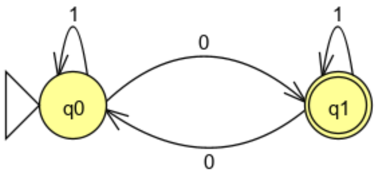
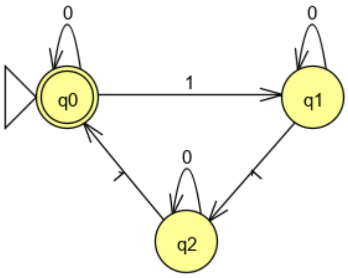
Simulácie

Máme dva konečné automaty, ktoré pracujú nad abecedou {0,1}:



A1



A2

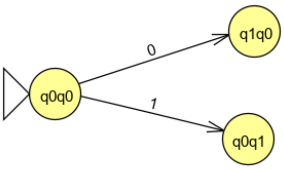
Aké slová rozpoznávajú?

Chceme navrhnúť nový automat A, ktorý bude paralelne simulovať oba automaty A_1 a A_2, krok po kroku t.j. jeden krok výpočtu v novom automate A bude simulovať jeden krok výpočtu v A_1 aj v A_2.

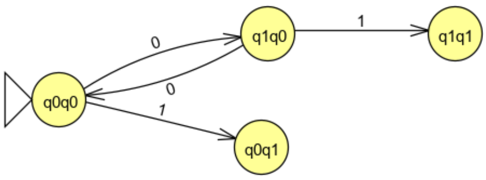
A_1 sa z počiatočného stavu po prečítaní 0 dostane do q_1, A_2 sa z počiatočného stavu po prečítaní 0 dostane do q_1. Ako tento krok oboch automatov bude simulovať automat A? Teda kam sa automat A dostane zo svojho z počiatočného stavu po prečítaní symbolu 0? A ako označíme stavy automatu A?

V stave automatu A si potrebujeme pamätať stavy oboch simulovaných automatov, najjednoduchšie je označovať stavy automatu A ako usporiadané dvojice, kde prvý z dvojice je stav automatu A_1 a druhý z dvojice je stav automatu A_2. Počiatočný stav automatu A by sme teda mohli označiť (q_0, q_0).

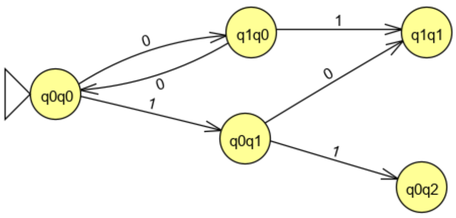
Z počiatočného stavu (q_0, q_0) po prečítaní 0 sa automat A dostane do stavu (q_1, q_0).
Kam sa dostane automat A zo stavu (q_0, q_0) po prečítaní 1? do (q_0, q_1)



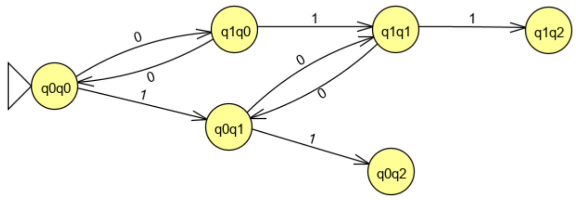
Kam sa dostane automat A zo stavu (q_1, q_0) po prečítaní 0? do (q_0, q_0)
Kam sa dostane automat A zo stavu (q_1, q_0) po prečítaní 1? do (q_1, q_1)



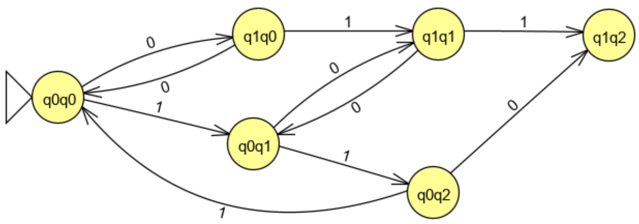
Kam sa dostane automat A zo stavu (q_0, q_1) po prečítaní 0? do (q_1, q_1)
Kam sa dostane automat A zo stavu (q_0, q_1) po prečítaní 1? do (q_0, q_2)



Kam sa dostane automat A zo stavu (q_1, q_1) po prečítaní 0? do (q_0, q_1)
Kam sa dostane automat A zo stavu (q_1, q_1) po prečítaní 1? do (q_1, q_2)

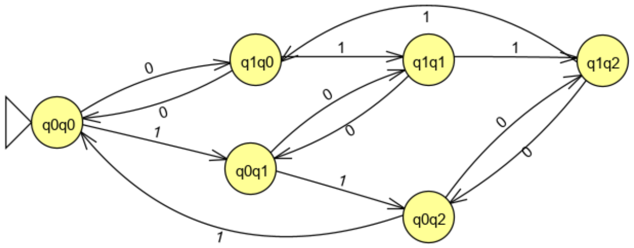


Kam sa dostane automat A zo stavu (q_0, q_2) po prečítaní 0? do (q_1, q_2)
Kam sa dostane automat A zo stavu (q_0, q_2) po prečítaní 1? do (q_0, q_0)



Kam sa dostane automat A zo stavu (q_1, q_2) po prečítaní 0? do (q_0, q_2)
Kam sa dostane automat A zo stavu (q_1, q_2) po prečítaní 1? do (q_1, q_0)

Toto je teda výsledný obrázok automatu A, ktorý simuluje oba automaty.



Teraz máme diagram automatu, ktorý simuluje automaty A_1 a A_2. Nemá zatiaľ určené žiadne akceptujúce stavy.

Chceme, aby výsledný automat rozpoznával slová, ktoré obsahujú **nepárny počet núl a zároveň počet jednotiek deliteľný troma**.
Čo budú akceptujúce stavy nového automatu? *$F = \{(q_1,q_0)\}$*
Tento automat rozpoznáva jazyk $L(A_1) \cap L(A_2)$.

Chceme, aby výsledný automat rozpoznával slová, ktoré obsahujú **nepárny počet núl ALEBO počet jednotiek deliteľný troma**.
Čo budú teraz akceptujúce stavy nového automatu? *$F = \{(q_1, q_0),(q_1, q_1), (q_1, q_2),(q_0, q_0)\}$*
Tento automat rozpoznáva jazyk $L(A_1) \cup L(A_2)$.

Aké akceptujúce stavy by mal mať automat, ktorý by rozpoznával slová majúce **nepárny počet núl a počet jednotiek nedeliteľný troma**?
Ako by sme vedeli pomocou množinových operácií a jazykov $L(A_1)$, $L(A_2)$ vyjadriť tento jazyk?

Aké akceptujúce stavy by mal mať automat, ktorý by rozpoznával slová majúce **počet jednotiek deliteľný tromi a párny počet núl**? Ako by sme vedeli pomocou množinových operácií a jazykov $L(A_1)$, $L(A_2)$ vyjadriť tento jazyk?

Lema 3.10 (Simulácia) Nech Σ je abeceda a $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ a $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ sú konečné automaty. Pre operáciu $\Theta \in \{\cup, \cap, -\}$ existuje KA M taký, že $L(M) = L(M_1)\ \Theta\ L(M_2)$.

// Ak máme dva deterministické konečné automaty s rovnakou abecedou, potom vieme skonštruovať taký deterministický konečný automat, ktorý bude naraz simulovať výpočty oboch z nich.

Dôkaz

Pozostáva z dvoch krokov:

1. Skonštruujeme taký automat M, ktorý bude simulovať M_1 aj M_2.

$M = (Q, \Sigma, \delta, q_0, F)$, kde:

(i) $Q = Q_1 \times Q_2$, //stavy nového automatu budú usporiadané dvojice, kde prvý z dvojice bude stav z 1. automatu a druhý z dvojice bude stav 2. automatu

(ii) $q_0 = (q_{01}, q_{02})$, // počiatočný stav nového automatu je dvojica, ktorú tvoria počiatočný stav 1. automatu a počiatočný stav 2. automatu

(iii) pre všetky $q \in Q_1$, $p \in Q_2$ a $a \in \Sigma$: $\delta((q, p), a) = (\delta_1(q, a), \delta_2(q, a))$
//V novom automate sa zo stavu (q, p) po prečítaní symbolu a dostaneme do stavu, ktorý bude tvoriť dvojica: stav, do ktorého sa zo stavu p po prečítaní symbolu a dostane 1. automat a stav, do ktorého sa zo stavu q po prečítaní symbolu a dostane 2. automat.

(iv) ak $\Theta = \cup$, $F = F_1 \times Q_2 \cup Q_1 \times F_2$,
ak $\Theta = \cap$, $F = F_1 \times F_2$,
ak $\Theta = -$, $F = F_1 \times (Q_2 - F_2)$.

2. Ukážeme, že takto skonštruovaný automat M spĺňa, že $L(M) = L(M_1) \odot L(M_2)$.

Musíme dokázať, že $\forall z \in \Sigma^*$: $\boldsymbol{\hat{\delta}((q_{01},q_{02}),z) = (\hat{\delta}_1(q_{01},z),\hat{\delta}_2(q_{02},z))}$ (1)

Dokážeme to matematickou indukciou vzhľadom na dĺžku slova z:

- Báza:
  $z| = 0$, $z = \lambda$;
  $\hat{\delta}((q_{01},q_{02}),\lambda) \overset{\text{podľa df } \delta\hat{}}{=} (q_{01},q_{02}) \overset{\text{podľa df } \delta\hat{}}{=} (\hat{\delta}_1(q_{01},\lambda),\hat{\delta}_2(q_{02},\lambda))$
- Indukčný krok: Predpokladajme, že (1) platí pre všetky slová $w \in \Sigma^n$ (slová dĺžky n). Ukážeme, že (1) platí aj pre $z \in \Sigma^{n+1}$.
  Nech $z = wa$, $w \in \Sigma^n$, $a \in \Sigma$.

$\boldsymbol{\hat{\delta}((q_{01},q_{02}),z)}$	$= \hat{\delta}((q_{01},q_{02}),wa)$	// z = wa
	$= \delta(\hat{\delta}((q_{01},q_{02}),w), a)$	//podľa definície $\hat{\delta}$
	$= \delta((\hat{\delta}_1(q_{01},w),\hat{\delta}_2(q_{02},w)), a)$	// podľa indukčného predpokladu
	$= (\delta_1(\hat{\delta}_1(q_{01},w),a), \delta_2(\hat{\delta}_2(q_{02},w), a))$	// podľa definície δ automatu M (iii)
	$= (\hat{\delta}_1(q_{01},wa),\hat{\delta}_2(q_{02},wa))$	//podľa definície $\hat{\delta}_1$, $\hat{\delta}_2$
	$\boldsymbol{= (\hat{\delta}_1(q_{01},z),\hat{\delta}_2(q_{02},z))}$	// wa =z

Použitá literatúra
Hromkovič Juraj: Teoretical Computer Science