q_1,

Vyřízené stavy = [q_0,

Nedosažitelné stavy = [

Nyní ze stavu q_1 přejdeme slovem 1 do dalšího stavu.
Je zřejmé, že tímto slovem se dostáváme do stavu q_5.
Tento stav označíme jako dosažitelný. Zároveň jsme vyřídili stav q_1 a je tedy označen jako vyřízený.



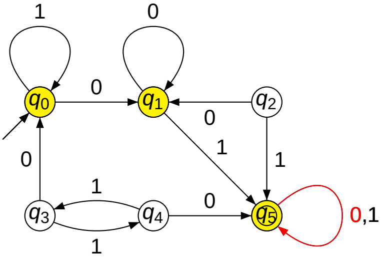
Dosažitelné stavy = [q_0, q_1, q_5

Vyřízené stavy = [q_0, q_1,

Nedosažitelné stavy = [

Eliminace nedosažitelných stavů

Nyní budeme vycházet ze stavu, který je zařazen mezi dosažitelné stavy, ale zároveň ještě není označen jako vyřízený. A to je stav q_5.
Z tohoto stavu přejdeme slovem 0 do dalšího stavu.
Z toho plyne, že tímto slovem se dostáváme do stavu q_5.



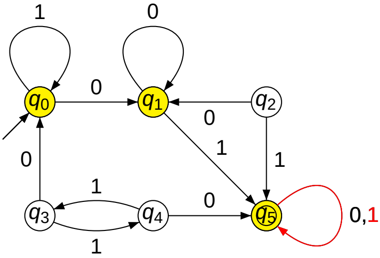
Dosažitelné stavy = [q_0, q_1, q_5

Vyřízené stavy = [q_0, q_1,

Nedosažitelné stavy = [

Eliminace nedosažitelných stavů

Nyní ze stavu q_5 přejdeme slovem 1 do dalšího stavu.
Je zřejmé, že tímto slovem se dostáváme do stavu q_5.
V tomto stavu jsme probrali všechny výstupní šipky, tudíž jej označíme jako vyřízený.



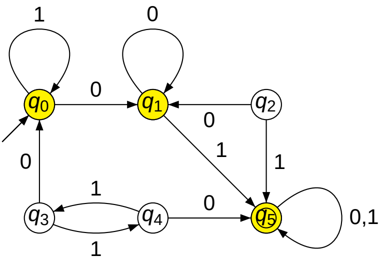
Dosažitelné stavy = [q_0, q_1, q_5

Vyřízené stavy = [q_0, q_1, q_5

Nedosažitelné stavy = [

Eliminace nedosažitelných stavů

Všechny stavy, které byly označeny jako dosažitelné, jsou nyní označeny i jako vyřízené. Tím algoritmus končí.



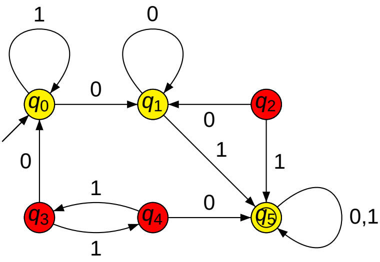
Dosažitelné stavy = [q_0, q_1, q_5]

Vyřízené stavy = [q_0, q_1, q_5]

Nedosažitelné stavy = [

Eliminace nedosažitelných stavů

Nyní stavy, které nejsou označené jako dosažitelné, označíme jako nedosažitelné a můžeme je z automatu vypustit, aniž by se změnil jazyk přijímaný tímto automatem.



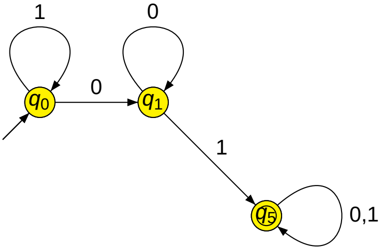
Dosažitelné stavy = [q_0, q_1, q_5]

Vyřízené stavy = [q_0, q_1, q_5]

Nedosažitelné stavy = [q_2, q_3, q_4]

Eliminace nedosažitelných stavů

Automat již neobsahuje nedosažitelné stavy.



Dosažitelné stavy = [q_0, q_1, q_5]

Vyřízené stavy = [q_0, q_1, q_5]

Nedosažitelné stavy = [q_2, q_3, q_4]

Eliminace nedosažitelných stavů

Algoritmus pro eliminaci nedosažitelných stavů

Vstup : Konečný automat $M = (Q, \Sigma, \delta, q_0, F)$
Výstup : Ekvivalentní automat M‘ bez nedosažitelných stavů

- $i = 0$
- $S_i = 0$
- repeat $S_{i+1} := S_i \cup \{q_0\} \cup \{q | \exists p \in S_i, a \in \Sigma : \delta(p, a) = q\}$
- $i := i + 1$
- until $S_i = S_{i-1}$
- $Q := S_i$
- $M‘ := (Q‘, \Sigma, \delta/Q‘, q_0, F \cap Q‘)$

Eliminace nedosažitelných stavů - Shrnutí

Definice

Nechť $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat.
Stav $q \in Q$ nazveme dosažitelný, pokud existuje
$w \in \Sigma^*$ takové, že $\hat{\delta}(q_0, w) = q$.
Stav je nedosažitelný, pokud není dosažitelný.

- Do nedosažitelných stavů nevede v grafu automatu žádná orientovaná cesta z počátečního stavu.

Eliminace nedosažitelných stavů - Shrnutí

Definice

Nechť $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat.
Stav $q \in Q$ nazveme dosažitelný, pokud existuje
$w \in \Sigma^*$ takové, že $\hat{\delta}(q_0, w) = q$.
Stav je nedosažitelný, pokud není dosažitelný.

- Do nedosažitelných stavů nevede v grafu automatu žádná orientovaná cesta z počátečního stavu.
- Nedosažitelné stavy můžeme z automatu odstranit se všemi přechody vedoucími z nich. Jazyk přijímaný automatem se nezmění.

Normovaný tvar

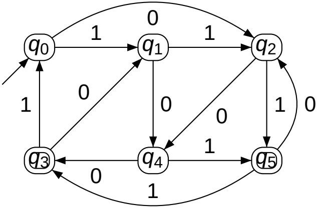
- Jeden automat lze prezentovat mnoha různými způsoby, proto nás zajímá nějaká jednoznačná prezentace.
- Automat je v normovaném tvaru, jestliže jeho stavy jsou očíslované 1,2,...,n v abecedním pořadí nejmenších slov, kterými tyto stavy dosáhneme.

Normovaný tvar

- Postup je stejný jako u hledání dosažitelných stavů. Rozdíl je v tom, že stavy neznačíme ($q_0, q_1, ...q_n$) jak byly zadány, ale značíme jej čísly (1,2,3....n).
- Postup:
  - *Počáteční stav označíme 1.*
  - *Dále např. v případě abecedy {a,b} zjistíme stav q , do něhož automat přejde ze stavu 1 symbolem a . Pokud q není označen, označíme jej 2 .*
  - *Pak zjistíme stav q , do něhož automat přejde ze stavu 1 symbolem b . Pokud stav q není dosud označen, označíme jej nejmenším dosud nepoužitým číslem.*
  - *Takto pokračujeme dále, dokud nezískáme všechny dosažitelné stavy.*

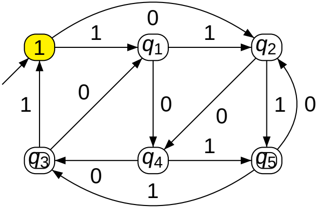
Normovaný tvar

Na jednoduchém příkladu si znázorníme postup převodu do normovaného tvaru.



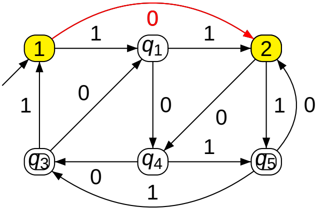
Normovaný tvar

Nejprve počáteční stav označíme číslem 1.



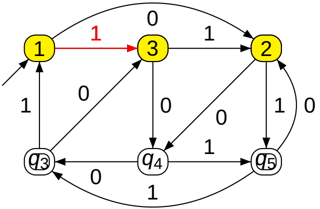
Normovaný tvar

Nyní z počátečního stavu přejdeme slovem 0 do dalšího stavu.
Pokud tento stav již není označen žádným číslem, označíme jej číslem 2.



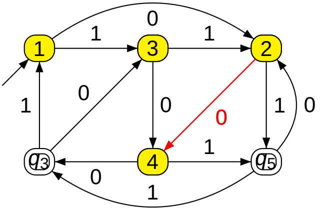
Normovaný tvar

Nyní z počátečního stavu přejdeme slovem 1 do dalšího stavu.
Vidíme, že tímto slovem se dostáváme do stavu q_1, který není označen číslem.
Označíme jej tedy číslem 3.
Z počátečního stavu jsme přešli všemi výstupními hranami do stavů, které jsou dosažitelné. Tudíž začneme procházet graf z následujícího stavu a to ze stavu označeného číslem 2.



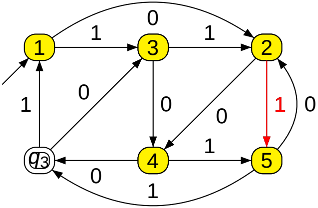
Normovaný tvar

Nyní tedy přejdeme ze stavu, který je označen číslem 2, slovem 0 do dalšího stavu. Z toho vyplývá, že tímto slovem se dostáváme do stavu q_4. Označíme jej číslem 4.



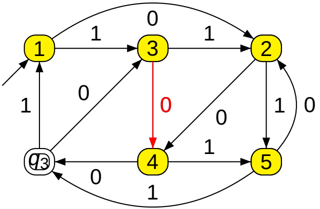
Normovaný tvar

Nyní tedy přejdeme ze stavu, který je označen číslem 2, slovem 1 do dalšího stavu. Vidíme, že tímto slovem se dostáváme do stavu q_5. Označíme jej číslem 5. Opět jsme prošli všechny výstupní hrany ze stavu označeného číslem 2. Začneme tedy procházet graf ze stavu označeného číslem 3.



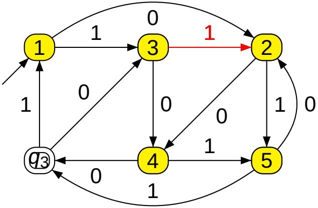
Normovaný tvar

Nyní tedy přejdeme ze stavu, který je označen číslem 3, slovem 0 do dalšího stavu. Je zřejmé, že tímto slovem se dostáváme do stavu, který je označen číslem 4.



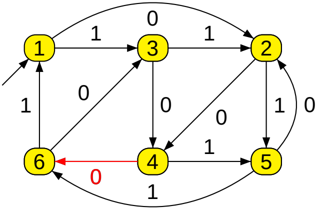
Normovaný tvar

Nyní tedy přejdeme ze stavu, který je označen číslem 3, slovem 1 do dalšího stavu. Tímto slovem se dostáváme do stavu, který je již označen číslem 2.
Opět jsme prošli všechny výstupní hrany ze stavu označeného číslem 3. Začneme tedy procházet graf ze stavu označeného číslem 4.



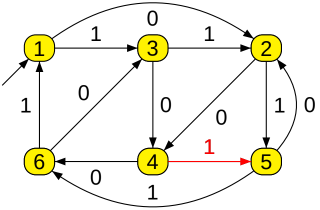
Normovaný tvar

Nyní tedy přejdeme ze stavu, který je označen číslem 4, slovem 0 do dalšího stavu. Vidíme, že tímto slovem se dostáváme do stavu q_3, který není označen, označíme ho tedy číslem 6.



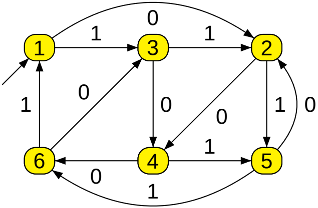
Normovaný tvar

Nyní tedy přejdeme ze stavu, který je označen číslem 4, slovem 1 do dalšího stavu. Z toho plyne, že tímto slovem se dostáváme do stavu, který není označen, označíme ho tedy číslem 5.



Normovaný tvar

Nyní vidíme, že všechny stavy jsou očíslovány $\{1,2,3...,n\}$. Automat je tedy převeden do normovaného tvaru.



Sjednocení ekvivalentních stavů

Definice

Pro každý stav q automatu $M = (Q, \Sigma, \delta, q_0, F)$ definuje $L(q) = L(M_q)$,
kde $M_q = (Q, \Sigma, \delta, q, F)$

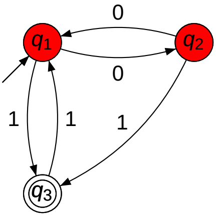
Definice

Stavy q_1, q_2 automatu $M = (Q, \Sigma, \delta, q_0, F)$ nazýváme jazykově ekvivalentní nebo zkráceně ekvivalentní, jestliže $L(q_1) = L(q_2)$

- Jsou-li dva stavy q_1, q_2 automatu ekvivalentní, můžeme jeden vypustit. Všechny šipky, které do něj směřují, musíme přesměrovat do druhého.
- Pokud byl vypouštěný stav počáteční, bude počáteční ten druhý.

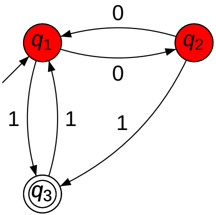
Sjednocení ekvivalentních stavů

Stavy q_1 a q_2 jsou ekvivalentní, můžeme tedy jeden z nich vypustit a všechny šipky do něj směřující přesměrovat do stavu druhého.



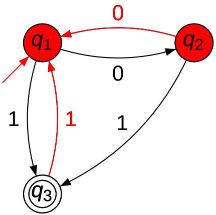
Sjednocení ekvivalentních stavů

- Vynecháme-li tedy stav q_1, který je počáteční, stane se počátečním stavem stav q_2.



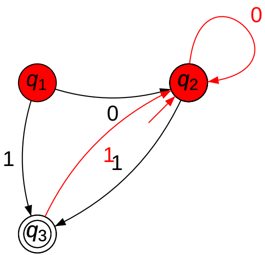
Sjednocení ekvivalentních stavů

- Vynecháme-li tedy stav q_1, který je počáteční, stane se počátečním stavem stav q_2.
- Hrany směřující do stavu q_1 přesměrujeme do stavu, který ponecháváme - tedy do q_2.



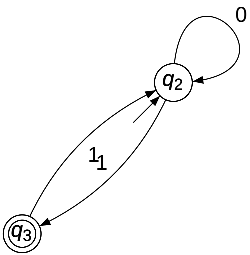
Sjednocení ekvivalentních stavů

- Vynecháme-li tedy stav q_1, který je počáteční, stane se počátečním stavem stav q_2.
- Hrany směřující do stavu q_1 přesměrujeme do stavu, který ponecháváme - tedy do q_2.



Sjednocení ekvivalentních stavů

- Vynecháme-li tedy stav q_1, který je počáteční, stane se počátečním stavem stav q_2.
- Hrany směřující do stavu q_1 přesměrujeme do stavu, který ponecháváme - tedy do q_2.
- Nyní tedy odstraníme stav q_1 a hrany z něj vycházející.



Sjednocení ekvivalentních stavů

- Dvojice vzájemně ekvivalentních stavů lze hledat rychlým algoritmem.

Sjednocení ekvivalentních stavů

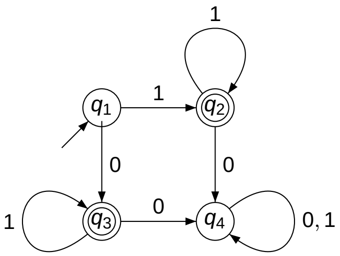
- Dvojice vzájemně ekvivalentních stavů lze hledat rychlým algoritmem.
- Postupujeme tak, že rozkládáme množinu všech stavů automatu na neekvivalentní podmnožiny. Pokračujeme v jednotlivých krocích tak dlouho, dokud ještě dochází k dalšímu rozložení.

Sjednocení ekvivalentních stavů

- Dvojice vzájemně ekvivalentních stavů lze hledat rychlým algoritmem.
- Postupujeme tak, že rozkládáme množinu všech stavů automatu na neekvivalentní podmnožiny. Pokračujeme v jednotlivých krocích tak dlouho, dokud ještě dochází k dalšímu rozložení.
- Po ukončení procedury jsou podmnožiny nerozlišitelných stavů sloučeny do jednotlivých stavů.

Sjednocení ekvivalentních stavů

Nejprve množinu všech stavů rozdělíme na dvě skupiny. První skupina bude obsahovat stavy přijímací. Druhá skupina bude obsahovat stavy nepřijímací.

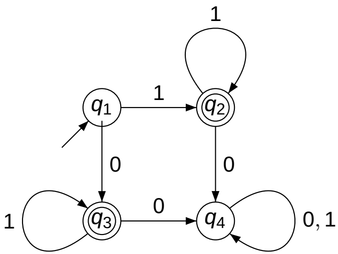


	0	1
$\rightarrow q_1$	q_3	q_2
$\leftarrow q_2$	q_4	q_2
$\leftarrow q_3$	q_4	q_3
q_4	q_4	q_4

	0	1
$I \rightarrow q_1$	q_3	q_2
q_4	q_4	q_4
$II \leftarrow q_2$	q_4	q_2
$\leftarrow q_3$	q_4	q_3

Sjednocení ekvivalentních stavů

Do tabulky si místo přechodů do konkrétních stavů vyznačíme skupinu, do které přecházíme.



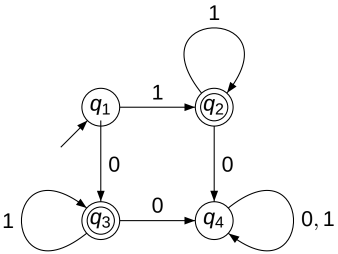
	0	1
$\rightarrow q_1$	q_3	q_2
$\leftarrow q_2$	q_4	q_2
$\leftarrow q_3$	q_4	q_3
q_4	q_4	q_4

	0	1
$I \rightarrow q_1$	q_3	q_2
q_4	q_4	q_4
$II \leftarrow q_2$	q_4	q_2
$\leftarrow q_3$	q_4	q_3

	0	1
$I \rightarrow q_1$	II	II
q_4	I	I
$II \leftarrow q_2$	I	II
$\leftarrow q_3$	I	II

Sjednocení ekvivalentních stavů

- Z tabulky vyplývá, že se skupina I rozpadá na dvě další.
- Stav q_1 se liší od q_4 , protože se dostává slovem 0 nebo 1 do skupiny, která následně přijímá slovo. Ze stavu q_4 přejdeme slovy 0 nebo 1 do skupiny, která následně nepřijímá prázdné slovo a tudíž nemohou být ekvivalentní.



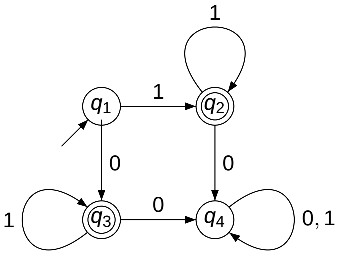
	0	1
$\rightarrow q_1$	q_3	q_2
$\leftarrow q_2$	q_4	q_2
$\leftarrow q_3$	q_4	q_3
q_4	q_4	q_4

	0	1
$I \rightarrow q_1$	q_3	q_2
q_4	q_4	q_4
$II \leftarrow q_2$	q_4	q_2
$\leftarrow q_3$	q_4	q_3

	0	1
$I \rightarrow q_1$	II	II
q_4	I	I
$II \leftarrow q_2$	I	II
$\leftarrow q_3$	I	II

Sjednocení ekvivalentních stavů

- Znovu vyplníme tabulku, protože se změnily skupiny.



	0	1
$\rightarrow q_1$	q_3	q_2
$\leftarrow q_2$	q_4	q_2
$\leftarrow q_3$	q_4	q_3
q_4	q_4	q_4

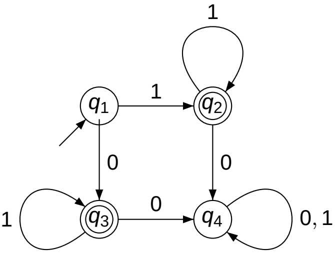
	0	1
$I \rightarrow q_1$	q_3	q_2
q_4	q_4	q_4
$II \leftarrow q_2$	q_4	q_2
$\leftarrow q_3$	q_4	q_3

	0	1
$I \rightarrow q_1$	II	II
q_4	I	I
$II \leftarrow q_2$	I	II
$\leftarrow q_3$	I	II

	0	1
$I \rightarrow q_1$	III	III
$II\ \ q_4$	II	II
$III \leftarrow q_2$	II	III
$\leftarrow q_3$	II	III

Sjednocení ekvivalentních stavů

- Nyní se už žádná skupina nerozpadá, algoritmus tedy končí.
- Stavy, které jsou v jedné skupině, jsou ekvivalentní, tudíž můžeme stavy sloučit.



	0	1
$\rightarrow q_1$	q_3	q_2
$\leftarrow q_2$	q_4	q_2
$\leftarrow q_3$	q_4	q_3
q_4	q_4	q_4

	0	1
$I \rightarrow q_1$	q_3	q_2
q_4	q_4	q_4
$II \leftarrow q_2$	q_4	q_2
$\leftarrow q_3$	q_4	q_3

	0	1
$I \rightarrow q_1$	II	II
q_4	I	I
$II \leftarrow q_2$	I	II
$\leftarrow q_3$	I	II

	0	1
$I \rightarrow q_1$	III	III
$II \;\; q_4$	II	II
$III \leftarrow q_2$	II	III
$\leftarrow q_3$	II	III

	0	1
$\rightarrow I$	III	III
II	II	II
$\leftarrow III$	II	III

Algoritmus pro sjednocení ekvivalentních stavů

Vstup : Konečný automat $M = (Q, \Sigma, \delta, q_0, F)$ bez nedosažitelných stavů s totální přechodovou funkcí.
Výstup : Redukt $M/_{\equiv}$

- $i := 0$
- $\equiv_0 := (p, q) | p \in F \Leftrightarrow p \in F$
- **repeat**
- $\equiv_{i+1} := \{(p, q) | p \equiv_i q \wedge \exists a \in \Sigma : \delta(p, a) \equiv_i \delta(q, a)\}$
- $i := i + 1$
- **until** $\equiv_i = \equiv_{i-1}$
- $\equiv := \equiv_i$
- $M/ \equiv := (Q/ \equiv, \Sigma, \eta, [q_0], F/ \equiv)$

Sjednocení ekvivalentních stavů - Shrnutí

- Nechť $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat bez nedosažitelných stavů.

Sjednocení ekvivalentních stavů - Shrnutí

- Nechť $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat bez nedosažitelných stavů.

Definice

Stavy p, q nazveme jazykově ekvivalentní, psáno $p \equiv q$, pokud $(p \equiv q \Leftrightarrow \forall x \in \Sigma^* : (\hat{\delta}(p, x) \in F \Leftrightarrow \hat{\delta}(q, x) \in F))$

Sjednocení ekvivalentních stavů - Shrnutí

Definice

Reduktem automatu M nazveme konečný automat $M/_{\equiv} = (Q/_{\equiv}, \Sigma, \eta, [q_0], F/_{\equiv},)$ tj. automat, kde

Sjednocení ekvivalentních stavů - Shrnutí

Definice

Reduktem automatu M nazveme konečný automat $M/_{\equiv} = (Q/_{\equiv}, \Sigma, \eta, [q_0], F/_{\equiv},)$ tj. automat, kde

- Stavy jsou třídy rozkladu $Q/_{\equiv}$

Definice

Reduktem automatu M nazveme konečný automat $M/_{\equiv} = (Q/_{\equiv}, \Sigma, \eta, [q_0], F/_{\equiv},)$ tj. automat, kde

- Stavy jsou třídy rozkladu $Q/_{\equiv}$
- Přechodová funkce η je nejmenší funkce splňující: $\forall (p, q \in Q), \forall a \in \Sigma : \delta(q, a) = p \Rightarrow \eta([q], a) = [p]$

Definice

Reduktem automatu M nazveme konečný automat $M/_{\equiv} = (Q/_{\equiv}, \Sigma, \eta, [q_0], F/_{\equiv},)$ tj. automat, kde

- Stavy jsou třídy rozkladu $Q/_{\equiv}$
- Přechodová funkce η je nejmenší funkce splňující: $\forall (p, q \in Q), \forall a \in \Sigma : \delta(q, a) = p \Rightarrow \eta([q], a) = [p]$
- Stavy jsou třídy rozkladu $Q/_{\equiv}$

Sjednocení ekvivalentních stavů - Shrnutí

Definice

Reduktem automatu M nazveme konečný automat $M/_{\equiv} = (Q/_{\equiv}, \Sigma, \eta, [q_0], F/_{\equiv},)$ tj. automat, kde

- Stavy jsou třídy rozkladu $Q/_{\equiv}$
- Přechodová funkce η je nejmenší funkce splňující: $\forall (p, q \in Q), \forall a \in \Sigma : \delta(q, a) = p \Rightarrow \eta([q], a) = [p]$
- Stavy jsou třídy rozkladu $Q/_{\equiv}$
- Počáteční stav je třída rozkladu $Q/_{\equiv}$ obsahující q_0.

Sjednocení ekvivalentních stavů - Shrnutí

Definice

Reduktem automatu M nazveme konečný automat $M/_{\equiv} = (Q/_{\equiv}, \Sigma, \eta, [q_0], F/_{\equiv},)$ tj. automat, kde

- Stavy jsou třídy rozkladu $Q/_{\equiv}$
- Přechodová funkce η je nejmenší funkce splňující: $\forall(p, q \in Q), \forall a \in \Sigma : \delta(q, a) = p \Rightarrow \eta([q], a) = [p]$
- Stavy jsou třídy rozkladu $Q/_{\equiv}$
- Počáteční stav je třída rozkladu $Q/_{\equiv}$ obsahující q_0.
- Koncové stavy jsou právě ty třídy rozkladu $Q/_{\equiv}$, které obsahují alespoň jeden koncový stav.

Věta o isomorfismu

- Pro libovolné dva redukované konečné automaty jsou následující tvrzení ekvivalentní:

Věta o isomorfismu

- Pro libovolné dva redukované konečné automaty jsou následující tvrzení ekvivalentní:
  - automaty jsou ekvivalentní

Věta o isomorfismu

- Pro libovolné dva redukované konečné automaty jsou následující tvrzení ekvivalentní:
  - automaty jsou ekvivalentní
  - automaty jsou isomorfní

Věta o isomorfismu

- Pro libovolné dva redukované konečné automaty jsou následující tvrzení ekvivalentní:
  - automaty jsou ekvivalentní
  - automaty jsou isomorfní

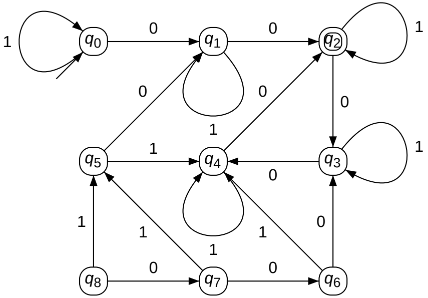
Důsledky

Dva redukty libovolných dvou ekvivalentních konečných automatů se shodují až na isomorfismus.
Pro každý KA je jeho redukt určen až na isomorfismus jednoznačně.

Minimalizace KA - Příklad

A nyní si vše předešlé názorně ukážeme na jednoduchém příkladu.

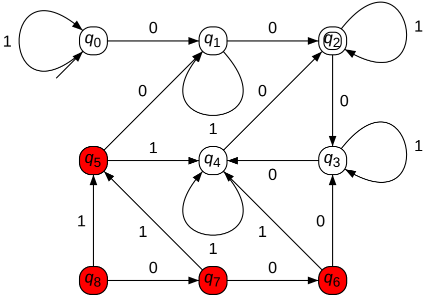
Minimalizace KA - Příklad



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

Nyní začneme minimalizaci.

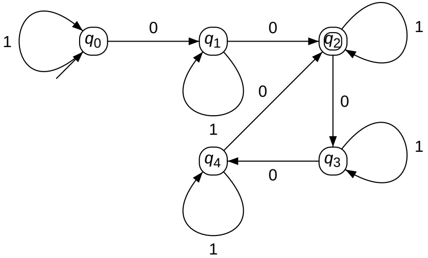
Minimalizace KA - Příklad



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

V prvním kroku odstraníme nedosažitelné stavy pomocí algoritmu, který byl zmíněn výše.
Zjistíme, že stavy q_5, q_6, q_7, q_8 jsou nedosažitelné a můžeme je vypustit.

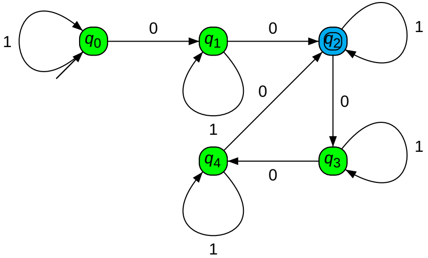
Minimalizace KA - Příklad



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

Nyní máme automat již bez nedosažitelných stavů.
Můžeme tedy nalézt množiny vzájemně ekvivalentních stavů.

Minimalizace KA - Příklad

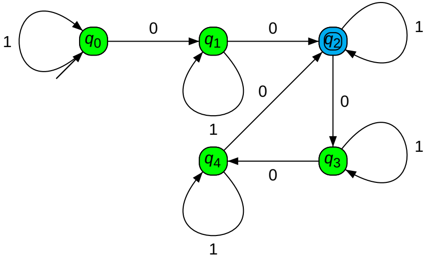


	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

$I : (q_0, q_1, q_3, q_4)$,$II : (q_2)$

Stavy automatu rozdělíme na dvě množiny, jedna množina $I = (q_0, q_1, q_3, q_4)$ obsahuje stavy, které nejsou přijímací.
Množina druhá $II = (q_2)$ obsahuje stavy přijímací.

Minimalizace KA - Příklad



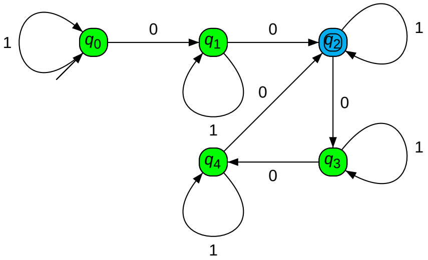
	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

$I : (q_0, q_1, q_3, q_4)$, $II : (q_2)$

	0	1
q_0	I	I
q_1	II	I
q_3	I	I
q_4	II	I
q_2	I	II

Nyní vyplníme přechodovou tabulku symboly množiny ekvivalence.

Minimalizace KA - Příklad

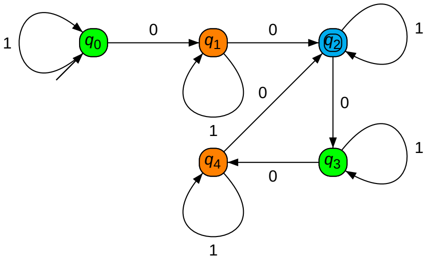


	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

$I : (q_0, q_1, q_3, q_4)$,$II : (q_2)$

	0	1
q_0	I	I
q_1	II	I
q_3	I	I
q_4	II	I
q_2	I	II

Z přechodové tabulky vyplývá, že se skupina $I = (q_0, a_1, q_3, q_4)$ rozkládá na dvě podmnožiny 1-ekvivalentních stavů a to na (q_0, q_3) a (q_1, q_4)



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

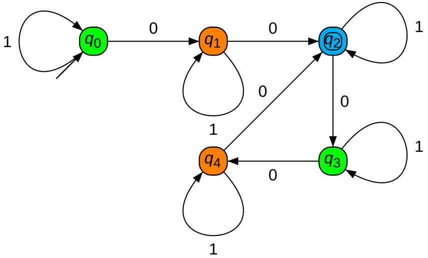
$I : (q_0, q_1, q_3, q_4)$,$II : (q_2)$

	0	1
q_0	I	I
q_1	II	I
q_3	I	I
q_4	II	I
q_2	I	II

$I : (q_0, q_3)$,$II : (q_1, q_4)$,$III : (q_2)$

Nyní máme tři množiny 1-ekvivalentních stavů.
Stavy v těchto množinách jsou vzájemně rozlišitelné slovy délky nejvýše 1.

Minimalizace KA - Příklad



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

$I : (q_0, q_1, q_3, q_4)$,$II : (q_2)$

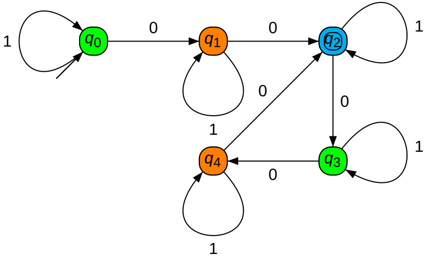
	0	1
q_0	I	I
q_1	II	I
q_3	I	I
q_4	II	I
q_2	I	II

$I : (q_0, q_3)$,$II : (q_1, q_4)$,$III : (q_2)$

	0	1
q_0	II	I
q_3	II	I
q_1	III	II
q_4	III	II
q_2	I	III

Nyní vyplníme přechodovou tabulku symboly množiny ekvivalence.

Minimalizace KA - Příklad



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

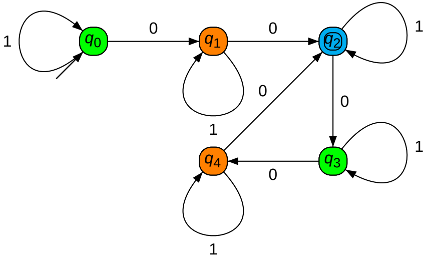
$I : (q_0, q_1, q_3, q_4)$,$II : (q_2)$

	0	1
q_0	I	I
q_1	II	I
q_3	I	I
q_4	II	I
q_2	I	II

$I : (q_0, q_3)$,$II : (q_1, q_4)$,$III : (q_2)$

	0	1
q_0	II	I
q_3	II	I
q_1	III	II
q_4	III	II
q_2	I	III

Žádná z těchto množin se již dále nerozkládá.
Tj. stavy v nich jsou vzájemně 2-ekvivalentní.



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

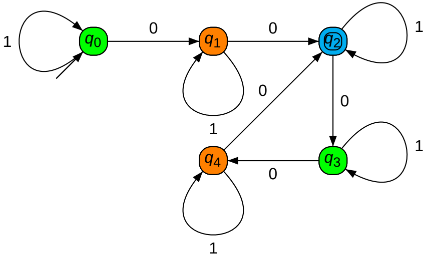
$I : (q_0, q_3), II : (q_1, q_4), III : (q_2)$

	0	1
q_0	II	I
q_3	II	I
q_1	III	II
q_4	III	II
q_2	I	III

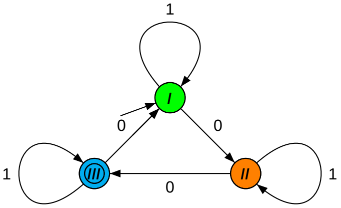
	0	1
$\rightarrow I$	II	I
II	III	II
$\leftarrow III$	I	III

Nyní každou množinu ekvivalentních stavů nahradíme stavem jediným.

Minimalizace KA - Příklad



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4

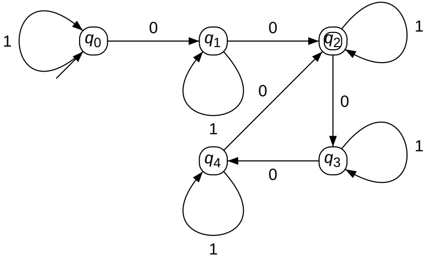


$I : (q_0, q_3), II : (q_1, q_4), III : (q_2)$

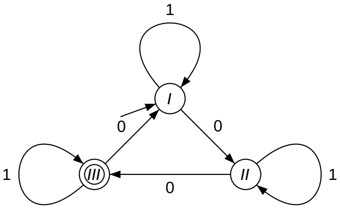
	0	1
$\rightarrow I$	II	I
II	III	II
$\leftarrow III$	I	III

A nyní vytvoříme automat podle přechodové tabulky.

Minimalizace KA - Příklad



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_1
$\leftarrow q_2$	q_3	q_2
q_3	q_4	q_3
q_4	q_2	q_4



	0	1
$\rightarrow I$	II	I
II	III	II
$\leftarrow III$	I	III

Daný minimalizovaný automat je jediný až na isomorfismus, tzn. různé pojmenování stavů. Toto můžeme odstranit převodem do normovaného tvaru.