

Turingův stroj, jazyky Turingových strojů

Jan Konečný

24. září 2013

Turingův stroj (Turing machine)

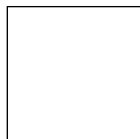
Turingův stroj (TS)

- Alan Turing, 1936 (a-machine)
- účel: porozumět omezením mechanického výpočtu

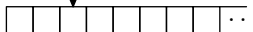
TS se skládá

- z řídicí jednotky, která se vždy nachází v jednom z konečného množství stavů
- ze zleva omezené nekonečné pásky rozdělené na políčka. V každém políčku je zapsán jeden symbol.
- z čtecí/zapisovací hlavy, která je vždy umístěna nad jedním políčkem pásky.

řídicí jednotka



čtecí/zapisovací hlava



páska

Konečné automaty – připomínka z KMI/FJAA

(pedagogická berlička)

Definice

Konečný automat je dán

- Q – abecedou stavů,
- Σ – abecedou symbolů,
- $\delta : Q \times \Sigma \rightarrow Q$ – přechodovou funkcí,
- $q_0 \in Q$ – počátečním stavem,
- $F \subseteq Q$ – množinou přijímacích (koncových) stavů.

Definice

Konfigurace konečného automatu: $\langle q, w \rangle \in Q \times \Sigma^*$

- $q \in Q$ – aktuální stav,
- $w \in \Sigma^*$ – nezpracovaná část vstupního slova.

Definice

Krok výpočtu $\vdash_{KA}$ $\mathbf{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ je binární relace nad množinou konfigurací KA $\mathbf{A}$ definovaná takto:

$$\langle q_1, w_1 \omega \rangle \vdash \langle q_2, \omega \rangle \text{ právě když } \delta(q_1, w_1) = q_2.$$

Výpočet je $\vdash^*$ – reflexivní, tranzitivní uzávěr relace $\vdash$.

Definice

Říkáme, že KA $\mathbf{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$ přijímá slovo $w \in \Sigma^*$, pokud $\langle q_0, w \rangle \vdash^* \langle f, \varepsilon \rangle$ a $f \in F$. V opačném případě říkáme, že KA zamítá slovo w .

(ε označuje prázdný řetězec)

Definice

Jazyk konečného automatu $\mathbf{A}$ je množina slov, které automat přijímá. Značíme $L(\mathbf{A})$.

Grafová reprezentace KA

(furt ještě berlička)

Orientovaný graf

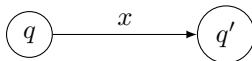
- **stavy** – uzly



- **počáteční stav a přijímací stavy**



- **přechody** $\delta(q, x) = q'$ – orientované ohodnocené hrany



(konec berličky)

Turingův stroj, definice

Program Turingova stroje lze chápat jako množinu elementárních instrukcí ve tvaru:

„Pokud je řídicí jednotka ve stavu q a čtecí/zapisovací hlava čte symbol a , tak změň stav řídicí jednotky na q' , na pásku zapiš a' a posuň čtecí/zapisovací hlavu o jedno políčko směrem d .“

Takováto instrukce se zapisuje jako $\delta(q, a) = (q', a', d)$ budeme ji nazývat přechod. Celý program, tedy množinu takovýchto instrukcí, pak nazýváme přechodovou funkcí TS.

Definice

Turingův stroj je struktura $\langle Q, \Sigma, \Gamma, \delta, q_{\text{start}}, q_+, q_- \rangle$ daná:

- 1 neprázdnou konečnou množinou stavů Q ,
- 2 vstupní abecedou Σ , t.ž. $\sqsubset \notin \Sigma$,
- 3 páskovou abecedou Γ , t.ž. $\Sigma \subset \Gamma$, $\sqsubset \in \Gamma$,
- 4 přechodovou funkcí $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$,
- 5 počátečním stavem $q_{\text{start}} \in Q$,
- 6 přijímacím stavem $q_+ \in Q$ a zamítacím stavem $q_- \in Q$. $q_+ \neq q_-$.

Příklad

TS $M_1 = \langle Q, \Sigma, \Gamma, \delta, q_0, q_+, q_- \rangle$, kde

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_+, q_-\}$,
- $\Sigma = \{0\}$,
- $\Gamma = \Sigma \cup \{\sqcup, \times\}$,
- δ je dána následující tabulkou:

	0	$\sqcup$	$\times$
q_0	$(q_1, \sqcup, R)$	$(q_-, \sqcup, R)$	$(q_-, \times, R)$
q_1	$(q_2, \times, R)$	$(q_+, \sqcup, R)$	$(q_1, \times, R)$
q_2	$(q_3, 0, R)$	$(q_4, \sqcup, L)$	$(q_2, \times, R)$
q_3	$(q_2, \times, R)$	$(q_-, \sqcup, R)$	$(q_3, \times, R)$
q_4	$(q_4, 0, L)$	$(q_1, \sqcup, R)$	$(q_4, 0, L)$

Turingův stroj, konfigurace

Definice

Konfigurace TS $T = \langle Q, \Sigma, \Gamma, \delta, q_{\text{start}}, q_+, q_- \rangle$ je uspořádaná trojice

$$(q, \alpha, n) \in Q \times \Gamma^* \times \mathbb{N}_0.$$

Slovy:

Konfigurace TS T – trojice, která zachycuje aktuální status všech tří komponent Turingova stroje:

- q je aktuální stav řídicí jednotky,
- α je obsah pásky,
- n je pozice hlavy.

Konfigurace $(q, \alpha_{\sqcup}, n)$ a (q, α, n) budeme považovat za totožné.

Definice

Iniciální konfigurace TS T pro vstup $w \in \Sigma^*$ je $(q_0, w, 0)$.

Přijímající konfigurace TS T je konfigurace

$$(q_+, \alpha, n) \text{ pro jakékoli } \alpha \in \Gamma^* \text{ a } n \in \mathbb{N}_0.$$

Zamítající konfigurace TS T je konfigurace

$$(q_-, \alpha, n) \text{ pro jakékoli } \alpha \in \Gamma^*, n \in \mathbb{N}_0.$$

Iniciální konfigurace – řídicí jednotka nachází v q_0 , na pásce je zapsáno vstupní slovo (následované nekonečným počtem prázdných symbolů $\sqcup$) a hlava je nad nejlevějším políčkem pásky (s indexem 0).

Přijímající konfigurace – řídicí jednotka je v q_+ .

Zamítající konfigurace – řídicí jednotka je v q_- .

Přijímající nebo zamítající konfigurace je *koncová konfigurace*.
Jiná konfigurace je *nekoncová konfigurace*.

Alternativně budeme konfiguraci zapisovat jako řetězec $\alpha q \beta \in \Gamma^* Q \Gamma^*$. Konfigurace $\alpha q \beta$ představuje status stroje, který má na pásce zapsán řetězec $\alpha \beta$, hlava je nad prvním symbolem řetězce β , řídicí jednotka je ve stavu q .

Definice

Krok výpočtu TS je definován jako binární relace na množině konfigurací: Nechť $(q, a_1 \dots a_n, i)$ je taková konfigurace T , kde $q \neq q_{\pm}, n \in \mathbb{N}_0, a_0, \dots, a_n \in \Gamma, i \leq n$.

(a) Je-li $1 \leq i \leq n$ a $\delta(q, a_i) = (q', b, L)$, pak

$$(q, a_0 \dots a_n, i) \vdash (q', a_0 \dots a_{i-1} b a_{i+1} \dots a_n, i - 1).$$

(b) Je-li $\delta(q, a_1) = (q', b, L)$, pak

$$(q, a_0 \dots a_n, 0) \vdash (q', b a_1 \dots a_n, 0).$$

(c) Je-li $\delta(q, a_i) = (q', b, R)$, pak

$$(q, a_0 \dots a_n, i) \vdash (q', a_0 \dots a_{i-1} b a_{i+1} \dots a_n, i + 1).$$

Definice kroku říká, jakým způsobem lze z nekonečné konfigurace odvodit novou podle přechodové funkce δ :

Pokud jsme ve stavu q a hlava čte páskový symbol a_i , a máme přechod $\delta(q, a_i) = (q', b, X)$, pak:

- řídicí jednotka změní stav na q' ,
- na pásku se do políčka, nad kterým je hlava, zapíše b (přepíše se tedy původní symbol a_i)
- pohyb hlavy:
 - (a) Pokud $X = L$ a hlava není nad nejlevějším políčkem pásky, pohne o jedno políčko doleva.
 - (b) Pokud $X = L$ a hlava je nad nejlevějším políčkem pásky, hlava se nepohne vůbec.
 - (c) Pokud $X = R$, hlava se pohne o jedno políčko doprava.

Definice

Výpočet $\vdash^*$ TS je reflexivní, tranzitivní uzávěr relace $\vdash$.

Zápis $\mathcal{C} \vdash \mathcal{C}'$ čteme takto:

konfigurace $\mathcal{C}'$ je odvoditelná z konfigurace $\mathcal{C}$ jedním krokem výpočtu.

Zápis $\mathcal{C} \vdash^* \mathcal{C}'$ čteme takto:

konfigurace $\mathcal{C}'$ je odvoditelná z konfigurace $\mathcal{C}$.

Definice

Nechť T je TS a $w \in \Sigma^*$:

T *přijímá* w , pokud $\mathcal{C}_0^w \vdash^* \mathcal{C}_+$, kde $\mathcal{C}_0^w$ je iniciální konfigurace T pro vstup w a $\mathcal{C}_+$ je přijímající konfigurace stroje T .

T *zamítá* w , pokud $\mathcal{C}_0^w \vdash^* \mathcal{C}_-$, kde $\mathcal{C}_0^w$ je iniciální konfigurace T pro vstup w a $\mathcal{C}_-$ je zamítající konfigurace stroje T .

T *cyklí* pro w , pokud w nepřijímá ani nezamítá.

TS $M_1 = \langle Q, \Sigma, \Gamma, \delta, q_0, q_+, q_- \rangle$, kde

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_+, q_-\}$,
- $\Sigma = \{0\}$,
- $\Gamma = \Sigma \cup \{\sqcup, \times\}$,
- δ je dána následující tabulkou:

	0	$\sqcup$	$\times$
q_0	$(q_1, \sqcup, R)$	$(q_-, \sqcup, R)$	$(q_-, \times, R)$
q_1	$(q_2, \times, R)$	$(q_+, \sqcup, R)$	$(q_1, \times, R)$
q_2	$(q_3, 0, R)$	$(q_4, \sqcup, L)$	$(q_2, \times, R)$
q_3	$(q_2, \times, R)$	$(q_-, \sqcup, R)$	$(q_3, \times, R)$
q_4	$(q_4, 0, L)$	$(q_1, \sqcup, R)$	$(q_4, \times, R)$

Příklad

Iniciální konfigurace M_1 pro vstupní slovo $w = 00000$ je $(q_0, \underline{00000}, 0)$. Z této konfigurace je v jednom kroku odvoditelná konfigurace $(q_1, \sqcup\underline{0000}, 1)$. Celý výpočet nad slovem w by vypadal takto:

$(q_0, \underline{00000}, 0) \vdash (q_1, \sqcup\underline{0000}, 1) \vdash (q_2, \sqcup\times\underline{000}, 2) \vdash (q_3, \sqcup\times\underline{000}, 3) \vdash (q_2, \sqcup\times\underline{0\times\underline{0}}, 4) \vdash (q_3, \sqcup\times\underline{0\times\underline{0\times\underline{0}}}, 5) \vdash (q_-, \sqcup\times\underline{0\times\underline{0\times\underline{0\times\underline{0}}}}, 6)$.

Výpočet tedy končí v zamítající konfiguraci $(q_-, \sqcup\times\underline{0\times\underline{0\times\underline{0\times\underline{0\times\underline{0}}}}}, 6)$, TS M_1 tedy zamítá slovo 00000.

TS $M_1 = \langle Q, \Sigma, \Gamma, \delta, q_0, q_+, q_- \rangle$, kde

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_+, q_-\}$,
- $\Sigma = \{0\}$,
- $\Gamma = \Sigma \cup \{\sqcup, \times\}$,
- δ je dána následující tabulkou:

	0	$\sqcup$	$\times$
q_0	$(q_1, \sqcup, R)$	$(q_-, \sqcup, R)$	$(q_-, \times, R)$
q_1	$(q_2, \times, R)$	$(q_+, \sqcup, R)$	$(q_1, \times, R)$
q_2	$(q_3, 0, R)$	$(q_4, \sqcup, L)$	$(q_2, \times, R)$
q_3	$(q_2, \times, R)$	$(q_-, \sqcup, R)$	$(q_3, \times, R)$
q_4	$(q_4, 0, L)$	$(q_1, \sqcup, R)$	$(q_4, \times, L)$

Příklad

Výpočet TS M_1 pro vstupní slovo $w = 0000$ by vypadal takto

$(q_0, \underline{0000}, 0) \vdash (q_1, \sqcup \underline{000}, 1) \vdash (q_2, \sqcup \times \underline{00}, 2) \vdash (q_3, \sqcup \times \underline{00}, 3) \vdash (q_2, \sqcup \times \times \underline{\sqcup}, 4) \vdash$
 $(q_4, \sqcup \times \underline{0} \times, 3) \vdash (q_4, \sqcup \times \underline{0} \times, 2) \vdash (q_4, \sqcup \times \underline{0} \times, 1) \vdash (q_4, \sqcup \times \underline{0} \times, 0) \vdash (q_0, \sqcup \times \underline{0} \times, 1) \vdash$
 $(q_1, \sqcup \times \times \times, 2) \vdash (q_1, \sqcup \times \times \times, 3) \vdash (q_1, \sqcup \times \times \times \underline{\sqcup}, 4) \vdash (q_+, \sqcup \times \times \times \underline{\sqcup}, 5)$

Výpočet tedy končí v přijímající konfiguraci $(q_+, \sqcup \times \times \times \underline{\sqcup}, 5)$, TS M_1 tedy přijímá slovo 0000.

Přechodový diagram – grafová reprezentace TS

Orientovaný graf

- **stavy** – uzly

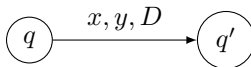


- **počáteční stav a koncové stavy**



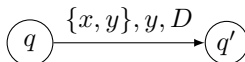
Zamítací stav (a do něj vedoucí přechody) se nezakreslují.

- **přechody** $\delta(q, x) = (q', y, D)$ – orientované ohodnocené hrany

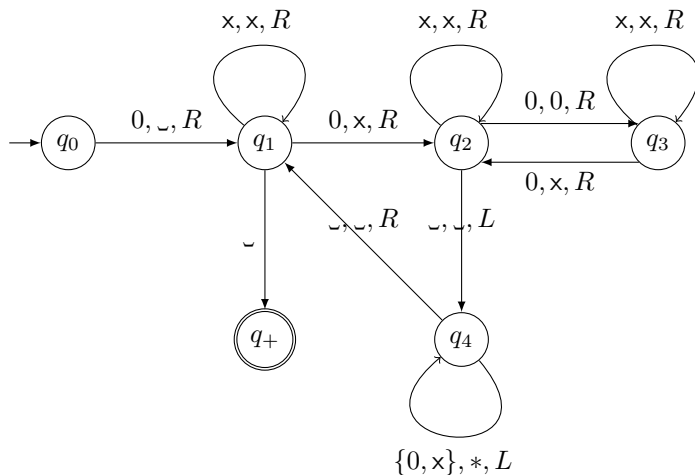


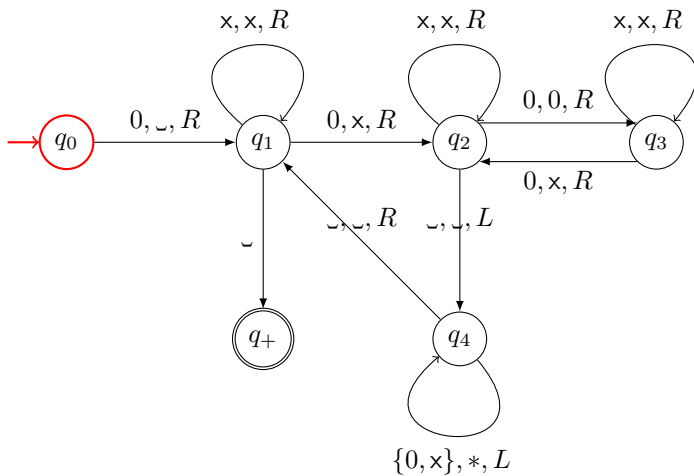
více přechodů se (někdy) dá zakreslit jako jeden, když jsou mezi stejnými stavy:

$\delta(q, y) = (q', y, D)$ a $\delta(q, x) = (q', y, D)$.

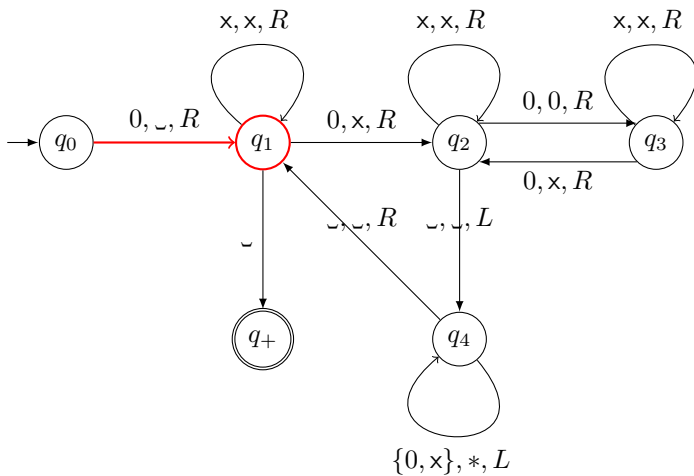


Grafová reprezentace TS (přechodový diagram TS)

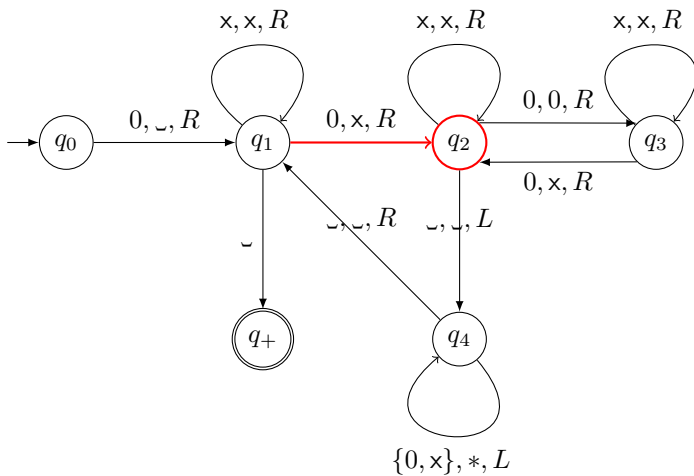




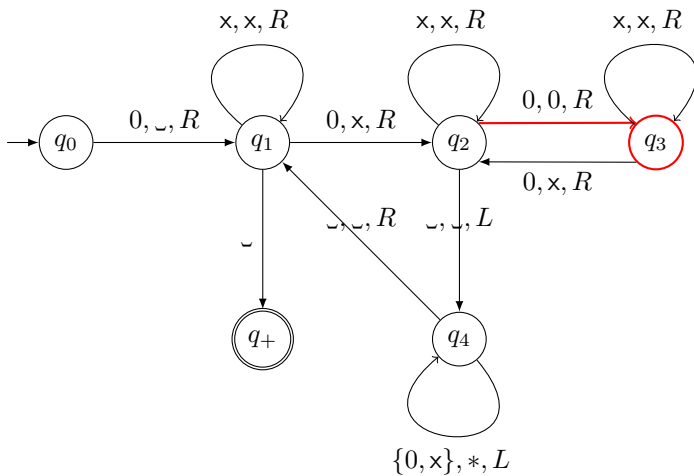
$(q_0, \underline{0}000, 0)$



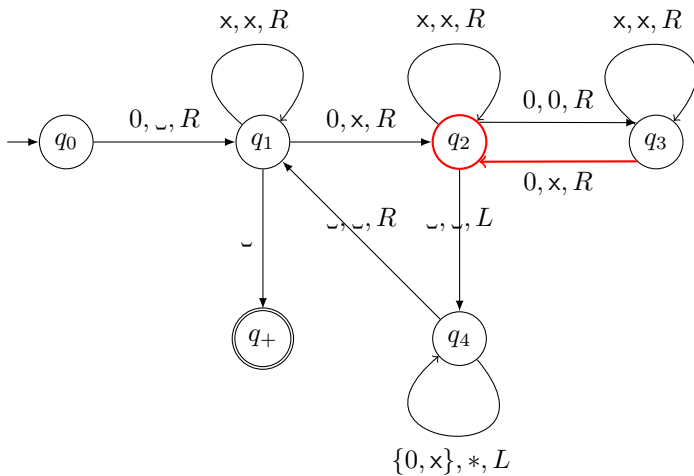
$(q_1, \sqcup \underline{0}00, 1)$



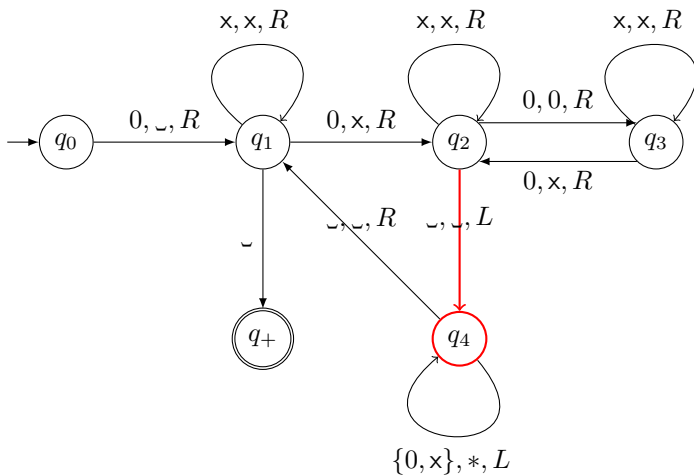
$(q_2, \sqcup x \underline{0} 0, 2)$



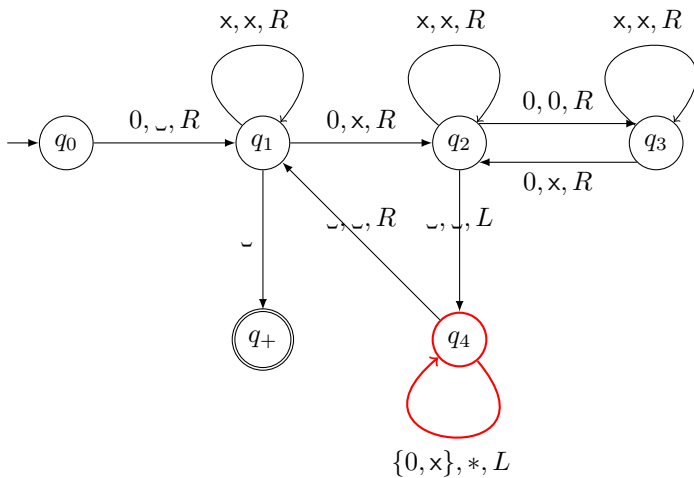
$(q_3, \sqcup x 0 \underline{0}, 3)$



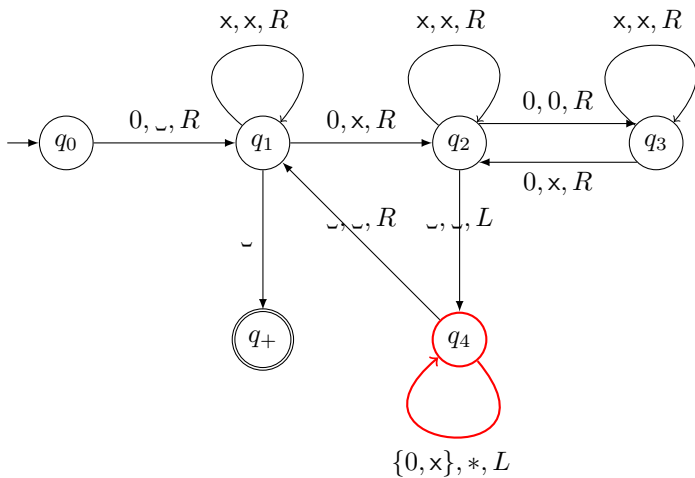
$(q_2, \sqcup x 0 x \sqcup, 4)$



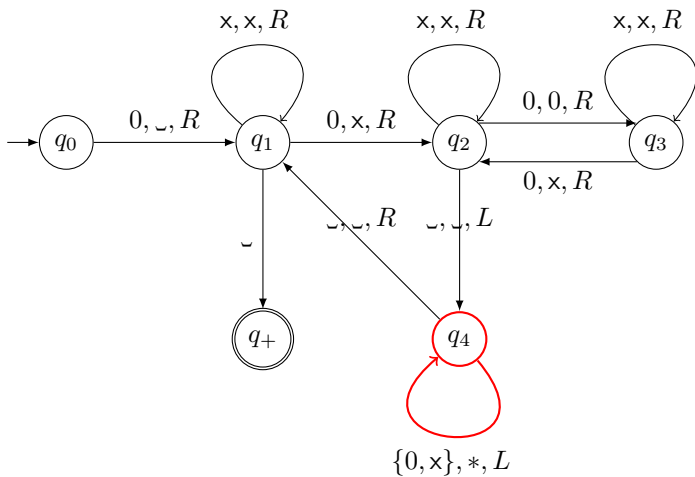
$(q_4, \sqcup x 0 \sqcup, 3)$



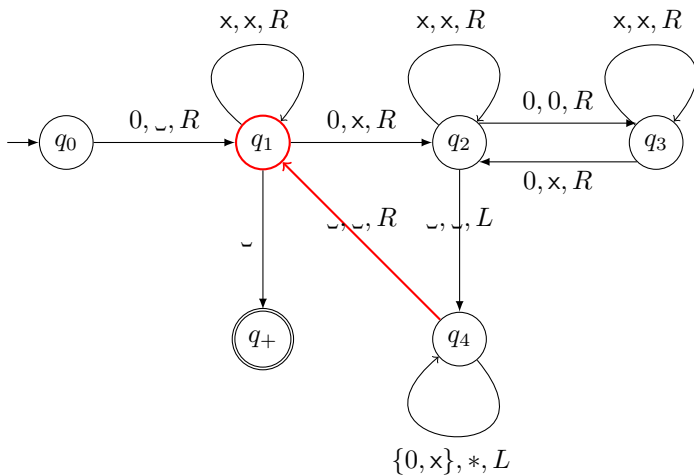
$(q_4, \sqcup x 0 \sqcup x \sqcup, 2)$



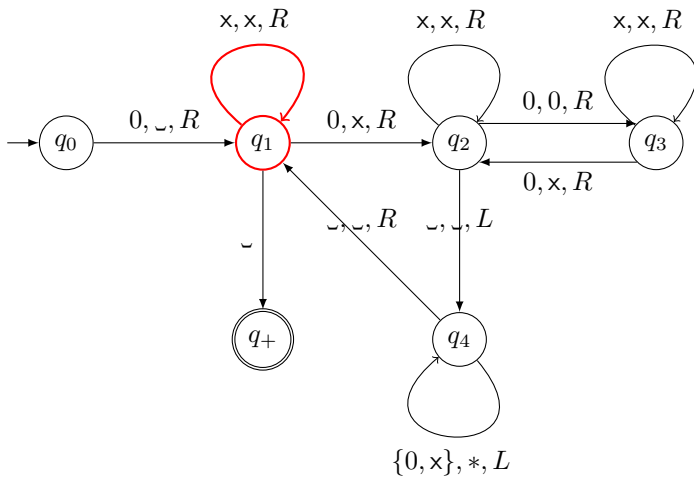
$(q_4, \sqcup x 0 x \sqcup, 1)$



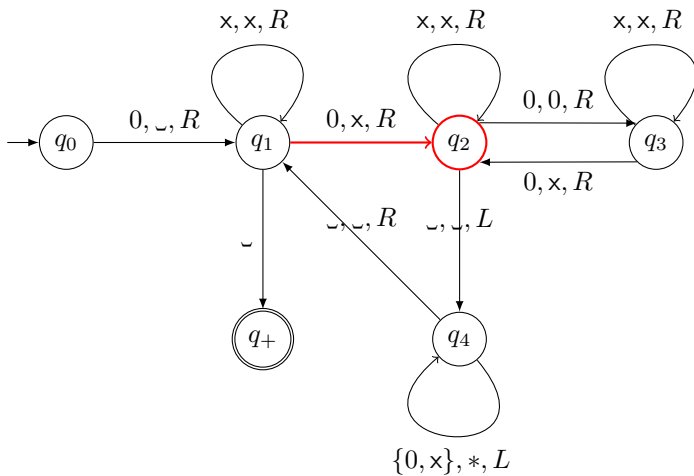
$(q_4, \sqcup x 0 x \sqcup, 0)$



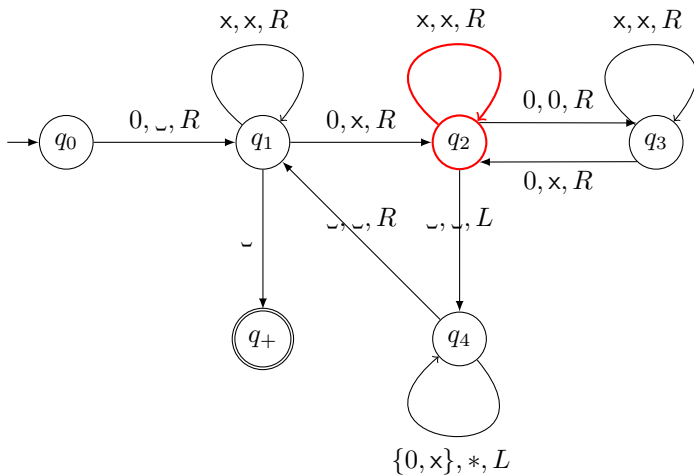
$(q_1, \sqcup x 0 x \sqcup, 1)$



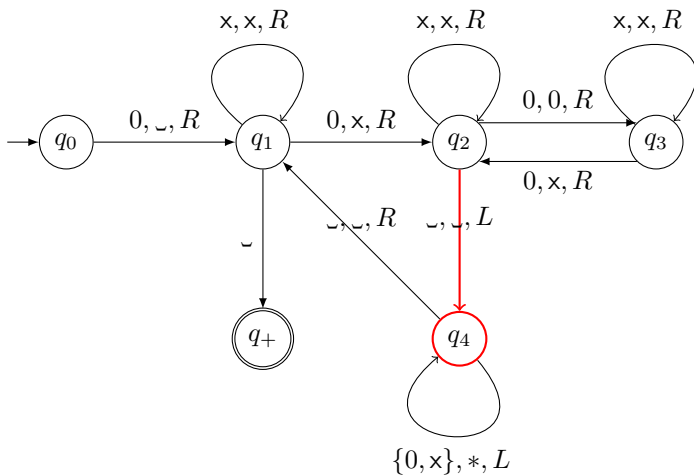
$(q_1, \sqcup x 0 x \sqcup, 2)$



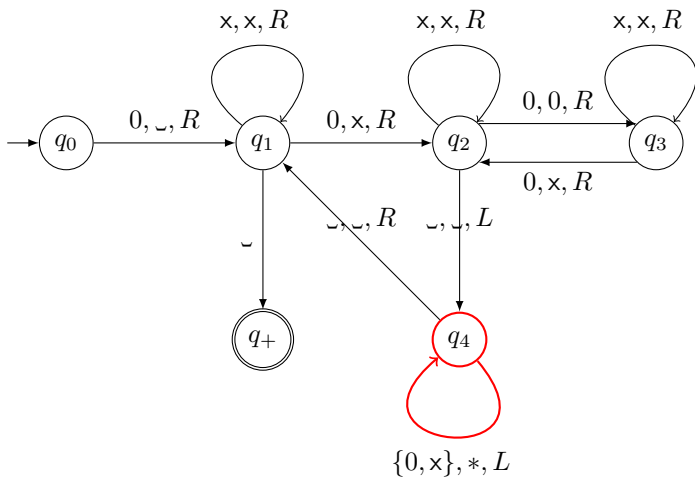
$(q_2, \sqcup x x \sqcup, 3)$



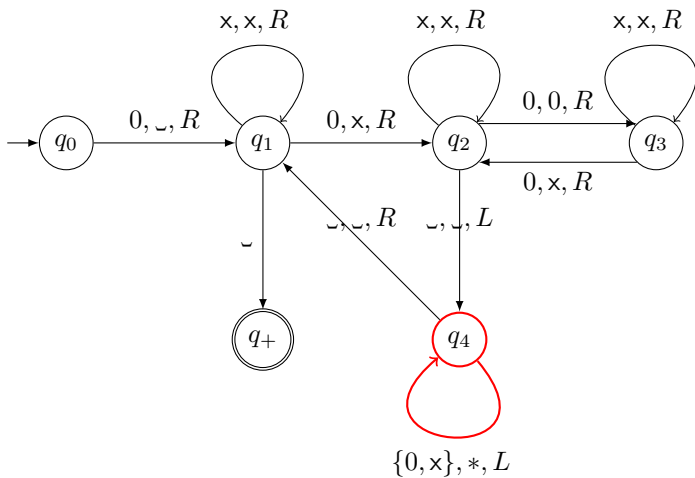
$(q_2, \sqcup xxx \sqcup, 4)$



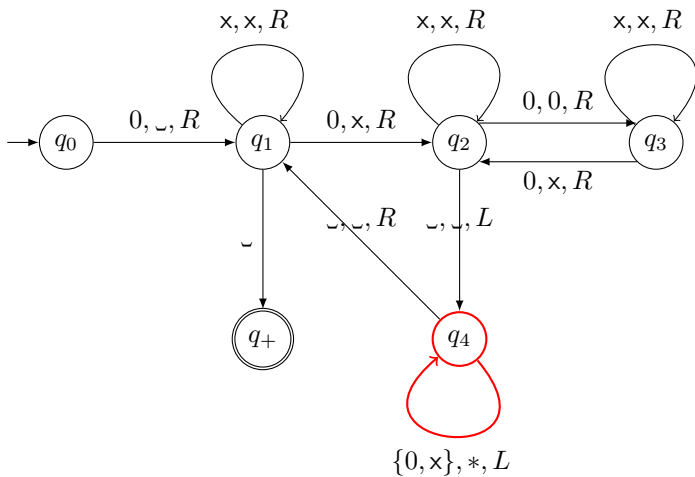
$(q_2, \sqcup x x \sqcup, 3)$



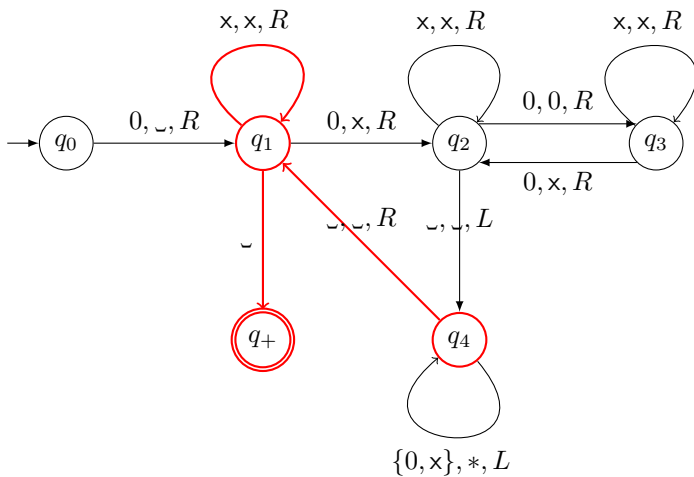
$(q_2, \sqcup x x x \sqcup, 2)$



$(q_2, \sqcup \underline{x} x x \sqcup, 1)$



$(q_2, \sqcup xxx \sqcup, 0)$



$(q_+, \sqcup xxx \sqcup \sqcup, 5)$

Definice

Množinu všech slov $w \in \Sigma^*$, které TS T přijímá značíme $L(T)$ a nazýváme *jazykem Turingova stroje*, t.j.

$$L(T) = \{w \mid w \in \Sigma^*, (w, q_0, 0) \vdash^* C_+\}$$

Jazyk $L(T)$ nazýváme *jazyk přijímaný TS T* .

Říkáme, že TS T *přijímá jazyk $L(T)$* .

Pokud navíc platí, že TS T zamítá každé $w \notin L(T)$, nazýváme jazyk $L(T)$ *jazyk rozhodovaný TS T* .

Říkáme, že TS T *rozhoduje jazyk $L(T)$* .

A říkáme, že TS T *rozhoduje jazyk*, pokud rozhoduje nějaký jazyk $L(T)$.

Pozn.: To znamená, že pokud TS rozhoduje jazyk, p.k. nikdy necyklí.

Pozn.: TS, který rozhoduje jazyk, je nazýván také *decider*.

Definice

Jazyk $L \subseteq \Sigma^*$ nazveme

- *jazyk rozhodovaný TS*, pokud existuje TS T , který jej rozhoduje.
- *jazyk přijímaný TS*, pokud existuje TS T , který jej přijímá.

Jazykům rozhodovaným TS říkáme také *rekurzivní jazyky*; jazykům přijímaným TS říkáme také *částečně rekurzivní jazyky* nebo *rekurzivně spočetné jazyky*.

Příklad

Například $A = \{0^{2^n} \mid n \geq 0\}$ je jazyk rozhodovaný TS, protože M_1 jej rozhoduje.

Třidu všech rekurzivních jazyků značíme R .

Třidu všech částečně rekurzivních jazyků značíme $\check{R}$.

Jasně platí $R \subseteq \check{R}$.

Rozhodovací problémy

Rozhodovací problém (též jen *problém*) je otázka, na kterou se odpovídá „ano“ nebo „ne“.

Instance problému je otázka doplněná konkrétními hodnotami, pro které je ta otázka pokládána.

Podle odpovědi budeme rozlišovat

- *instance s odpovědí „ano“* a
- *instance s odpovědí „ne“*.

Příklad

Rozhodovacím problémem je například:

„Je x menší než y ?“,

kde instance jsou dvojice x, y přirozených čísel.

$x = 3, y = 7$ je instance s odpovědí „ano“.

$x = 3, y = 3$ je instance s odpovědí „ne“.

Problémy budu zapisovat následovně

<i>název problému</i>
Instance: <i>instance</i>
Otázka: <i>otázka</i>

Máme-li problém $\mathcal{P}$,

- pro instance x s odpovědí „ano“ píšeme $x \in \mathcal{P}$,
- pro instance x s odpovědí „ne“ píšeme $x \notin \mathcal{P}$.

Příklad

Např.: Označme $\mathcal{P}_{x < y}$ problém z předchozího slajdu:

Porovnání čísel ($\mathcal{P}_{x < y}$)
Instance: $x, y \in \mathbb{N}$
Otázka: Platí $x < y$?

Píšeme například $\langle x = 3, y = 7 \rangle \in \mathcal{P}_{x < y}$ a $\langle x = 3, y = 3 \rangle \notin \mathcal{P}_{x < y}$.

Nadále budeme uvažovat pouze problémy jejichž instance jsou konečné a lze je tedy reprezentovat řetězci.

Reprezentaci instance x řetězcem budeme nazývat *zakódování* a značit $[x]$

Příklad

instance problému $\mathcal{P}_{x < y}$, tedy dvojice x, y můžeme zakódovat takto

$$[x, y] = 0^x 10^y.$$

Definice

Říkáme, že TS T řeší problém P , pokud

- přijímá všechny jeho instance s odpovědí „ano“,
- zamítá všechny jeho instance s odpovědí „ne“.

Říkáme, že TS T částečně řeší problém P , pokud přijímá právě všechny jeho instance s odpovědí „ano“.

Definice

Problém $\mathcal{P}$ nazveme

- *řešitelný*, pokud existuje TS T , který ho řeší.
- *částečně řešitelný*, pokud existuje TS T , který ho částečně řeší.

Jazyk a problém je tatáž věc

- Problém $\mathcal{P}$ můžeme chápat jako jazyk $L_{\mathcal{P}}$

$$L_{\mathcal{P}} = \{[x] \mid x \text{ je instancí } \mathcal{P} \text{ s odpovědí „ano“}\}$$

- Jazyk $L \in \Sigma^*$ pak můžeme chápat jako jako problém $\mathcal{P}_L$:

Problém $\mathcal{P}_L$
Instance: Řetězec $x \in \Sigma^*$
Otázka: Patří x do jazyka L ?

Příklad

Problém $\mathcal{P}_{x < y}$ můžeme považovat za jazyk

$$L_{x < y} = \{0^x 10^y \mid x, y \in \mathbb{N}, x < y\}$$

Zjevně také platí, že:

- řešitelný problém je rekurzivní jazyk,
- částečně řešitelný problém je částečně rekurzivní jazyk.

Příklad

Příklady problémů z FJaA:

Problém, který je řešitelný

Ekvivalence konečných automatů

Instance: Konečné automaty K_1 a K_2 .

Otázka: Přijímají K_1 a K_2 stejný jazyk?
--

Problém, který je částečně řešitelný, ale není řešitelný

Nonekvivalence bezkontextových gramatik

Instance: Bezkontextové gramatiky G_1 a G_2 .
--

Otázka: Generují G_1 a G_2 různé jazyky?

Problém, který není částečně řešitelný

Ekvivalence bezkontextových gramatik

Instance: Bezkontextové gramatiky G_1 a G_2 .
--

Otázka: Generují G_1 a G_2 stejný jazyk?

Definice

Nechť $f : \Sigma^* \rightarrow \Sigma^*$ je funkce.

Říkáme, že (deterministický) *TS* *vyčísluje funkci* f , pokud pro každé vstupní slovo $w \in \Sigma^*$ zapíše na pásku $f(w)$ a skončí.

Funkce f se nazývá *vyčíslitelná*, pokud existuje (deterministický) TS, který ji vyčísluje.

Poznámka: Stroji, který vyčísluje funkci se také říká *algorithmus*. Toto je v souladu s tím, jak známe tento pojem z **KMI/ALM1**.

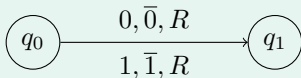
Příklad

TS vyčísлюjící funkci $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ zobrazující $w \mapsto 0w$ (tedy přidání počáteční nuly).

- 1 Označíme začátek pásky (první symbol).
- 2 Najdeme konec vstupního slova.
- 3 Dokud nepřesuneme označené symboly, přesouváme každý symbol vstupního slova o jedno pole doprava.
- 4 Přidáme počáteční nulu.

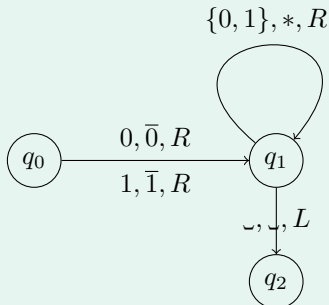
Příklad

- 1 Označíme začátek pásky (první symbol).
- 2 Najdeme konec vstupního slova.
- 3 Dokud nepřesuneme označené symboly, přesouváme každý symbol vstupního slova o jedno pole doprava.
- 4 Přidáme počáteční nulu.



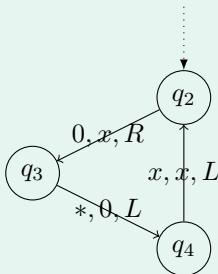
Příklad

- 1 Označíme začátek pásky (první symbol).
- 2 Najdeme konec vstupního slova.
- 3 Dokud nepřesuneme označené symboly, přesouváme každý symbol vstupního slova o jedno pole doprava.
- 4 Přidáme počáteční nulu.



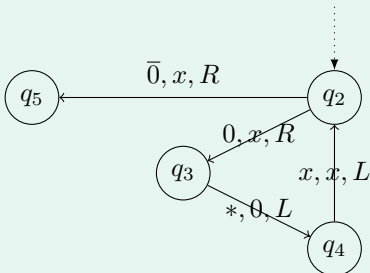
Příklad

- 1 Označíme začátek pásky (první symbol).
- 2 Najdeme konec vstupního slova.
- 3 Dokud nepřesuneme označené symboly, přesouváme každý symbol vstupního slova o jedno pole doprava.
- 4 Přidáme počáteční nulu.



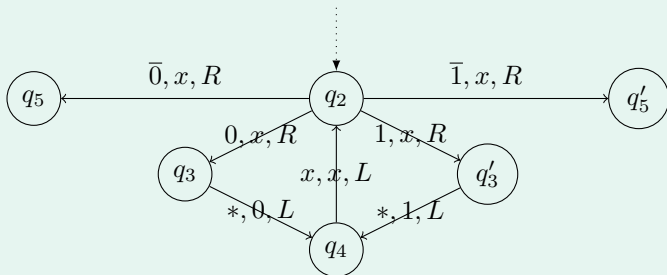
Příklad

- 1 Označíme začátek pásky (první symbol).
- 2 Najdeme konec vstupního slova.
- 3 Dokud nepřesuneme označené symboly, přesouváme každý symbol vstupního slova o jedno pole doprava.
- 4 Přidáme počáteční nulu.



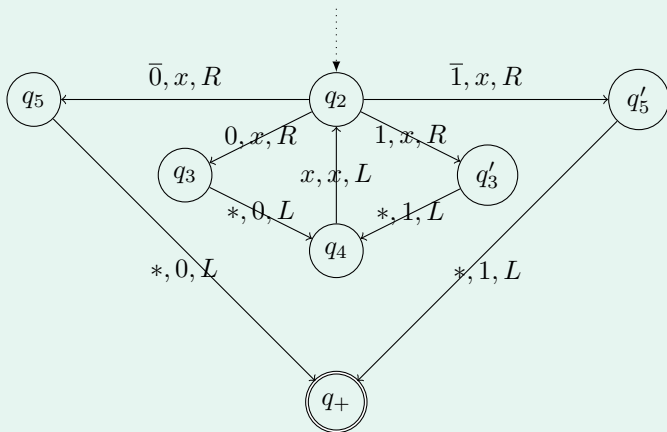
Příklad

- 1 Označíme začátek pásky (první symbol).
- 2 Najdeme konec vstupního slova.
- 3 Dokud nepřesuneme označené symboly, přesouváme každý symbol vstupního slova o jedno pole doprava.
- 4 Přidáme počáteční nulu.



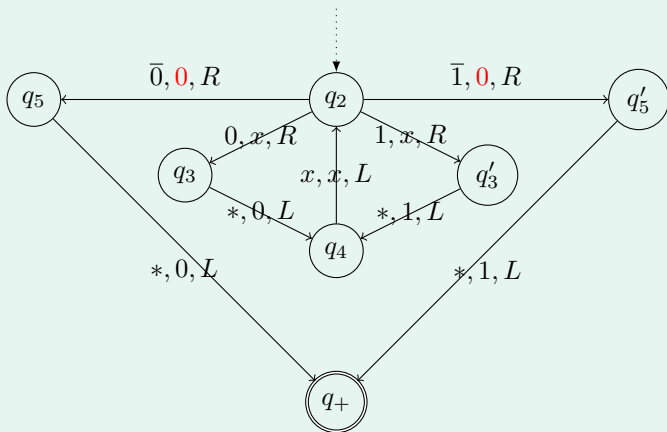
Příklad

- 1 Označíme začátek pásky (první symbol).
- 2 Najdeme konec vstupního slova.
- 3 Dokud nepřesuneme označené symboly, přesouváme každý symbol vstupního slova o jedno pole doprava.
- 4 Přidáme počáteční nulu.



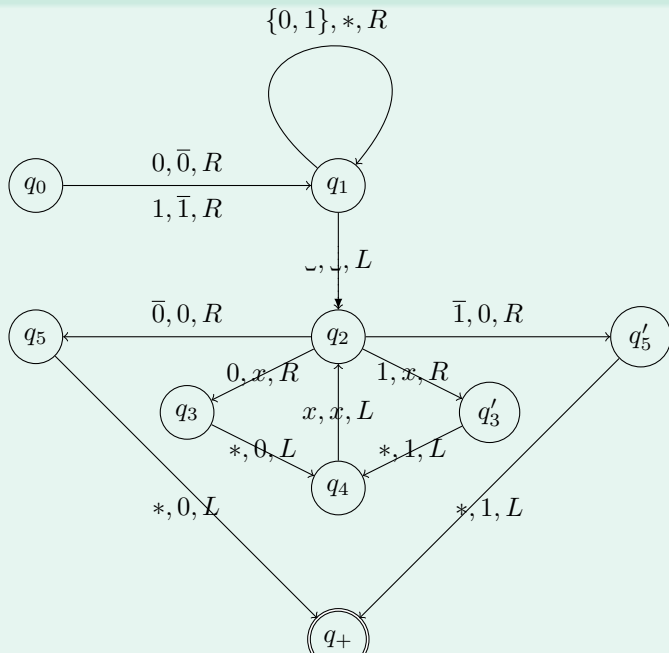
Příklad

- 1 Označíme začátek pásky (první symbol).
- 2 Najdeme konec vstupního slova.
- 3 Dokud nepřesuneme označené symboly, přesouváme každý symbol vstupního slova o jedno pole doprava.
- 4 Přidáme počáteční nulu.



Příklad

Celý stroj:



Zakódování TS

TS T do řetězce $[T]$.

Q Předpokládejme, $Q = \{q_1, q_2, \dots, q_n\}$.

Množinu zakódujeme jako: 0^n .

Σ a Γ : Předpokládejme,

$\Sigma = \{a_1, a_2, \dots, a_m\}$, $\Gamma = \Sigma \cup \{a_m + 1, \dots, a_{m+p}\}$ a $a_{m+p} = \sqcup$.

Kódujeme jako: $0^m 10^p$.

δ : Každý přechod $\delta(q_i, a_j) = (q_l, a_k, X)$ zakódujeme jako

$$0^i 10^j 10^l 10^k 10^x, \text{ kde } x = \begin{cases} 1 & \text{pokud } X = L, \\ 2 & \text{pokud } X = R. \end{cases}$$

Celou přechodovou funkci pak zakódujeme jako zřetězení zakódování jednotlivých přechodů oddělených dvěma jedničkami 11.

q_{start} nekódujeme, uvažujeme $q_{\text{start}} = q_1$.

q_+ zakódujeme jako 0^e takže $q_+ = q_e$.

q_- zakódujeme jako 0^f takže $q_- = q_f$.

Zakódování celého TS je pak zřetězení všech těchto částí oddělených třemi jedničkami 111: tedy

$$[Q]111[\Sigma, \Gamma]111[\delta]111[q_+]111[q_-]$$

Zakódování TS $M_1 = \langle Q, \Sigma, \Gamma, \delta, q_0 = q_1, q_+ = q_6, q_- = q_7 \rangle$, kde

- $Q = \{q_1, \dots, q_7\}$,
- $\Sigma = \{0 = a_1\}$,
- $\Gamma = \Sigma \cup \{x = a_2, \sqcup = a_3\}$,
- δ je dána následující tabulkou:

	0	$\sqcup$	x
q_1	$(q_2, \sqcup, R)$	$(q_7, \sqcup, R)$	(q_7, x, R)
q_2	(q_3, x, R)	$(q_6, \sqcup, R)$	(q_2, x, R)
q_3	$(q_4, 0, R)$	$(q_5, \sqcup, L)$	(q_3, x, R)
q_4	(q_3, x, R)	$(q_6, \sqcup, R)$	(q_4, x, R)
q_5	$(q_5, 0, L)$	$(q_2, \sqcup, R)$	(q_5, x, L)

0000000111

0100111

010100100010011010001000000010001001101001000000010010011

001010001001001100100010000001000100110010010010010011

00010100001010011000100010000010001011000100100010010011

0000101000100100110000100010000010001001100001001000010010011

0000010100000101011000001000100100010011000001001000001010111

0000001110000000

- To samozřejmě není jediný způsob jak kódovat TS.
- K jednomu stroji existuje několik kódů (např. přechody mohou být kódované v libovolném pořadí)

Důsledkem toho, že TS lze kódovat do řetězců je: ČR je spočetně nekonečná, R je spočetně nekonečná.

Víme, že jazyků nad jakoukoli abecedou je nespočetně mnoho...

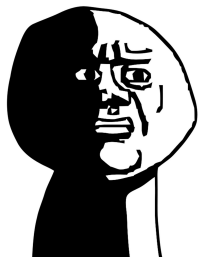
protože kdyby ne, šly by očíslovat, seřadit do řádků tabulky, jako je tahle (sloupce reprezentují řetězce, x znamená přítomnost řetězce v jazyce)

	s_1	s_2	s_3	s_4	...
L_1	x				
L_2	x		x		
L_3		x	x		
L_4	x			x	
$\vdots$					

Jazyk $L_? = \{s_n \mid s_n \notin L_n\}$ nemůže být v žádném řádku tabulky – spor.

To mimo jiné znamená, že

- neřešitelných problémů je nekonečněkrát víc než řešitelných.
- náhodně vybraný problém je řešitelný s pravěpodobností 0.



ALMOST NO PROBLEM CAN BE SOLVED!

- naštěstí problémy, které jsou důležité v praxi, mají obvykle jasně specifikované zadání a obvykle řešitelné jsou.

Kódování slov, konfigurací a výpočtů.

- slova (nad abecedou $\Sigma = \{a_1, a_2, \dots, a_m\}$)

$$[a_{i_1} a_{i_2} \dots a_{i_n}] = 0^{i_1} 10^{i_2} 1 \dots 10^{i_n}$$

např.: $[a_1 a_2 a_2 a_3 a_1] = 0100100100010$

- konfigurace

$$(q, \alpha, n) = [q]11[\alpha]110^n$$

- výpočet $k_1 \vdash k_2 \vdash \dots \vdash k_n$

$$[k_1]111[k_2]111 \dots 111[k_n]$$

Poznámka: doted' jsme v žádném zakódování neměli víc, jak tři jedničky po sobě. čtyři jedničky budeme používat jako oddělovač.

Např. zakódování TS M se vstupním slovem w , tedy $[M, w] = [M]1111[w]$.
Tímto způsobem můžeme „přdat Turingovu stroji více vstupů“.

Church-Turingova Teze

Intuitivní pojem algoritmu = algoritmus TS.

cvičení 1

- 1 Navrhňte TS, který rozhoduje jazyk 0^{3m+1} .
(hint: toto je regulární jazyk vzpomeňte si na kurz FJAA)
- 2 Navrhňte TS vyčísлюjící *nulovou funkci* $o(n) = 0$ pro každé $n \in \mathbb{N}_0$.
- 3 Navrhňte TS vyčísлюjící *funkci následníka* $s(n) = n + 1$ pro každé $n \in \mathbb{N}_0$.
- 4 Navrhňte TS vyčísлюjící *funkci projekce* $p(x, y, z) = y$ pro každé $x, y, z \in \mathbb{N}_0$.

Na co byste měli být schopní odpovědět: Jak je definován TS, výpočet, jak TS přijímá a zamítá slova, kdy TS cyklí, co jsou rekurzivní jazyky, částečně rekurzivní jazyky. Co je rozhodovací problém, co je instance problému, a jak to souvisí s jazyky. Co je řešitelnost problému a částečná řešitelnost problému. Jak vypadá grafová reprezentace TS, jak dá zakódovat TS, jak se dá zakódovat slovo, konfigurace, výpočet? Kolik je kterých problémů a proč má informatik depresi?

Reading assignment: Pročtěte si někde, kdo to byli Alan Mathison Turing, Alonzo Church, Stephen Cole Kleene, Emil Leon Post.