Udalostné systémy

Formálne jazyky a konečné automaty

Obsah prednášky

- Definícia formálneho jazyka
- Reťazce udalostí, množiny reťazcov udalostí a definície operácií s nimi
- Možnosti špecifikácie formálneho jazyka (= definovanie správania DUS)
- Definícia deterministického konečného automatu
- Definícia jazyka generovaného automatom
- Definícia cieľového jazyka
- Definícia prefixového jazyka
- Blokujúce a neblokujúce automaty
- Definícia deterministického konečného automatu s výstupmi

Formálny jazyk

- Každý DUS má svoju množinu udalostí Σ, abecedu.
- Zreťazením udalostí vieme tvoriť konečné udalostné reťazce, slová.
- Všetky prípustné reťazce, ktoré vie DUS spracovať tvoria jazyk L.

Nech DUS má udalostnú množinu $\Sigma = \{a, b, g\}$

Konečné udalostné reťazce potom tvoria nekonečnú množinu Σ^* :
$\Sigma^* = \{a, aa, aaaaaa, abg, bga, gab, bbaaa, b, ggggggb, agag, \ldots\}$

Nech všetky prípustné reťazce v danom DUS sú nasledovné:
$L = \{aa, bga, b, agag\}$ – tieto tvoria jazyk L, pričom platí $L \subseteq \Sigma^*$

Formálny jazyk - príklad

- Udalostný systém = Automat na cestovné lístky MHD (zjednodušene)

Udalostná množina: $\Sigma = \{voľbaCL,\ platba,\ storno\}$

$\Sigma^* = \{voľbaCL,\ platba,\ storno,\ voľbaCL\ voľbaCL,\ storno\ platba\ storno,\ \dots\}$

$L = \{voľbaCL,\ voľbaCL\ voľbaCL,\ voľbaCL\ storno\ voľbaCL\ platba, \dots\}$

$platba \notin L$
$storno\ storno\ storno \notin L$
$voľbaCL\ storno\ platba \notin L$
...

Formálny jazyk – operácie s reťazcami

Kľúčové operácie pri tvorbe reťazcov udalostí sú **zreťazenie** a **iterácia**

- **Zreťazenie**
  - Reťazec ggb vznikol zreťazením udalosti g a reťazca gb.
  - Reťazec gb tiež vznikol zreťazením udalostí g a b.
  - Reťazec môžeme reprezentovať jedným symbolom, napr: $u = ggb$ alebo $v = aabb$
  - Potom reťazec $w = uv = ggbaabb$
  - Zreťazenie n rovnakých udalostí alebo reťazcov r zapíšeme aj r^n, napr: $r^3 = rrr$
  - Prázdny reťazec ε neobsahuje žiadnu udalosť a pri reťazení platí: $u\varepsilon = \varepsilon u = u$

  - Dĺžka reťazca sa označuje $|w| = 7$ (platí pre príklad $w = ggbaabb$)
  - Dĺžka prázdneho reťazca $|\varepsilon| = 0$
  - Platí tiež $r^0 = \varepsilon$
  - Platí tiež $\varepsilon^n = \varepsilon$

Formálny jazyk – operácie s reťazcami

Kľúčové operácie pri tvorbe reťazcov udalostí sú **zreťazenie** a **iterácia**

- **Iterácia (Kleeneho uzáver)**
  - Nech r je reťazec, potom iterácia r^*vytvorí množinu reťazcov $r^* = \{\varepsilon, r, rr, rrr, \dots\}$
  - Princíp: $r^* = \bigcup_{i=0}^{\infty} r^i = r^0 \cup r^1 \cup r^2 \cup r^3 \cup \cdots = \varepsilon \cup r \cup rr \cup rrr \cup \cdots$
  - Nech $r = ab$, potom $r^* = \{\varepsilon, ab, abab, ababab, \dots\}$

Formálny jazyk – časti reťazcov

Nech s je reťazec dĺžky $|s| = n$, napr. $s = \mathrm{abcdef}$

- **Prefix** t je časť reťazca s dĺžky $|t| = \langle 0, n \rangle$ začínajúca rovnako ako reťazec s

  $t = \varepsilon$ $\quad t = a$ $\quad t = abc$ $\quad t = abcdef$

- **Podreťazec** u je ktorákoľvek časť reťazca s dĺžky $|u| = \langle 0, n \rangle$

  $u = abcdef$ $\quad u = \varepsilon$ $\quad u = ab$ $\quad u = cde$

- **Sufix** v je časť reťazca s dĺžky $|v| = \langle 0, n \rangle$ končiaca rovnako ako reťazec s

  $v = abcdef$ $\quad v = cdef$ $\quad v = f$ $\quad v = \varepsilon$

Formálny jazyk – časti reťazcov

Nech s je reťazec dĺžky $|s| = n$, napr. $s = \mathrm{abcdef}$

- **Reťazec po prefixe** t, formálne $s/t = v$ je zvyšok reťazca po odstrihnutí prefixu

  Nech $t = ab$ potom $s/t = cdef$

  Nech $t = \varepsilon$ potom $s/t = abcdef$

  Nech $t = abcdef$ potom $s/t = \varepsilon$

- Poznámka:

  Reťazce s a ε sú súčasne **prefixom**, **podreťazcom** aj **sufixom** reťazca s

Formálny jazyk – operácie s množinami reťazcov

Jazyk L je vo všeobecnosti množina udalostných reťazcov

Nech $L_1 = \{\varepsilon, a, bgg\}$ a $L_2 = \{r, s\}$

- **Zreťazenie** $L_1 L_2 = \{r, s, ar, as, bggr, bggs\}$
  - Vzájomné zreťazenie všetkých reťazcov z oboch množín

Formálny jazyk – operácie s množinami reťazcov

Jazyk L je vo všeobecnosti množina udalostných reťazcov

Nech $L_1 = \{\varepsilon, a, bgg\}$

- **Iterácia** $L_1{}^* = \varepsilon \cup L_1 \cup L_1L_1 \cup L_1L_1L_1 \cup \cdots$

  $L_1{}^* = \{\varepsilon, a, bgg, aa, abgg, bgga, bggbgg, \ldots\}$

1. Vytvorí sa nová množina s prázdnym reťazcom ε
2. Do novej množiny sa prevezmú všetky samostatné reťazce z množiny L_1, ktoré tam nie sú
3. Zreťazia sa všetky možné dvojice z novej množiny a reťazce sa do nej pridajú
4. Zreťazia sa všetky možné trojice z novej množiny a reťazce sa do nej pridajú
5. ...pokračujeme donekonečna. (<u>Do množiny sa nepridávajú také reťazce, ktoré tam už sú</u>)

Formálny jazyk – operácie s množinami reťazcov

Nech $L = \{\varepsilon, a, bgg\}$

- **Prefixový jazyk** $\bar{L}$

$$\bar{L} = \overline{\{\varepsilon, a, bgg\}} = \{\varepsilon, a, b, bg, bgg\}$$

  – Množina všetkých prefixov jazyka L

Špecifikácia formálnych jazykov

- Znamená definovanie správania DUS
- Formálne jazyky sú množiny ⇒ môžeme využiť spôsoby špecifikácií množín

Nech DUS má udalostnú množinu $\Sigma = \{a, b, g\}$

Všetky konečné udalostné reťazce získame iteráciou Σ^* :

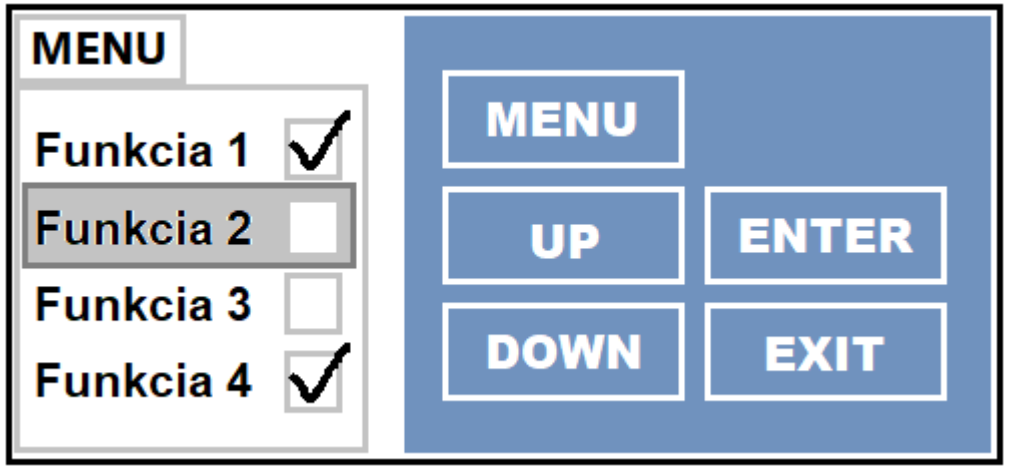
$\Sigma^* = \{a, aa, aaaaaa, abg, bga, gab, bbaaa, b, ggggggb, agag, \ldots\}$

Jazyk $L \subseteq \Sigma^*$ môže byť špecifikovaný napríklad vymenovaním všetkých prípustných reťazcov:

$L = \{aa, bga, b, agag\}$

Špecifikácia formálnych jazykov

- Príklad - Tlačidlá a obrazovka s menu
  - Stlačením MENU sa vyroluje ponuka a vysvieti 1. položka
  - Tlačidlami UP, DOWN sa cyklicky pohybujeme v menu
  - ENTER označí alebo vymaže výber
  - EXIT opustíme menu



Udalostná množina $\Sigma = \{\text{MENU}, \text{UP}, \text{DOWN}, \text{ENTER}, \text{EXIT}\}$

Reťazce patriace do jazyka L sú napríklad:

MENU DOWN DOWN ENTER EXIT

MENU DOWN DOWN UP EXIT

MENU ENTER EXIT

MENU EXIT

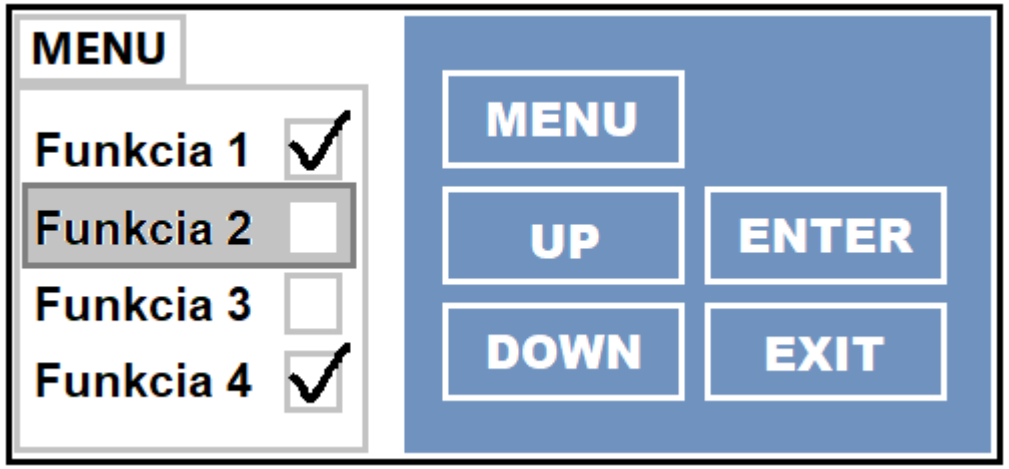
MENU UP ENTER ENTER UP ENTER EXIT

Jazyk L môžeme špecifikovať aj slovným opisom:

$$L = \left\{ \begin{matrix} \textit{Všetky reťazce začínajúce udalosťou} \\ \text{MENU } \textit{a končiace udalosťou } \text{EXIT} \end{matrix} \right\}$$

Špecifikácia formálnych jazykov

- Príklad - Tlačidlá a obrazovka s menu
  - Stlačením MENU sa vyroluje ponuka a označí 1. položka
  - Tlačidlami UP, DOWN sa cyklicky pohybujeme v menu
  - ENTER označí alebo vymaže výber
  - EXIT opustíme menu



Udalostná množina $\Sigma = \{\text{MENU}, \text{UP}, \text{DOWN}, \text{ENTER}, \text{EXIT}\}$

Nasledovné reťazce teda nepatria do jazyka L:

EXIT UP UP ENTER MENU

DOWN DOWN DOWN

ENTER

$$L = \left\{ \begin{matrix} \textit{Všetky reťazce začínajúce udalosťou} \\ \text{MENU } \textit{a končiace udalosťou } \text{EXIT} \end{matrix} \right\}$$

Náš DUS ich totiž nevie spracovať – nevie rozpoznať slová, ktoré nepatria do jeho jazyka.

Špecifikácia formálnych jazykov

- Jazyk L môže byť špecifikovaný aj pomocou regulárnych výrazov
- Rekurzívna definícia regulárnych výrazov:
  - Prázdna množina $\emptyset$ a prázdny reťazec ε sú regulárne výrazy
  - Každá udalosť z množiny udalostí $e \in \Sigma$ je regulárnym výrazom
  - Regulárnymi výrazmi sú aj všetky ostatné prvky vytvorené operáciami:
    - zjednotenie, zreťazenie, iterácia.

- Regulárne výrazy špecifikujú tzv. regulárne jazyky (jazyky typu 3)
  - Bližšie vysvetlenie nájdete v odporúčanej literatúre

- Príklad

$$L = \alpha\beta^* + \beta = \{\varepsilon, \beta, \alpha, \alpha\beta, \alpha\beta\beta, \alpha\beta\beta\beta, \alpha\beta\beta\beta\beta, \dots\}$$

Špecifikácia formálnych jazykov

- Vymenovaním prípustných reťazcov
- Slovným opisom
- Regulárnym výrazom
- Udaním logického predikátu nad množinou Σ^* ($\forall\ s \in \Sigma^*$ | také, že platí...)
- ...

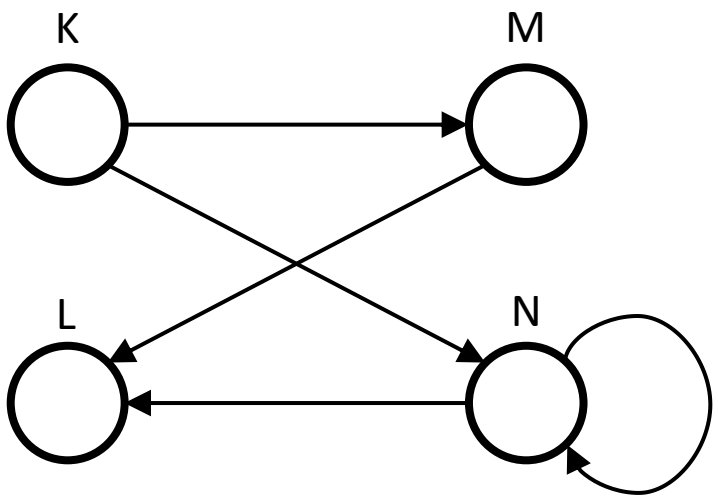
- Často je zložité takto definovať správanie reálneho DUS
- Často je zložité pracovať s takýmito špecifikáciami
  - Potrebujeme **kompaktné štruktúry**, ktoré vedia definovať jazyk
  - Potrebujeme, aby mali dobre **definované operácie** pre manipuláciu s nimi
  - Potrebujeme, aby sa pomocou nich dali **analyzovať aj zložité systémy**

- Využijeme grafický nástroj – **matematické grafy**

Matematický graf

- Jednoduchý orientovaný matematický graf G
  - Je daný usporiadanou dvojicou G = (A, R)
    - A – konečná neprázdna množina uzlov grafu (krúžky)
    - R – binárna relácia na množine A, ktorá môže byť aj prázdna – orientované hrany (šipky)
- Príklad

  G = ({K, L, M, N}, { (K, M), (K, N), (M, L), (N, L), (N, N) })



Matematický graf

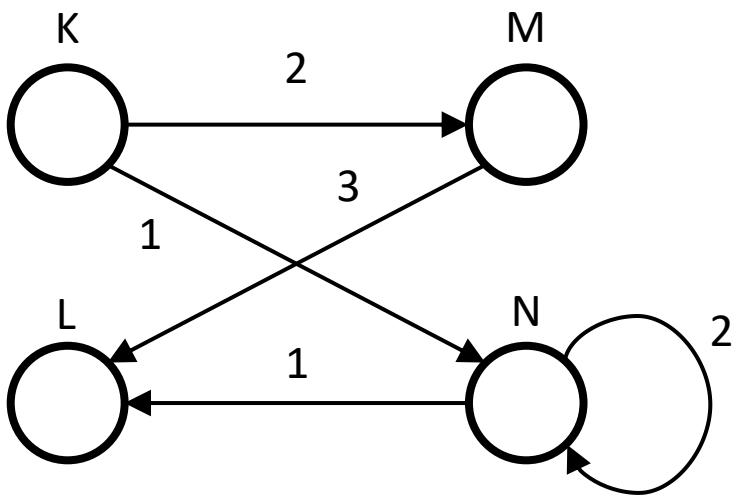
- Ohodnotený orientovaný matematický graf G
  - Je daný usporiadanou 6-ticou $G = (A, R, f_1, f_2, S_1, S_2)$
    - A – konečná neprázdna množina uzlov grafu (krúžky)
    - R – binárna relácia na množine A, ktorá môže byť aj prázdna – orientované hrany (šipky)
    - f_1 – funkcia, ktorá priraďuje ohodnotenie (váhy) jednotlivým uzlom $A \rightarrow S_1$
    - f_2 – funkcia, ktorá priraďuje ohodnotenie (váhy) jednotlivým hranám $R \rightarrow S_2$
    - S_1, S_2 – množiny váh. Jedna alebo obe môžu byť prázdne.
- Príklad 1

  Nech $S_1 = \emptyset$

  Nech $S_2 = N$

  f_1 – uzlom nepriradí nič

  f_2 – hranám priradí čísla



Matematický graf

- Ohodnotený orientovaný matematický graf G
  - Je daný usporiadanou 6-ticou $G = (A, R, f_1, f_2, S_1, S_2)$
    - A – konečná neprázdna množina uzlov grafu (krúžky)
    - R – binárna relácia na množine A, ktorá môže byť aj prázdna – orientované hrany (šipky)
    - f_1 – funkcia, ktorá priraďuje ohodnotenie (váhy) jednotlivým uzlom $A \rightarrow S_1$
    - f_2 – funkcia, ktorá priraďuje ohodnotenie (váhy) jednotlivým hranám $R \rightarrow S_2$
    - S_1, S_2 – množiny váh. Jedna alebo obe môžu byť prázdne.
- Príklad 2

  A = {ON, OFF}

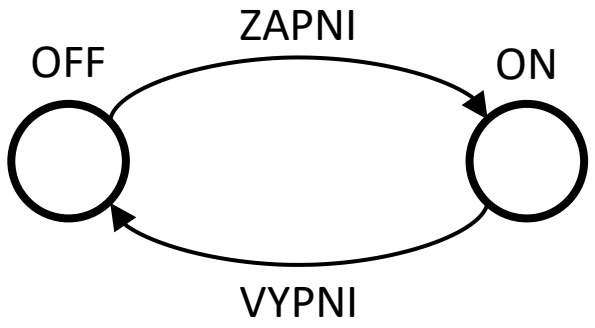
  R = {(ON, OFF), (OFF, ON)}

  $S_1 = \emptyset$

  S_2 = {ZAPNI, VYPNI}

  f_1 – uzlom nepriradí nič

  f_2 – hranám priradí udalosti



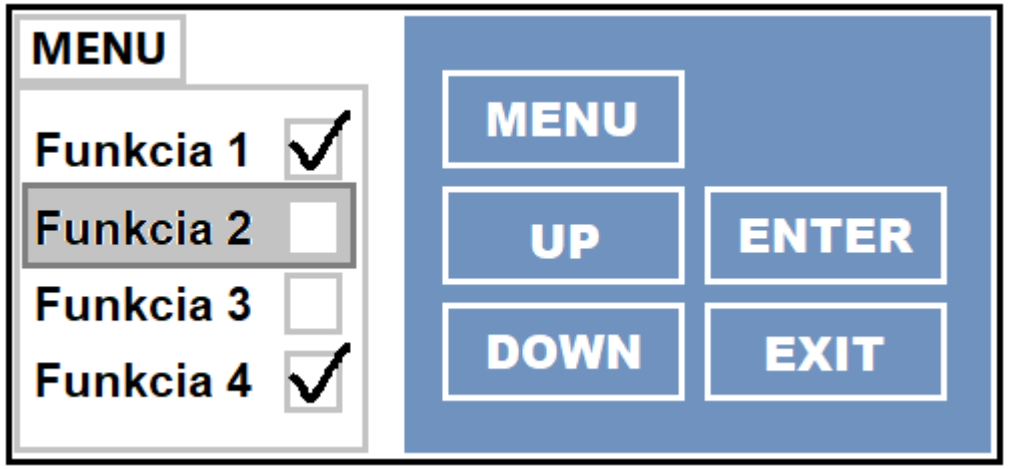
Príklad Menu – jazyk definovaný grafom

$\Sigma = \{\text{MENU}, \text{UP}, \text{DOWN}, \text{ENTER}, \text{EXIT}\}$

$Q = \{\text{stand_by}, \text{in_menu}\}$

$q_0 = \textbf{stand_by}$

$L = \left\{ \begin{array}{c} \textit{Všetky reťazce začínajúce udalosťou} \\ \text{MENU } \textit{a končiace udalosťou } \text{EXIT} \end{array} \right\}$



Reťazce patriace do jazyka L sú napríklad:

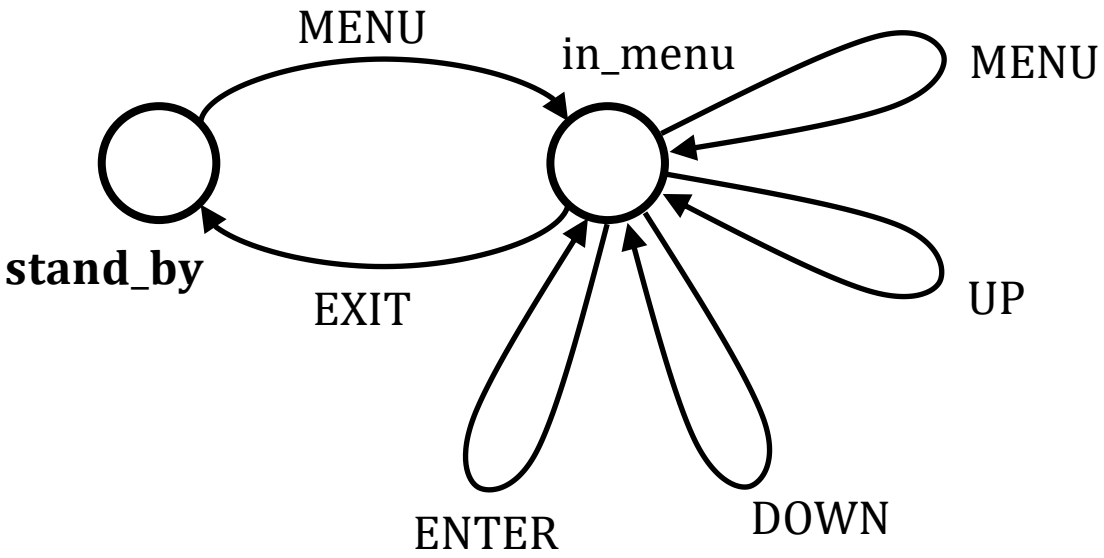
MENU DOWN DOWN ENTER EXIT

MENU DOWN DOWN UP EXIT

MENU ENTER EXIT

MENU EXIT

MENU UP ENTER ENTER UP ENTER EXIT



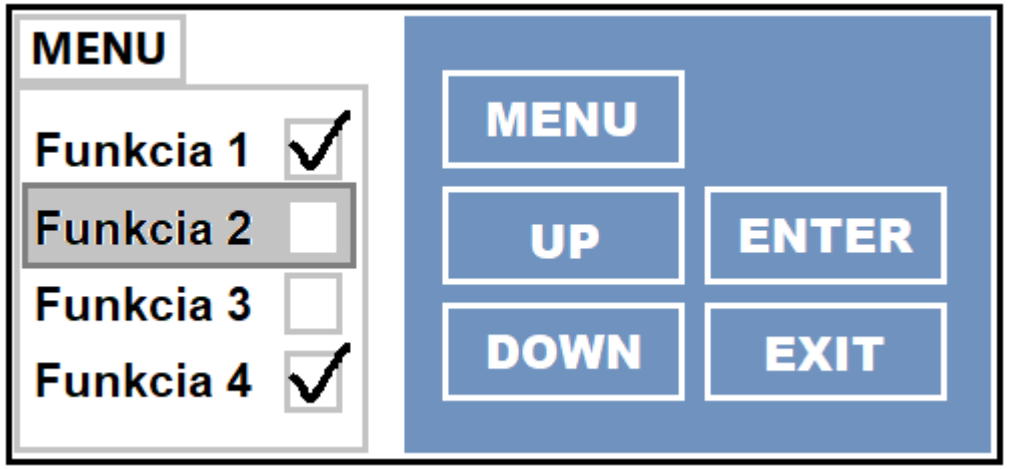
Príklad Menu – jazyk definovaný grafom

$\Sigma = \{\text{MENU}, \text{UP}, \text{DOWN}, \text{ENTER}, \text{EXIT}\}$

$Q = \{\text{stand_by}, \text{in_menu}\}$

$q_0 = \textbf{stand_by}$

$L = \left\{ \begin{array}{c} \textit{Všetky reťazce začínajúce udalosťou} \\ \text{MENU } \textit{a končiace udalosťou } \text{EXIT} \end{array} \right\}$

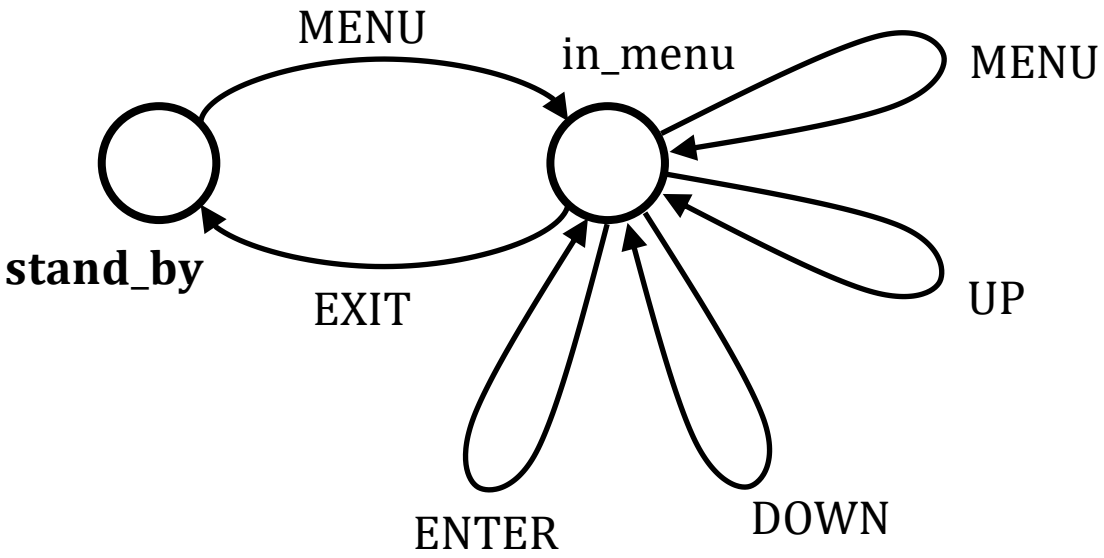


Reťazce nepatriace do jazyka L:

EXIT UP UP ENTER MENU

DOWN DOWN DOWN

ENTER

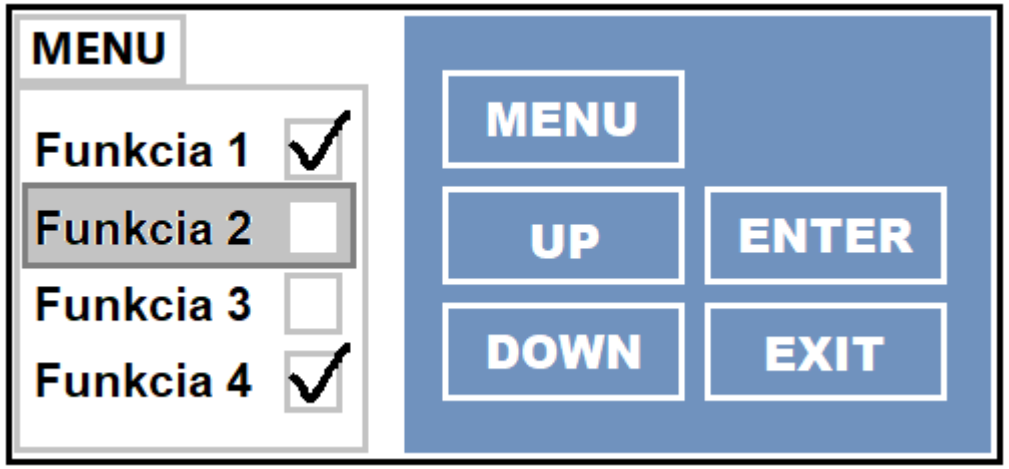


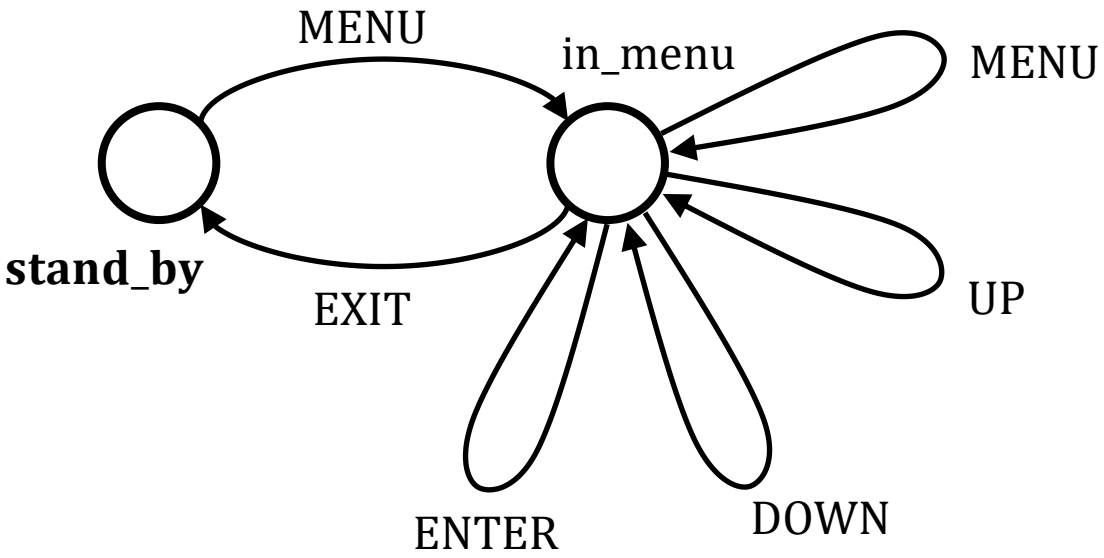
Doplníme nejaké fičury

$\Sigma = \{\text{MENU}, \text{UP}, \text{DOWN}, \text{ENTER}, \text{EXIT}\}$

$Q = \{\text{stand_by}, \text{in_menu}\}$

$q_0 = \mathbf{stand_by}$

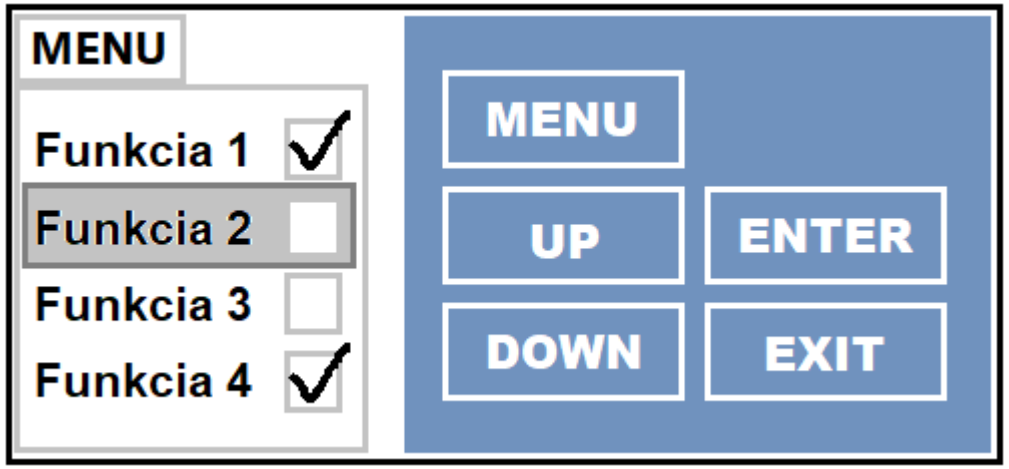


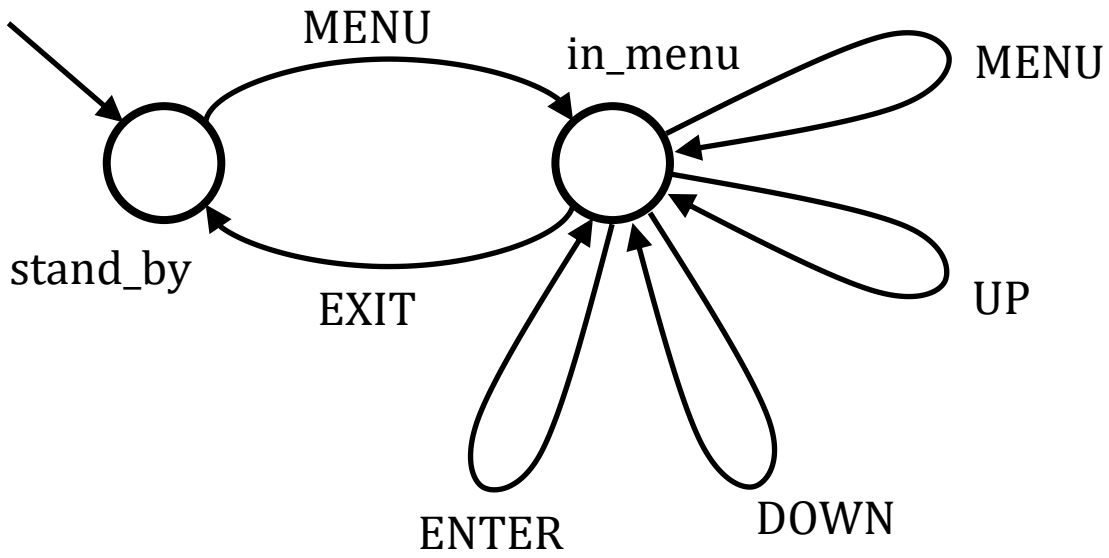


Doplníme nejaké fičury

$\Sigma = \{\text{MENU}, \text{UP}, \text{DOWN}, \text{ENTER}, \text{EXIT}\}$
$Q = \{\text{stand_by}, \text{in_menu}\}$
$q_0 = \text{stand_by}$

Počiatočný stav q_0 označíme šípkou „odnikiaľ“



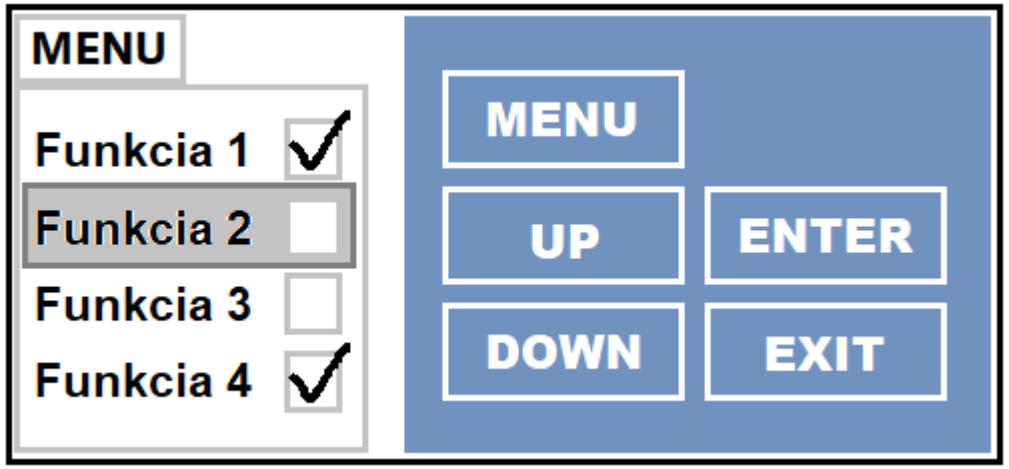


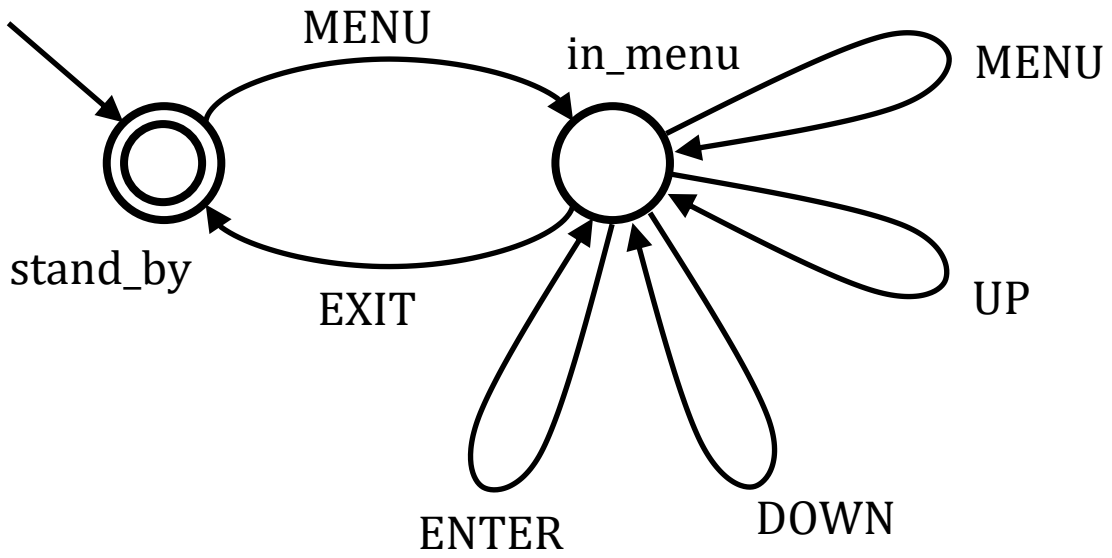
Doplníme nejaké fičury

$\Sigma = \{\text{MENU}, \text{UP}, \text{DOWN}, \text{ENTER}, \text{EXIT}\}$
$Q = \{\text{stand_by}, \text{in_menu}\}$
$q_0 = \text{stand_by}$
$F = \{\text{stand_by}\}$

F je množina finálnych stavov.
Označuje stavy, v ktorých systém dokončil svoju úlohu alebo operáciu.

Označuje sa dvojitým krúžkom.





Deterministický konečný automat (DKA)

- Je definovaný 5-ticou:

$$A = (\Sigma, Q, q_0, \delta, F)$$

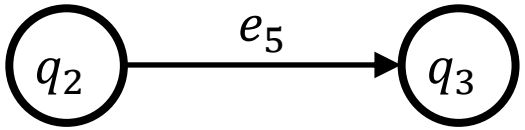
- $\Sigma = \{e_1, e_2, \dots, e_n\}$ je množina udalostí
- $Q = \{q_0, q_1, \dots, q_m\}$ je konečná množina stavov
- $q_0 \in Q$ je počiatočný stav
- $\delta\colon\ Q \times \Sigma \to Q$ je parciálna prechodová funkcia
- $F \subseteq Q$ je množina finálnych stavov (môže byť aj prázdna)

Parciálna prechodová funkcia δ: $Q \times \Sigma \rightarrow Q$

- Funkcia reprezentuje reláciu R v matematickom grafe
- Priraďuje príslušný STAV ku dvojici STAV-UDALOSŤ

  Napr.:

$$\delta(q_2, e_5) = q_3$$

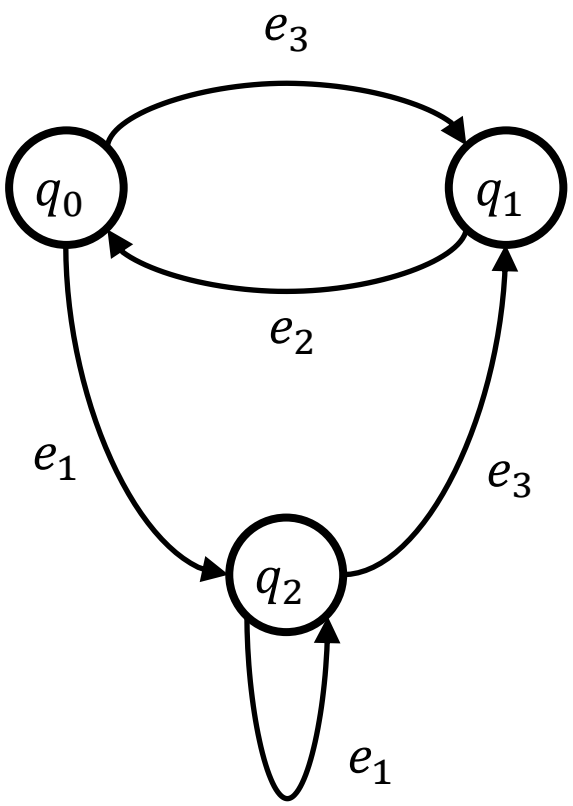


- Je parciálna, pretože nie je definovaná pre všetky dvojice STAV-UDALOSŤ
- Je definovaná iba pre prípustné udalosti v danom stave
- Funkcia priraďuje nasledovný stav jednoznačne $\Longrightarrow$ automat je deterministický

Parciálna prechodová funkcia $\delta\colon\ Q \times \Sigma \to Q$

- Dá sa znázorniť tabuľkou alebo grafom

$\delta(q, e\ )$	e_1	e_2	e_3
q_0	q_2	$-$	q_1
q_1	$-$	q_0	$-$
q_2	q_2	$-$	q_1



Ďalšie pojmy a označenia

- $\Gamma(q)$ je množina všetkých prípustných udalostí v danom stave q.

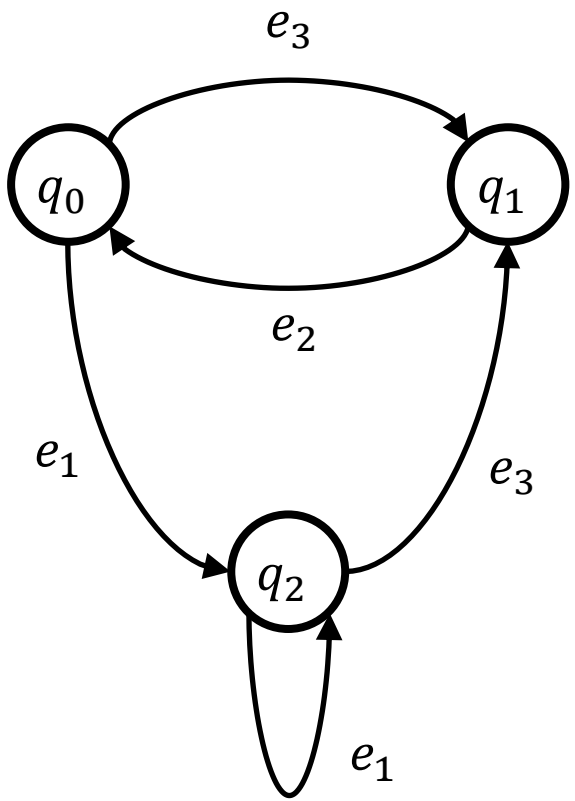
$$\Gamma(q) = \{e \in \Sigma \mid \delta(q, e) \text{ je definovaná } \}$$

$\Gamma(q_0) = \{e_1, e_3\}$

$\Gamma(q_1) = \{e_2\}$

$\Gamma(q_2) = \{e_1, e_3\}$

$\delta(q, e)$	e_1	e_2	e_3
q_0	q_2	$-$	q_1
q_1	$-$	q_0	$-$
q_2	q_2	$-$	q_1



Ďalšie pojmy a označenia

- Stavový sled ω – postupnosť stavov začínajúca v q_0, pre ktorú je v každom kroku definovaná funkcia $\delta\left(q_{i_{p-1}}, e_{j_p}\right) = q_{i_p}$ pre kroky $p = 1 \dots k$

$\omega = q_0, q_{i_1}, q_{i_2}, \dots, q_{i_k}$ pre $i \in \langle 0, m \rangle$ pretože $Q = \{q_0, q_1, \dots, q_m\}$

- Udalostný reťazec η – postupnosť udalostí, ktorá spôsobuje stavový sled ω

$\eta = e_{j_1}, e_{j_2}, e_{j_3}, \dots, e_{j_k}$ pre $j \in \langle 1, n \rangle$ pretože $\Sigma = \{e_1, e_2, \dots, e_n\}$

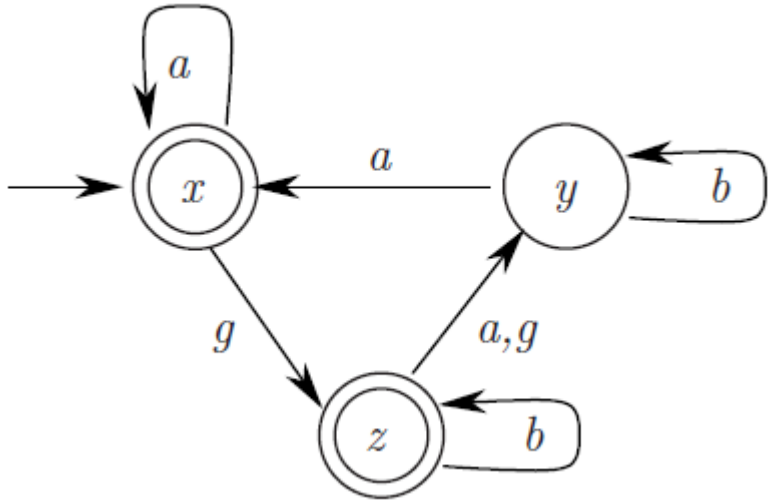
Ďalšie pojmy a označenia

- Rozšírená definícia prechodovej funkcie $\delta\colon\ Q \times \Sigma^* \to Q$
  - Funkcia vráti výsledný stav, do ktorého sa automat dostane z počiatočného stavu po prijatí udalostného reťazca η

Nech $\eta = agab$

$\delta(x, \eta) = y$

$\delta(x, agab) = \delta(\delta(\delta(\delta(x, a), g), a), b) = y$



  - Rozšírená definícia umožňuje kompaktnejší zápis oproti pôvodnej definícii

Jazyky reprezentované automatom

- Nech deterministický konečný automat je generátor $G = (\Sigma, Q, q_0, \delta, F)$

- **Jazyk generovaný automatom** G je $L(G)$

  $L(G) = \{\eta \in \Sigma^* \mid \delta(q_0, \eta)\ je\ definovaná\}$

  – Je množina takých udalostných reťazcov η, pre ktoré je prechodová funkcia δ definovaná
  – Množina prípustných udalostných reťazcov η v automate G

- **Cieľový jazyk** (*marked*) **automatu** G je $L_m(G)$

  $L_m(G) = \{\eta \in \Sigma^* \mid \delta(q_0, \eta) \in F\}$

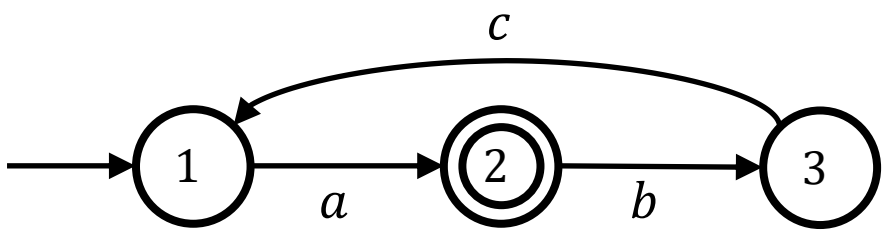
  – Je množina takých prípustných udalostných reťazcov η, ktorých stavový sled končí vo finálnom stave

$$L_m(G) \subseteq L(G)$$

Jazyky reprezentované automatom - príklad

- Nech deterministický konečný automat je generátor $G = (\Sigma, Q, q_0, \delta, F)$

- $\Sigma = \{a, b, c\}$ $\quad Q = \{1, 2, 3\}$
- $q_0 = 1$ $\quad F = \{2\}$



- **Jazyk generovaný automatom** G

  $L(G) = \{a, ab, abc, abca, abcab, abcabc, \dots\}$

- **Cieľový jazyk automatu** G

  $L_m(G) = \{a, abca, abcabca, abcabcabca, \dots\}$

Automat G dokončí úlohu iba vtedy, keď vykoná (príjme) udalostný reťazec z množiny $L_m(G)$.

Prefixový jazyk

- $\overline{L(G)} = \{s \in \Sigma^* , \exists t \in \Sigma^* \mid st \in L(G)\}$
  - Množina všetkých prefixov jazyka $L(G)$

Prefixový jazyk - príklad

- Jazyk generovaný automatom

  $L(G) = \{a, ab, abc, abca, abcab, abcabc, \dots\}$

- Prefixový jazyk generovaného jazyka

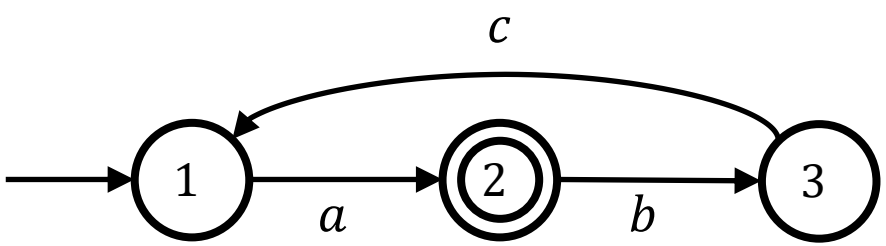
  $\overline{L(G)} = \{a, ab, abc, abca, abcab, abcabc, \dots\}$

- Cieľový jazyk automatu

  $L_m(G) = \{a, abca, abcabca, abcabcabca, \dots\}$

- Prefixový jazyk cieľového jazyka

  $\overline{L_m(G)} = \{a, ab, abc, abca, abcab, abcabc, \dots\}$



Prefixový jazyk

- $\overline{L(G)} = \{s \in \Sigma^* , \exists t \in \Sigma^* \mid st \in L(G)\}$
  - Množina všetkých prefixov jazyka $L(G)$

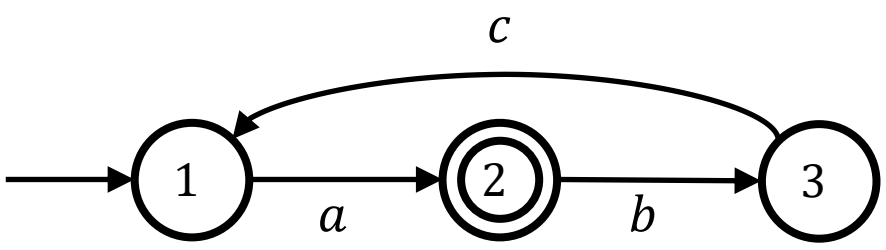
- Jazyk generovaný automatom je vždy prefixovo uzavretý $L(G) = \overline{L(G)}$
  - Keď je možné vykonať v automate G nejaký udalostný reťazec,

  ...tak je možné v automate G vykonať aj jeho prefixy

- Cieľový jazyk nemusí byť prefixovo uzavretý $L_m(G) \subseteq \overline{L_m(G)}$
  - Nie všetky prefixy cieľového jazyka musia byť prefixami reťazcov, ktoré skončia vo finálnom stave

- Všeobecne platí: $L_m(G) \subseteq \overline{L_m(G)} \subseteq L(G)$

Prefixový jazyk - príklad



- Jazyk generovaný automatom

  $L(G) = \{a, ab, abc, abca, abcab, abcabc, \dots\}$

- Prefixový jazyk generovaného jazyka

  $\overline{L(G)} = \{a, ab, abc, abca, abcab, abcabc, \dots\}$ Každý $\eta \in \overline{L(G)}$ je zároveň $\eta \in L(G)$

- Cieľový jazyk automatu

  $L_m(G) = \{a, abca, abcabca, abcabcabca, \dots\}$

- Prefixový jazyk cieľového jazyka

  $\overline{L_m(G)} = \{a, ab, abc, abca, abcab, abcabc, \dots\}$ abc môže byť prefixom $abcab \notin L_m(G)$

Príklad

- Stavy diskrétneho udalostného systému sa menia podľa automatu na obrázku
- Systém dokončí úlohu po výskyte udalostných reťazcov:

$$L_m(G) = \{ab, abgab, abgabgab \dots\}$$

- Aké má systém možnosti po prijatí reťazca: ag

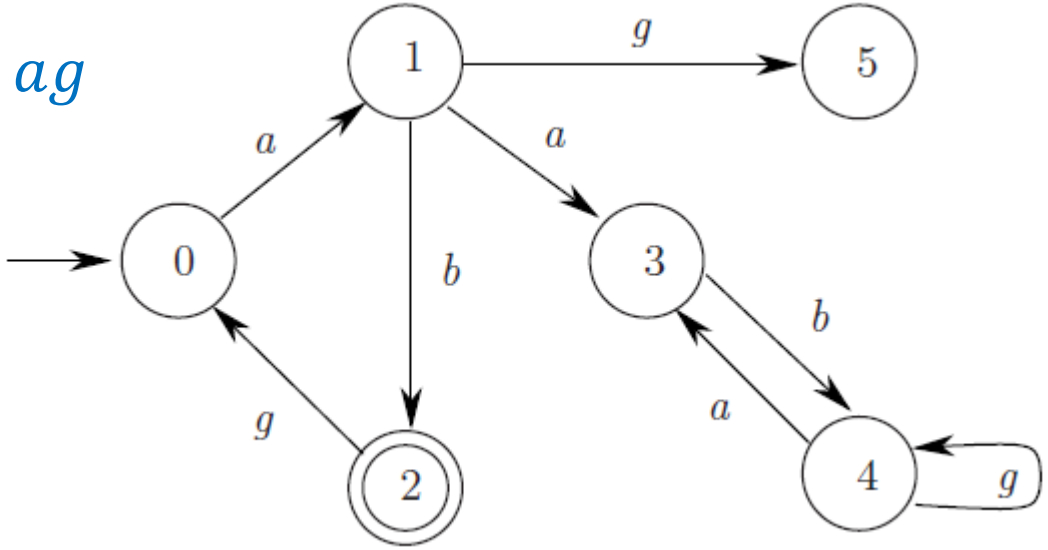
$$\Gamma(5) = \emptyset \quad \Rightarrow \quad \text{Zostane blokovaný}$$

  – Nikdy nedokončí operáciu

  – Z obrázka je zrejmé: $ag, abgag \in L(G)$

  – Vidíme že: $ag, abgag \notin \overline{L_m(G)}$

  – Žiaden reťazec, ktorý dovedie systém do stavu blokovania nemôže byť prefixom reťazca z cieľového jazyka



Príklad

- Stavy diskrétneho udalostného systému sa menia podľa automatu na obrázku
- Systém dokončí úlohu po výskyte udalostných reťazcov:

$$L_m(G) = \{ab, abgab, abgabgab \dots\}$$

- Aké má systém možnosti po prijatí reťazca: aa

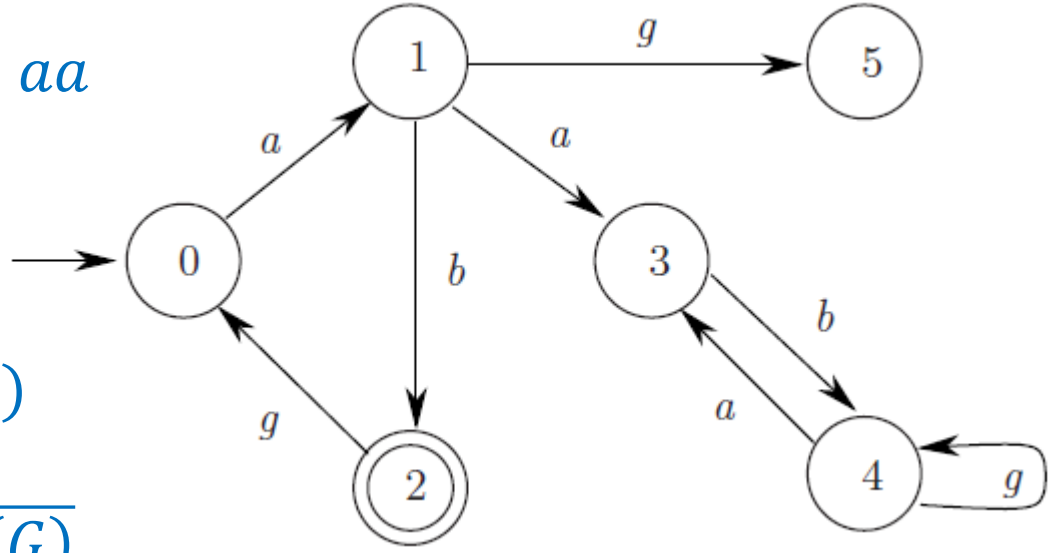
  Uviazne v stavoch 3 a 4 - Zostane blokovaný

  – Nikdy nedokončí operáciu

  – Z obrázka je zrejmé: $aa, aabg, abgaab \in L(G)$

  – Vidíme že: $aa, aabg, abgaab \notin \overline{L_m(G)}$

  – Žiaden reťazec, ktorý dovedie systém do stavu blokovania nemôže byť prefixom reťazca z cieľového jazyka



Blokujúci automat

- Automat G je blokujúci, keď cieľový jazyk, je pravou podmnožinou jazyka generovaného automatom:

$$\overline{L_m(G)} \subset L(G)$$

  - V jazyku $L(G)$ vtedy existujú reťazce, ktoré nie sú prefixami cieľového jazyka
  - A teda tieto reťazce nikdy nedovedú automat do cieľového stavu, čiže zostane blokovaný

Neblokujúci automat

- Automat G je neblokujúci, keď cieľový jazyk sa rovná jazyku generovaného automatom:

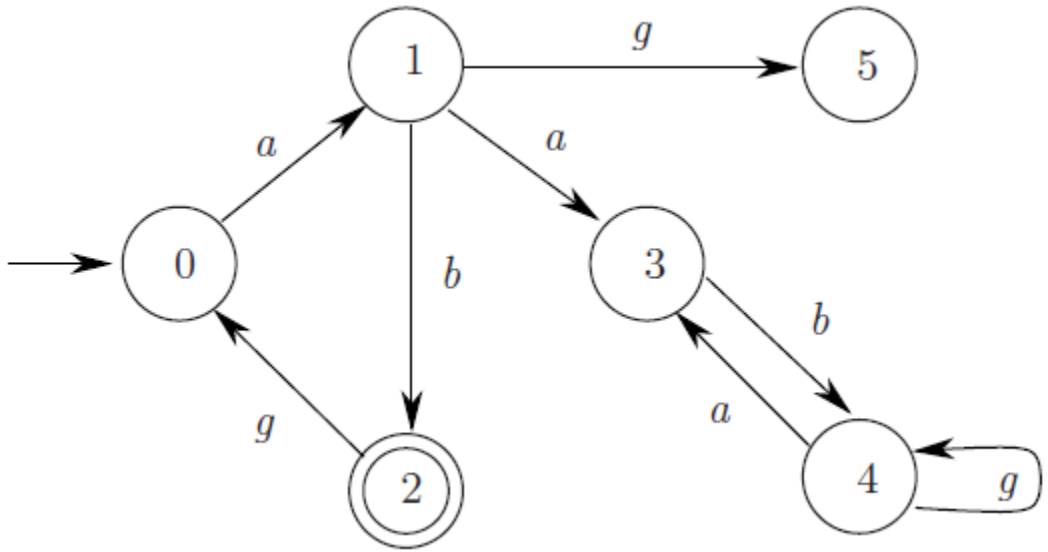
$$\overline{L_m(G)} = L(G)$$

  - Všetky reťazce v jazyku $L(G)$ sú aj prefixami cieľového jazyka
  - A teda tieto reťazce vždy dovedú automat do cieľového stavu

Blokujúci automat: Deadlock / Livelock

- Deadlock: Stav q, pre ktorý $\Gamma(q) = \emptyset$, teda funkcia $\delta(q, \cdot)$ nie je pre žiadnu udalosť e definovaná a zároveň stav q nie je finálny stav, t.j. $q \notin F$
  - V príklade: $q = 5$

- Livelock: Množina stavov, ktoré nie sú z množiny F, sú vzájomne dosiahnuteľné (tvoria silne súvislý komponent), a žiadna udalosť nespôsobí prechod von z tejto množiny.
  - V príklade stavy 3 a 4.



Deterministický konečný automat s výstupmi (DKAV)

- Automat interpretovaný pre riadenie
  - Má generovať takú riadiacu postupnosť, ktorá zabezpečí požadované správanie

- Ku každému stavu (alebo dvojici stav-udalosť) je definovaný výstup

$$AC = (A, Y, \varphi\ )$$

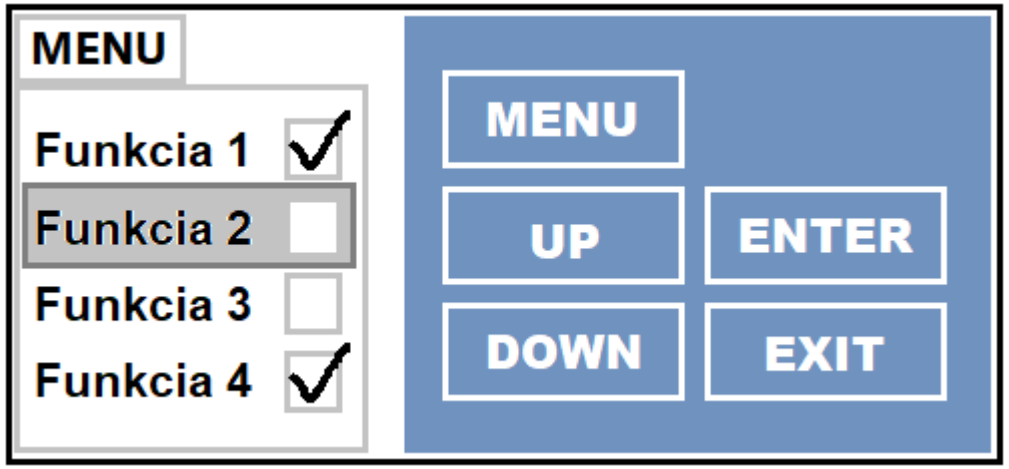
- A je deterministický konečný automat
- Y je konečná množina výstupov
- $\varphi\colon\ Q \longrightarrow Y$ je výstupná funkcia v prípade Mooreovho automatu

($\varphi\colon\ Q \times \Sigma \longrightarrow Y$ je výstupná funkcia v prípade Mealyho automatu)

DKAV - príklad

$AC = (A, Y, \varphi\ )$

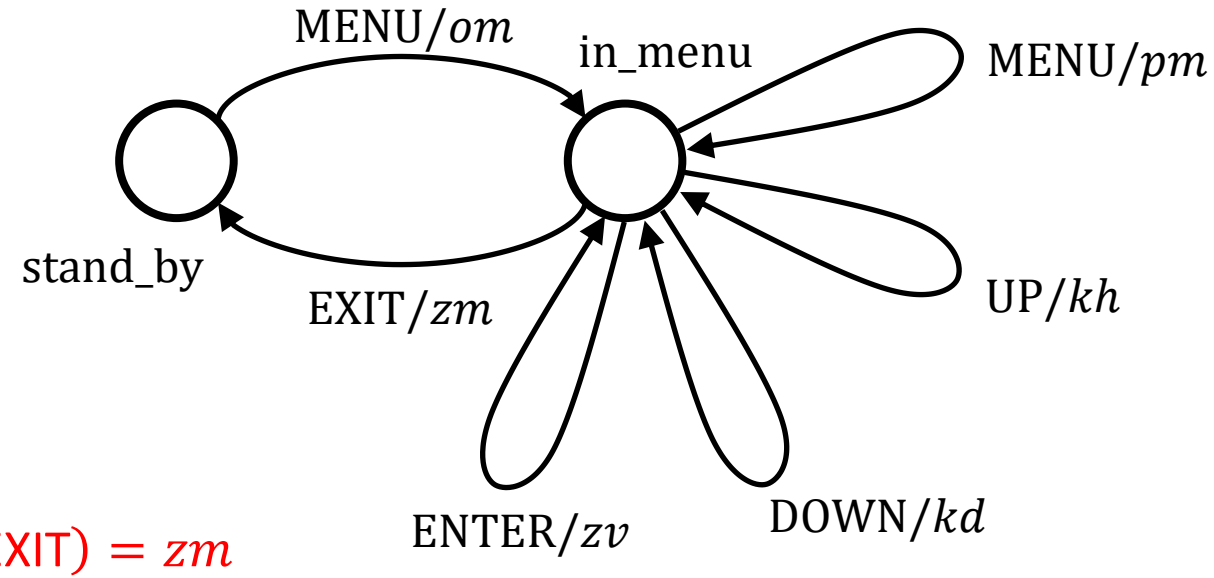
- A DKA z príkladu - Menu
- $Y = \{om, pm, kh, kd, zv, zm\}$
- $\varphi\colon\ Q \times \Sigma \longrightarrow Y$ výstupná funkcia (Mealy)



Výstupy sú napr. príkazy v programe

- $om - otvor\ menu$
- $pm - prekresli\ menu$
- $kh - kurzor\ hore$
- $kd - kurzor\ dolu$
- $zv - zmeň\ výber$
- $zm - zatvor\ menu$

$\varphi(\text{in_menu, EXIT}) = zm$



Ďakujem za pozornosť.