

Ekvivalentne: $G'_1 = (\{S\}, \{f, i, n, t\}, P, S)$.

$S \rightarrow if$

$S \rightarrow int$

$L(G'_1) = \{if, int\} = L(G_1)$.



Príklad: $G_2 = (\{S, A, B\}, \{+, -, 0, 1, 2, \dots, 9\}, P, S)$.

$$S \rightarrow +A \mid -A \mid 1B \mid 2B \mid \dots \mid 0 \mid 1 \mid \dots \mid 9$$

$$A \rightarrow 1B \mid 2B \mid \dots \mid 0 \mid 1 \mid \dots \mid 9$$

$$B \rightarrow 0B \mid 1B \mid \dots \mid 0 \mid 1 \mid \dots \mid 9$$

$L(G_2)$ tvoria všetky celočíselné konštanty (prípadne so znamienkom) bez bezvýznamných núl zľava.



Akceptačná špecifikácia jazyka

- Alternatívne môžeme uvažovať nasledovný spôsob špecifikácie jazyka.
- Uvažujme nejaké výpočtové zariadenie, ktorého vstup môže byť ľubovoľný reťazec.
- Toto zariadenie tento reťazec spracuje (napr. prečíta každý jeho symbol) a v konečnom čase jeho výstup predstavuje jedna z dvoch hodnôt: **akceptujem** / **neakceptujem**.
- Na základe tohto výstupu zariadenia potom hovoríme, že **akceptuje** / **neakceptuje** vstupný reťazec.
- Takéto zariadenie teda má nejakú množinu reťazcov, pre ktoré vráti odpoveď **akceptujem** a pre ostatné reťazce vracia odpoveď **neakceptujem**.



Konečný automat (ďalej len KA)

- Vstupom KA je páska so vstupným slovom, zloženého zo symbolov Σ .
- Automat začína svoju činnosť v stave q_0 .
- Automat vždy "vidí" jeden symbol zo vstupnej pásky. Ten prečíta svojou **čítacou hlavou** a na základe aktuálneho stavu a hodnoty symbolu podľa prechodovej funkcie prejde do nasledujúceho stavu.
- Následne sa posúva čítacia hlava na ďalší symbol vstupnej pásky. Posun pri čítaní späť na predchádzajúce symboly vstupného slova **nie je možný**.
- Ak automat po prečítaní celého vstupného slova skončí v akceptačnom stave, vstupné slovo **akceptuje**. Inak slovo **neakceptuje**.



- Na popis činnosti, resp. spracovania nejakého vstupu konečným automatom používame tzv. konfigurácie.
- Konfigurácia KA je spôsob zachytenia informácie o tom, v akom **stave sa aktuálne** konečný automat nachádza a aká je **nespracovaná** (neprečítaná) **časť vstupného reťazca**.



Konfigurácia KA

Definícia

Nech $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat. Potom:

- usporiadanú dvojicu $(q, w) \in Q \times \Sigma^*$ nazývame **konfiguráciou konečného automatu M** .
- Konfiguráciu (q_0, w) , kde w je celý vstupný reťazec, nazývame **začiatočnou konfiguráciou automatu M** ,
- konfiguráciu (q, ϵ) , kde $q \in F$, nazývame **akceptujúcou konfiguráciou automatu M** .

Konfiguráciu teda definuje **aktuálny stav** automatu a **neprečítaná** časť vstupného slova.



Prechody KA

Definícia

Nech $M = (Q, \Sigma, \delta, q_0, F)$ je deterministický konečný automat. Potom nad množinou konfigurácií $Q \times \Sigma^*$ definujeme **reláciu prechodu** (krok výpočtu) \vdash takto:

Nech $q, p \in Q, w \in \Sigma^*, a \in \Sigma$. Potom $(q, aw) \vdash (p, w)$ práve vtedy, keď $\delta(q, a) = p$.

T.j. ak je automat v stave q , na vstupe je neprečítaná časť slova aw , t.j. číta prvý symbol zľava a a jeho prechodová funkcia $\delta(q, a) = p$, potom po prečítaní tohto symbolu prejde do stavu p a na vstupe ostane neprečítaná časť w .



Prechody KA

Môžeme použiť aj symbol \vdash_M , ak chceme zdôrazniť, že relácia prechodu sa týka daného konečného automatu M - ak máme napríklad viacero automatov a potrebujeme ich rozlíšiť.



Výpočet KA

Definícia

Nech $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat. Potom postupnosť konfigurácií $(q_0, w_0), (q_1, w_1), \dots, (q_k, w_k)$ takých, že

$$(q_0, w_0) \vdash (q_1, w_1) \vdash \dots \vdash (q_k, w_k),$$

$q_i \in Q, w_i \in \Sigma^, i = 0, 1, \dots, k$ sa nazýva **výpočet konečného automatu M z (q_0, w_0) do (q_k, w_k)** a vyjadrovať ho budeme v tvare $(q_0, w_0) \vdash^k (q_k, w_k)$.*

*Číslo k predstavuje **počet krokov výpočtu**.*



Výpočet KA - nešpecifikovaný počet krokov

Definícia

Nech $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat. Potom:

- **tranzitívnym uzáverom relácie krok výpočtu** nazývame reláciu \vdash^+ definovanú na množine konfigurácií nasledovne: $(q, w) \vdash^+ (p, x)$ práve vtedy, keď existuje $n \geq 1$ také, že $(q, w) \vdash^n (p, x)$ a
- **reflexívnym a tranzitívnym uzáverom relácie krok výpočtu** nazývame reláciu \vdash^* definovanú na množine konfigurácií nasledovne: $(q, w) \vdash^* (p, x)$ práve vtedy, keď existuje $n \geq 0$ také, že $(q, w) \vdash^n (p, x)$,

pričom $p, q \in Q$ a $w, x \in \Sigma^*$.



T.j. pomocou relácií \vdash^+ a \vdash^* vieme vyjadriť výpočet automatu začínajúci v konfigurácii (q, w) a končiaci v (p, x) s nešpecifikovaným počtom krokov.

T.j. že existuje spôsob, akým sa automat dostane zo stavu q a neprečítaným vstupným slovom w do stavu p a neprečítaným vstupným slovom x .

Je zrejmé, že x je prípona slova w .



Jazyk KA

Definícia

*Nech $M = (Q, \Sigma, \delta, q_0, F)$ je konečný automat. Potom jazyk $L(M)$ **rozpoznávaný(akceptovaný)** konečným automatom M je množina:*

$$L(M) = \{w \mid (q_0, w) \vdash^* (q, \varepsilon), w \in \Sigma^*, q \in F\}.$$

T.j. jazyk KA je množina slov, pre ktoré sa automat po ich úplnom spracovaní dostane z počiatočného stavu do jedného z akceptačných stavov.



Príklad: Je daný DKA $M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$, pričom prechodová funkcia δ :

δ	0	1
q_0	q_1	q_2
q_1	q_1	q_2
q_2	q_2	q_2

Výpočet pre **neakceptovaný** reťazec 001:

$$(q_0, 001) \vdash (q_1, 01) \vdash (q_1, 1) \vdash (q_2, \varepsilon); q_2 \notin F$$

Výpočet pre **akceptovaný** reťazec 000:

$$(q_0, 000) \vdash (q_1, 00) \vdash (q_1, 0) \vdash (q_1, \varepsilon); q_1 \in F$$

$L(M)$ sú reťazce zložené zo samých núl dĺžky aspoň 1.

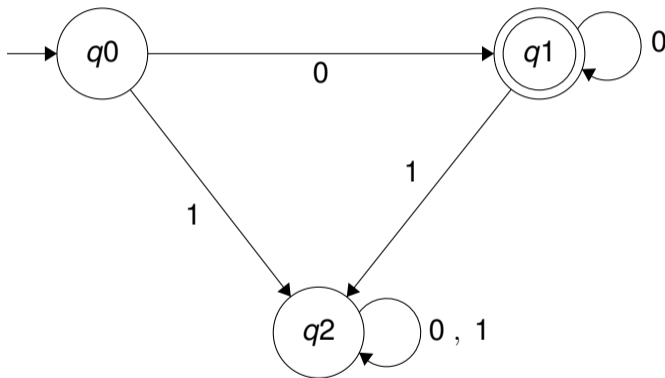


Grafická reprezentácia KA

KA môžeme reprezentovať **prechodovým diagramom** - orientovaný graf, ktorého vrcholy sú stavy, hrany sú ohodnotené symbolmi vstupnej abecedy a reprezentujú prechodovú funkciu, počiatočný stav je označený vstupujúcou hranou a akceptačné stavy sú označené dvojitým krúžkom.



Grafická reprezentácia KA



Neúplné DKA

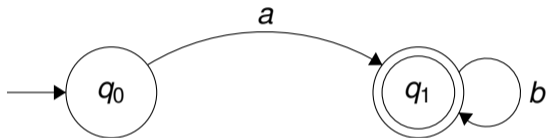
Niekedy je možné, že prechodová funkcia δ nie je úplná, t.j. nie je definovaná pre všetky kombinácie stavov a symbolov vstupnej abecedy.

V grafe by to znamenalo, že z vrcholov by vždy nevychádzalo toľko hrán, koľko je symbolov vstupnej abecedy.



Neúplné DKA

V uvedenom automate nie sú definované prechody: $\delta(q_0, b)$ a $\delta(q_1, a)$:



Reťazce, pre ktoré sa výpočet zasekne v nejakom stave a neboli dočítané do konca, sú **neakceptované**. Napríklad:

- aa neakceptované, pretože: $(q_0, aa) \vdash (q_1, a) \vdash$ ZÁSEK!
- bb neakceptované, pretože $(q_0, bb) \vdash$ ZÁSEK!

Neúplné DKA

Avšak, ak potrebujeme mať úplný DKA, t.j. definovanú prechodovú funkciu pre všetky symboly vstupnej abecedy a pre všetky stavy, nie je problém automat rozšíriť o tzv. **pascu** - nový neakceptujúci stav, ktorý zachytí tie symboly, pre ktoré pôvodne nebola prechodová funkcia definovaná.



Kompletizácia DKA

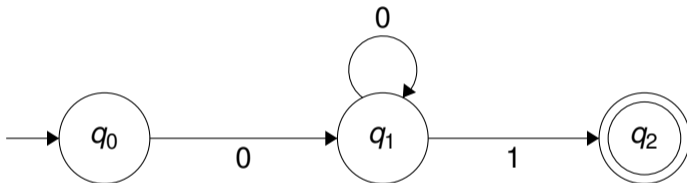
Veta

Nech $M = (Q, \Sigma, \delta, q_0, F)$ je DKA. Potom k nemu možno zostrojiť DKA $M_K = (Q_K, \Sigma, \delta_K, q_0, F)$ taký, že $(q_0, w) \vdash_{M_K}^ (q, \varepsilon)$ pre každé $w \in \Sigma^*$ a navyše $L(M) = L(M_K)$.*

Inými slovami, vždy sa dá doplniť prechodová funkcia na úplnú bez toho, aby sa zmenil jazyk, ktorý automat akceptuje.



Príklad neúplného DKA:



Príklad: Uvažujme NKA $M = (\{q_0, q_1, q_2, q_f\}, \{+, -, 0, 1, 2, \dots, 9\}, \delta, q_0, \{q_f\})$, kde prechodová funkcia δ :

δ	ε	$+, -$	0	$1, 2, \dots, 9$
q_0	$\{q_1\}$	$\{q_1\}$	\emptyset	\emptyset
q_1	\emptyset	\emptyset	$\{q_f\}$	$\{q_2, q_f\}$
q_2	\emptyset	\emptyset	$\{q_2, q_f\}$	$\{q_2, q_f\}$
q_f	\emptyset	\emptyset	\emptyset	\emptyset

Symbolom , (čiarka) sú oddelené alternatívne symboly pre jednoduchší zápis.

Príklad, pokr. Vstup: +1:

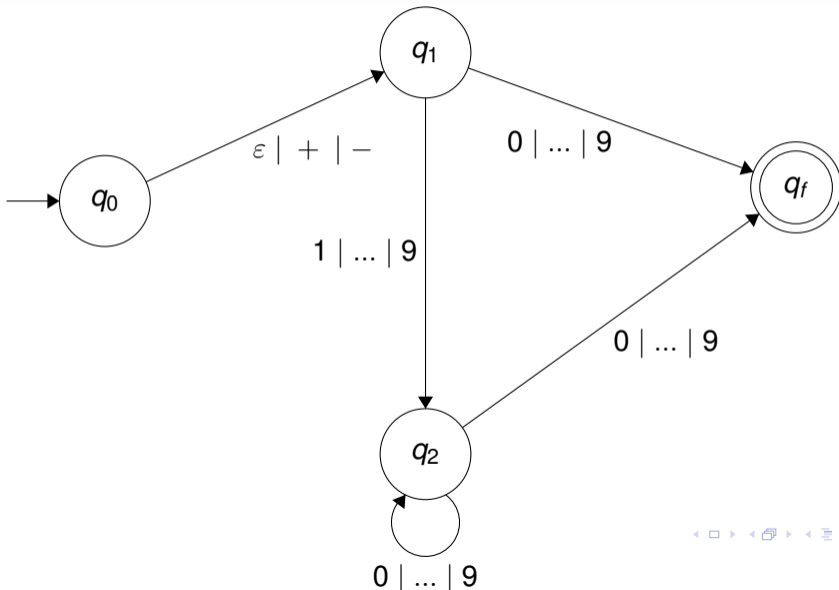
$(q_0, +1) \vdash (q_1, +1)$ (použilo sa $\delta(q_0, \varepsilon) = q_1$)

$(q_0, +1) \vdash (q_1, 1) \vdash (q_2, \varepsilon)$

$(q_0, +1) \vdash (q_1, 1) \vdash (q_f, \varepsilon)$



Príklad, pokr.



Vzťah medzi DKA a NKA

- Je jasné, že keďže NKA sú zovšeobecnením DKA, tak každý jazyk, ktorý je akceptovateľný deterministickým konečným automatom, je aj akceptovateľný nedeterministickým konečným automatom.
- Čo je prekvapivejšie je fakt, že to platí aj naopak, t.j. ku každému jazyku, ku ktorému sa dá zostrojiť NKA, ktorý ho akceptuje, sa dá zostrojiť aj DKA, ktorý ho akceptuje.
- NKA a DKA majú teda **rovnakú silu/akceptačnú schopnosť**.



Ekvivalencia NKA a DKA

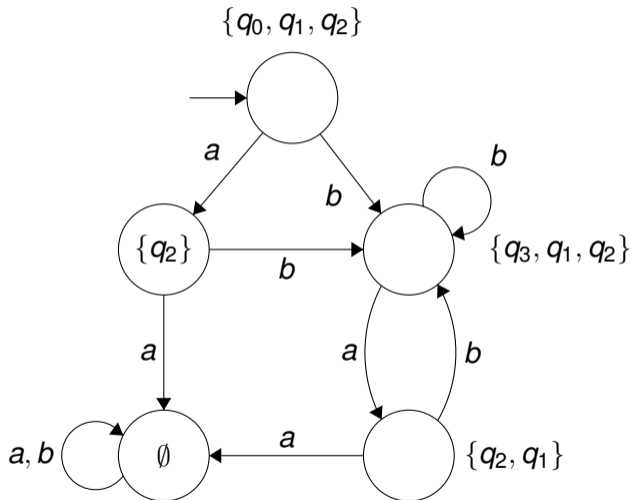
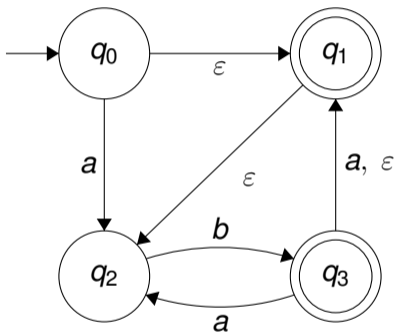
Veta

*Nech L je jazyk akceptovaný nejakým NKA. Potom **existuje** DKA, ktorý akceptuje ten istý jazyk L .*

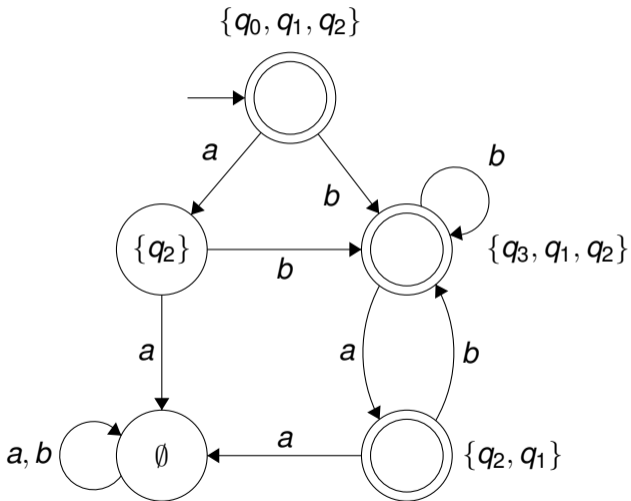
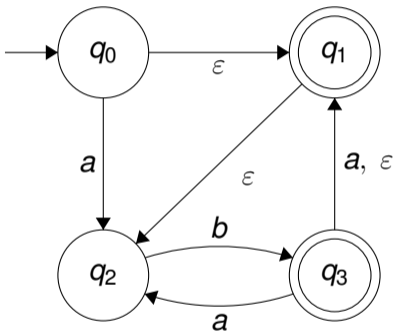
Idea zostrojenia ekvivalentného DKA je v tom, že nový DKA „sleduje“ všetky výpočtové cesty, ktorými mohol ísť pôvodný NKA.



Zo stavu $\{q_2, q_1\}$



Akceptačné stavy



Operácia $CLOSURE_{\epsilon}$

Pri konštrukcii DKA si zadefinujeme operáciu $CLOSURE_{\epsilon}$, ktorá pre množinu stavov NKA vráti množinu stavov uzavretú na kroky na ϵ .

Vstup: Prechodová funkcia δ NKA M , množina stavov $S \subseteq Q$.

Výstup: Množina stavov S uzavretá na kroky na ϵ .

1: **opakuj**

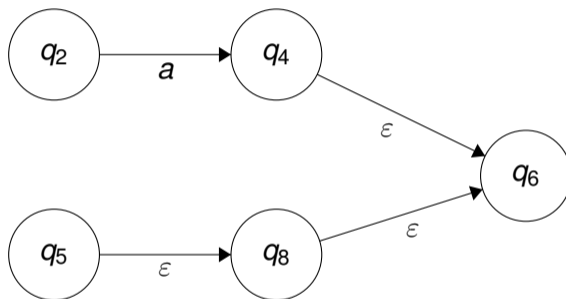
2: $\hat{S} \leftarrow S;$

3: $S \leftarrow \hat{S} \cup \bigcup_{q \in \hat{S}} \delta(q, \epsilon);$

4: **pokiaľ** $S \neq \hat{S}$

5: **vráť** $S;$



Operácia $CLOSURE_{\epsilon}$ 

$$CLOSURE_{\epsilon}(\{q_2, q_5\}) = \{q_2, q_5, q_6, q_8\}.$$

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

Výstup: Množina stavov \hat{Q} a prechodová funkcia $\hat{\delta}$ ekvivalentného DKA.

- 1: zarad' stav $CLOSURE_{\varepsilon}(\{q_0\})$ do \hat{Q} a označ ho ako nespracovaný
- 2: **pokiaľ** v \hat{Q} existuje nespracovaný stav **rob**
- 3: vyber z \hat{Q} ľubovoľný nespracovaný stav q a označ ho ako spracovaný
- 4: **pre všetky** $a \in \Sigma$ **rob**
- 5: urči stav $p = \delta(q, a)$ podľa vzťahu (1)
- 6: **ak** $p \notin \hat{Q}$ **potom**
- 7: zarad' p do \hat{Q} a označ ho ako nespracovaný
- 8: **koniec ak**
- 9: zaznamenaj prechod zo stavu q do stavu p na symbol a
- 10: **koniec pre**
- 11: **koniec pokiaľ**



Prechodová funkcia δ :

δ	0	1	ε
q_0	$\{q_0, q_1\}$	$\{q_0, q_2\}$	\emptyset
q_1	$\{q_1\}$	$\{q_1, q_f\}$	\emptyset
q_2	\emptyset	$\{q_2\}$	$\{q_f\}$
q_f	\emptyset	\emptyset	\emptyset

Prechodová funkcia δ :

δ	0	1
$CLOSURE_{\varepsilon}(\{q_0\}) = \{q_0\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_f\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_1, q_2, q_f\}$
$\{q_0, q_2, q_f\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_f\}$
$\{q_0, q_1, q_2, q_f\}$	$\{q_0, q_1\}$	$\{q_0, q_1, q_2, q_f\}$

Ekvivalencia NKA a DKA

- Počet možných stavov DKA rastie exponenciálne k počtu stavov NKA.
- Výhodou NKA je ich jednoduchšia štruktúra.
- Výhodou DKA je ich jednoduchšia implementácia a následná simulácia ich činnosti.
- Simulácia NKA pre dané vstupné slovo má exponenciálnu zložitosť.
- Simulácia DKA pre dané vstupné slovo má lineárnu zložitosť.



