Kapitola 2

Turingovy stroje

V této části textu se budeme zabývat tím, které úlohy je možné řešit mechanickým výpočtem (například na počítači) a které ne. To že musí existovat problémy, které řešit nelze, vyplývá z následující jednoduché úvahy.

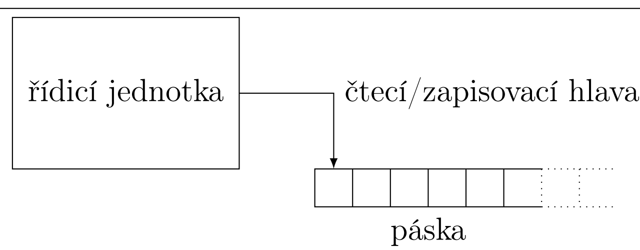
Proč musí existovat problémy, které počítačový program neumí řešit? Uvažujme tento typ problému: Patří zadané slovo ω do množiny $L \subseteq \Sigma^*$? Těchto *problémů je nespočetné nekonečno*, protože podmnožin L množiny Σ^* je nespočetné nekonečno. Naopak *programů je spočetné nekonečno*, protože programy můžeme považovat za řetězce nad nějakou abecedou (třeba nad ASCII).

My tuto problematiku budeme zkoumat ne na počítačích a počítačových programech, ale na jednoduchém formalismu – Turingově stroji.

2.1 Turingovy stroje

Turingův stroj (TS) je teoretický stroj, který 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.



Obrázek 2.1: Schéma Turingova stroje

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.“

Takovou instrukci nazývat přechod. Celý program, tedy množinu takovýchto instrukcí, pak nazýváme přechodovou funkcí Turingova stroje.

2.2 Formální definice TS a výpočet TS

Definice 1. *Turingův stroj* je dán:

1. neprázdnou konečnou množinou stavů Q,
2. vstupní abecedou Σ, t.ž. $\llcorner\!\lrcorner \notin \Sigma$,
3. páskovou abecedou Γ, t.ž. $\Sigma \subset \Gamma$,
4. přechodovou funkcí $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$,
5. počátečním stavem $q_0 \in Q$,
6. přijímacím stavem $q_+ \in Q$,
7. zamítacím stavem $q_- \in Q$, t.ž. $q_- \neq q_+$.

Formálně budeme TS zapisovat jako strukturu $\langle Q, \Sigma, \Gamma, \delta, q_0, q_+, q_- \rangle$.

Příklad 1. V tomto příkladu uvedeme M_1, který budeme v používat jako příklad v následujícím textu.
TS $M_1 = \langle Q, \Sigma, \Gamma, \delta, q_0, q_+, q_- \rangle$, kde

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_+ q_-\}$,
- $\Sigma = \{0\}$,
- $\Gamma = \Sigma \cup \{\llcorner\!\lrcorner, \mathrm{x}\}$,
- δ je dána následující tabulkou:

	0	$\llcorner\!\lrcorner$	x
q_0	$(q_1, \llcorner\!\lrcorner, R)$	$(q_-, \llcorner\!\lrcorner, R)$	(q_0, x, R)
q_1	$(q_2, 0, R)$	$(q_+, \llcorner\!\lrcorner, R)$	(q_1, x, R)
q_2	$(q_3, 0, R)$	$(q_4, \llcorner\!\lrcorner, L)$	(q_2, x, R)
q_3	(q_2, x, R)	$(q_-, \llcorner\!\lrcorner, R)$	(q_3, x, R)
q_4	$(q_4, 0, L)$	$(q_0, \llcorner\!\lrcorner, R)$	(q_4, x, L)

Definice 2. *Konfigurace* TS $T = \langle Q, \Sigma, \Gamma, \delta, q_0, q_+, q_- \rangle$ je uspořádaná trojice $(q, \alpha, n) \in Q \times \Gamma^* \times \mathbb{N}_0$.

- *iniciální konfigurace* TS T pro vstup $\omega \in \Sigma^*$ je $(q_{\text{start}}, \omega, 0)$; budeme ji označovat $\mathcal{C}_0^\omega$
- *přijímající konfigurace* TS T je konfigurace (q_+, α, n), kde $\alpha \in \Gamma^*, n \in \mathbb{N}_0$. Libovolnou přijímající konfiguraci budeme označovat $\mathcal{C}^+$.
- *zamítající konfigurace* TS T je konfigurace (q_-, α, n), kde $\alpha \in \Gamma^*, n \in \mathbb{N}_0$. Libovolnou zamítající konfiguraci budeme označovat $\mathcal{C}^-$.

Konfiguraci, která je přijímající nebo zamítající, říkáme *koncová konfigurace*. Konfigurace, která není ani přijímající ani zamítající, říkáme *nekoncová konfigurace*.

Konfigurace TS T je trojice, která zachycuje aktuální status všech tří komponent Turingova stroje. q v trojici (q, α, n) je aktuální stav řídící jednotky, α je obsah pásky a n je pozice hlavy[1]. Při iniciální konfiguraci TS T se tedy řídící jednotka nachází v počátečním stavu q_0, na pásce je zapsáno vstupní slovo (následované nekonečným počtem prázdných symbolů $\llcorner\!\lrcorner$) a hlava je nad nejlevějším políčkem pásky (s indexem 0). Přijímající konfigurace je tedy jakákoli konfigurace, ve které je řídící jednotka v přijímacím stavu, a zamítající konfigurace je jakákoli konfigurace, ve které je řídící jednotka v zamítacím stavu.

[1]pozice hlavy na pásce budeme číslovat od 0

Poznámka 1. 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. Analogicky s předchozím zápisem, konfigurace $\alpha q\beta$ a $\alpha q\beta\llcorner\!\lrcorner$ považujeme za stejné.

Definice 3. Krok výpočtu TS je definován jako binární relace na množině konfigurací: Nechť $(q, a_1 \dots a_n, i)$ je taková konfigurace T, kde $q \neq q_\pm, n \in \mathbb{N}_0, a_0, \dots, a_{n-1} \in \Gamma, i < n$.

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

$$(q, a_0 \dots a_{n-1}, i) \vdash (q', a_1 \dots a_{i-1} b a_{i+1} \dots a_{n-1}, i - 1).$$

(b) Je-li $\delta(q, a_0) = (q', b, \mathrm{L})$, pak

$$(q, a_0 \dots a_{n-1}, 0) \vdash (q', b a_1 \dots a_{n-1}, 0).$$

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

$$(q, a_0 \dots a_{n-1}, i) \vdash (q', a_1 \dots a_{i-1} b a_{i+1} \dots a_{n-1}, i + 1).$$

Definice 15 říká, jakým způsobem lze z nekoncové konfigurace odvodit novou podle přechodové funkce δ. Pokud jsme ve stavu q a hlava čte páskový symbol a_i, a máme $\delta(q, a_i) = (q', b, \mathrm{X})$, pak:

- řídící 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 (položky (a), (b) a (c) souhlasí se stejně označenými položkami v Definici 15):
  (a) Pokud X = L a hlava není nad nejlevějším políčkem pásky, pohne se 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.

Zápis $\mathcal{C} \vdash \mathcal{C}'$ čteme takto: konfigurace $\mathcal{C}'$ *je odvoditelná* z konfigurace $\mathcal{C}$ *jedním krokem výpočtu.*

Definice 4. *Výpočet* $\vdash^*$ TS je reflexivní, tranzitivní uzávěr relace $\vdash$.
Zápis $\mathcal{C} \vdash^* \mathcal{C}'$ jako: „konfigurace $\mathcal{C}'$ *je odvoditelná* z konfigurace $\mathcal{C}$.“

Definice 5. Turingův stroj T *přijímá* $\omega \in \Sigma^*$, pokud $\mathcal{C}_0^\omega \vdash^* \mathcal{C}_+$, kde $\mathcal{C}_0^\omega$ je iniciální konfigurace T pro vstup ω a $\mathcal{C}_+$ je přijímající konfigurace stroje T.

Turingův stroj T *zamítá* $\omega \in \Sigma^*$, pokud $\mathcal{C}_0^\omega \vdash^* \mathcal{C}_-$, kde $\mathcal{C}_0^\omega$ je iniciální konfigurace T pro vstup ω a $\mathcal{C}_-$ je zamítající konfigurace stroje T.

Pro přehlednější zápis konfigurace (q, α, i) budeme podtrhávat i-tý symbol v řetězci α (tj. symbol, nad kterým je hlava). Budeme tedy například zapisovat $(q_1, 11\underline{0}00, 2)$ místo $(q_1, 11000, 2)$.

Definice 6. *Historie výpočtu* TS T nad slovem ω je maximální posloupnost konfigurací $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \dots$ TS T tak, že $\mathcal{C}_0 = \mathcal{C}_0^\omega$ a pro všechna $i \in \mathbb{N}$ platí $\mathcal{C}_{i-1} \vdash \mathcal{C}_i$.

Použitím pojmu historie výpočtu můžeme ekvivalentně definovat přijímání, zamítání a cyklení Turingova stroje.

Definice 7. Turingův stroj T

- *přijímá* $\omega \in \Sigma^*$, pokud je historie výpočtu TS T nad slovem ω konečná a končí přijímající konfigurací.

- *zamítá* $\omega \in \Sigma^*$, pokud je historie výpočtu TS T nad slovem ω konečná a končí zamítající konfigurací.
- *cyklí* pro slovo $\omega \in \Sigma^*$, pokud je historie výpočtu TS T nad slovem ω nekonečná.

Příklad 2. Iniciální konfigurace M_1 pro vstupní slovo $\omega = 00000$ je $(q_0, \underline{0}0000, 0)$. Z této konfigurace je v jedním krokem výpočtu odvoditelná konfigurace $(q_1, \sqcup\underline{0}000, 1)$. Celá historie výpočtu nad slovem ω by vypadala takto:

$(q_0, \underline{0}0000, 0) \vdash (q_1, \sqcup\underline{0}000, 1) \vdash (q_2, \sqcup 0\underline{0}00, 2) \vdash (q_3, \sqcup 0\mathrm{x}\underline{0}0, 3) \vdash (q_2, \sqcup 0\mathrm{x}0\underline{0}, 4) \vdash (q_3, \sqcup 0\mathrm{x}0\mathrm{x}\underline{\sqcup}, 5) \vdash (q_-, \sqcup 0\mathrm{x}0\mathrm{x}\sqcup\underline{\sqcup}, 6)$

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

Historie výpočtu TS M_1 pro vstupní slovo $\omega = 0000$ by vypadal takto: $(q_0, \underline{0}000, 0) \vdash (q_1, \sqcup\underline{0}00, 1) \vdash (q_2, \sqcup 0\underline{0}0, 2) \vdash (q_3, \sqcup 0\mathrm{x}\underline{0}, 3) \vdash (q_2, \sqcup 0\mathrm{x}0\underline{\sqcup}, 4) \vdash (q_4, \sqcup 0\mathrm{x}\underline{0}, 3) \vdash (q_4, \sqcup 0\underline{\mathrm{x}}0, 2) \vdash (q_4, \sqcup\underline{0}\mathrm{x}0, 1) \vdash (q_4, \underline{\sqcup}0\mathrm{x}0, 0) \vdash (q_0, \sqcup\underline{0}\mathrm{x}0, 1) \vdash (q_1, \sqcup\sqcup\underline{\mathrm{x}}0, 2) \vdash (q_1, \sqcup\sqcup\mathrm{x}\underline{0}, 3) \vdash (q_2, \sqcup\sqcup\mathrm{x}0\underline{\sqcup}, 4) \vdash (q_4, \sqcup\sqcup\mathrm{x}\underline{0}\sqcup, 3) \vdash (q_4, \sqcup\sqcup\underline{\mathrm{x}}0\sqcup, 2) \vdash (q_4, \sqcup\underline{\sqcup}\mathrm{x}0\sqcup, 1) \vdash (q_0, \sqcup\sqcup\underline{\mathrm{x}}0\sqcup, 2) \vdash (q_0, \sqcup\sqcup\sqcup\underline{0}\sqcup, 3) \vdash (q_1, \sqcup\sqcup\sqcup\sqcup\underline{\sqcup}, 4) \vdash (q_+, \sqcup\sqcup\sqcup\sqcup\sqcup\underline{\sqcup}, 5)$.

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

Používaje alternativní zápis konfigurací bychom mohli historii výpočtu zapsat takto: $q_0\underline{0}000 \vdash \sqcup q_1\underline{0}00 \vdash \sqcup 0q_2\underline{0}0 \vdash \sqcup 0\mathrm{x}q_3\underline{0} \vdash \sqcup 0\mathrm{x}0q_2\underline{\sqcup} \vdash \sqcup 0\mathrm{x}q_4\underline{0} \vdash \sqcup 0q_4\underline{\mathrm{x}}0 \vdash \sqcup q_4\underline{0}\mathrm{x}0 \vdash q_4\underline{\sqcup}0\mathrm{x}0 \vdash \sqcup q_0\underline{0}\mathrm{x}0 \vdash \sqcup\sqcup q_1\underline{\mathrm{x}}0 \vdash \sqcup\sqcup\mathrm{x}q_1\underline{0} \vdash \sqcup\sqcup\mathrm{x}0q_2\underline{\sqcup} \vdash \sqcup\sqcup\mathrm{x}q_4\underline{0}\sqcup \vdash \sqcup\sqcup q_4\underline{\mathrm{x}}0\sqcup \vdash \sqcup q_4\underline{\sqcup}\mathrm{x}0\sqcup \vdash \sqcup\sqcup q_0\underline{\mathrm{x}}0\sqcup \vdash \sqcup\sqcup\sqcup q_0\underline{0}\sqcup \vdash \sqcup\sqcup\sqcup\sqcup q_1\underline{\sqcup} \vdash \sqcup\sqcup\sqcup\sqcup\sqcup q_+\underline{\sqcup}$.

2.3 Jazyky přijímané a jazyky rozhodované Turingovými stroji

Definice 8. Množinu všech slov $w \in \Sigma^*$, které TS T přijímá, značíme $L(T)$. Jazyk $L(T)$ nazýváme *jazyk přijímaný TS T* a říkáme, že TS T *přijímá jazyk* $L(T)$.

Pokud navíc platí, že TS T zamítá každé slovo, které nepatří do $L(T)$, nazýváme jazyk $L(T)$ *jazyk rozhodovaný TS T* a říkáme, že TS T *rozhoduje jazyk* $L(T)$.

Definice 9. Jazyk $L \subseteq \Sigma^*$ nazveme *jazyk rozhodovaný TS*, pokud existuje TS T, který jej rozhoduje.

Jazyk $L \subseteq \Sigma^*$ nazveme *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*.

Příklad 3. Například $A = \{0^{2^n} \mid n \geq 0\}$ je jazyk rozhodovaný TS, protože M_1 jej rozhoduje.

Namísto pojmu jazyk je v teorii vyčíslitelnosti běžné používat pojem *rozhodovací problém* (též jen *problém*). Tímto pojmem budeme myslet otázku, na kterou se odpovídá „ano“ nebo „ne“. *Instancí problému* pak myslíme otázku s konkrétními hodnotami, pro které je otázka pokládána. Podle odpovědi budeme rozlišovat *instance s odpovědí „ano“* a *instance s odpovědí „ne“*. Rozhodovacím problémem je například otázka: „Je dané číslo x menší než dané číslo y?“. Instanci tohoto problému (s odpovědí „ano“ a s odpovědí „ne“ v tomto pořadí) jsou třeba otázky „Je číslo 3 menší než číslo 7? Je číslo 15 menší než číslo 2?“.

Pojem jazyk a pojem problém lze považovat za totéž[2]:

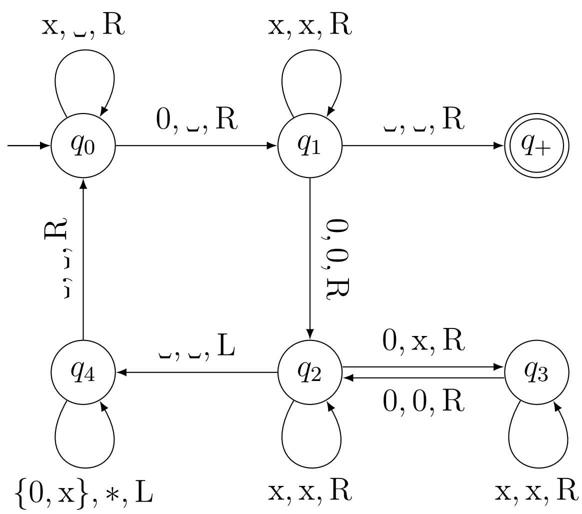
- Problém P můžeme chápat jako jazyk

$$\{x \mid x \text{ je instancí } P \text{ s odpovědí „ano“}\}$$

- Jazyk L pak můžeme chápat jako jako problém

„Patří slovo x do jazyka L?“

[2] Ve skutečnosti toto platí jen, pokud libovolnou instanci problému můžeme reprezentovat řetězci. V tomto textu se budeme zabývat jen takovými problémy.



Obrázek 2.2: Přechodový diagram Turingova stroje M_1 z příkladu 1.

Také pak o jazycích rozhodovaných TS můžeme mluvit jako o *řešitelných problémech* (na TS), a o jazycích přijímaných TS jako o *problémech částečně řešitelných* (na TS). V tomto textu se budeme držet spíše pojmu jazyk a s ním spjatých pojmů.

V následující části této kapitoly budu také používat pojem vyčíslitelná funkce, který zavedeme takto:

Definice 10. Nechť $f : \Sigma^* \to \Sigma^*$ 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 2. Vyčíslitelné funkci se říká také *algoritmus*.

2.4 Grafová representace TS (přechodový diagram TS)

Přechody a stavy TS můžeme zakreslit do grafu, tzv. *přechodového diagramu* (podobně jako je tomu u konečných automatů nebo zásobníkových automatů). Přechodový diagram se skládá z množiny uzlů, která koresponduje s množinou Q stavů TS. Každá hrana vedoucí z uzlu q do uzlu p je nadepsána jednou nebo více položkami ve tvaru „a, b, D“, kde $a, b \in \Gamma$ jsou páskové symboly a $D \in \{\mathrm{L}, \mathrm{R}\}$ je směr pohybu hlavy. Tyto položky odpovídají přechodům TS: položka $\langle a, b, D\rangle$ hrany vedoucí z uzlu q do uzlu p odpovídá přechodu $\delta(q, X) = (p, Y, D)$.

Uzel odpovídající počátečnímu stavu q_0 se označuje malou šipkou vedoucí do tohoto uzlu, Uzel odpovídající přijímajícímu stavu q_+ se označuje dvojitým ohraničením. Uzel odpovídající zamítacímu stavu q_- (a hrany do něj vedoucí) je obvyklé vůbec nezobrazovat a předpokládat, že všechny přechody, které nejsou zobrazeny v grafové reprezentaci, vedou do tohoto stavu.

Obrázek 2.2 ukazuje grafovou reprezentaci TS M_1 z příkladu 1.

Poznámka 3. V následujících kapitolách budeme často v representaci TS používat abstrakce – část stroje vyjmeme a nazveme subrutinou. Subrutinu pak zakreslujeme jako uzel ve tvaru obdélník. Pro jednoduchost u přechodů jdoucích do subrutiny a vycházejících ze subrutiny budeme uvažovat, že nečtou z pásky, nezapisují na pásku a nepohybují hlavou.

2.5 Příklad TS

V této sekci sestavíme stroj, který rozhoduje jazyk

$$L = \{\alpha \in \{0,1\}^* \mid \alpha \text{ obsahuje stejně 0 jako 1 }\}.$$

Na tento příklad pak budeme navazovat v následujících kapitolách.

Sestavovaný TS bude fungovat takto:

krok 0. Stroj najde nejlevější 0 nebo 1. Pokud takový symbol neexistuje, TS přejde do přijímacího stavu. Přepíše nalezený symbol na x a pokračuje krokem 2.

krok 1. Najde nejlevější opačný symbol (pokud předtím přepsal 0, najde 1, a naopak). Pokud takový symbol neexistuje, TS přejde do zamítacího stavu. Přepíše nalezený symbol na y a pokračuje krokem 3.

krok 2. Vrátí se na nejpravější symbol x a pokračuje bodem 1.

2.5.1 Konstrukce TS

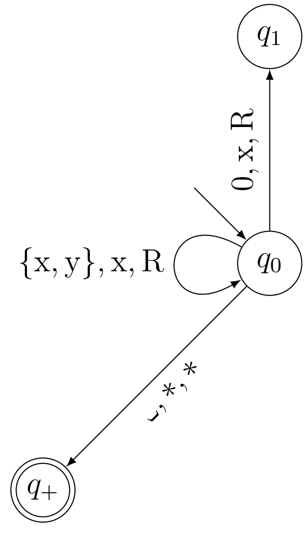
Krok 0 je reprezentován počátečním stavem q_0. Přechody (smyčka na obrázku 2.3)

$$\delta(q_0, \mathrm{y}) = (q_0, \mathrm{x}, \mathrm{R}) \qquad \delta(q_0, \mathrm{x}) = (q_0, \mathrm{x}, \mathrm{R})$$

reprezentují hledání prvního symbolu, který není x nebo y. Pokud jako první takový symbol byl nalezen prázdný symbol ␣, znamená to, že na pásce nezbývá žádná nezpracovaná 0 nebo 1, a můžeme přejít do přijímajícího stavu. To je realizováno přechodem

$$\delta(q_0, ␣) = (q_+, ␣, \mathrm{R}).$$

Pokud je nalezen symbol 0, přejde stroj do stavu q_1, který realizuje krok 1.



Obrázek 2.3: Realizace kroku 0

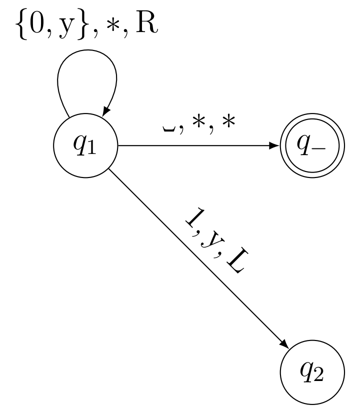
Krok 1 je reprezentován stavem q_1 (prozatím předpokládáme, že symbol přepsaný v předchozím kroku je 0). Stroj se pokusí najít nejlevější symbol 1. Přejede všechna políčka pásky na kterých je 0 nebo y, to je realizováno přechody (smyčka na obrázku 2.4)

$$\delta(q_1, 0) = (q_0, 0, \mathrm{R}) \qquad \delta(q_1, \mathrm{y}) = (q_0, \mathrm{y}, \mathrm{R}).$$

Pokud je první symbol, který není 0 nebo y, prázdný symbol, znamená to, že na pásce není 1, která by odpovídala přepsané nule z předchozího kroku. Stroj tedy přejde do zamítacího stavu použitím přechodu

$$\delta(q_1, \smile) = (q_-, \smile, \mathrm{R}).$$

Pokud je nalezen symbol 1, přejdeme ke kroku 2.



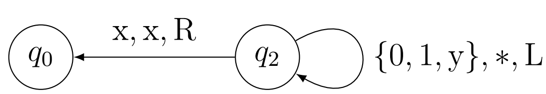
Obrázek 2.4: Realizace kroku 1

Krok 2 je realizován stavem q_2. V tomto kroku se vracíme na levý okraj pásky. Přesněji na první symbol zprava, který není x. To je realizováno přechody (smyčka na obrázku 2.5)

$$\delta(q_2, 0) = (q_2, 0, \mathrm{L}), \qquad \delta(q_2, 1) = (q_2, 1, \mathrm{L}), \qquad \delta(q_2, \mathrm{y}) = (q_2, \mathrm{y}, \mathrm{L}).$$

Jakmile najdeme x, přejdeme do kroku 0, realizovaného stavem q_0, požitím přechodu

$$\delta(q_2, \mathrm{x}) = (q_0, \mathrm{x}, \mathrm{R}).$$



Obrázek 2.5: Realizace kroku 2

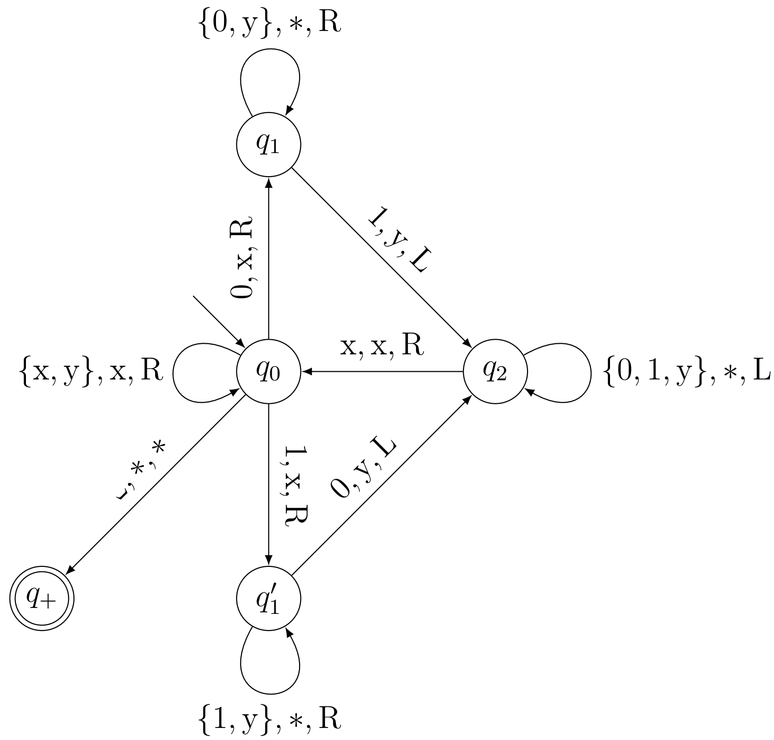
2.5.2 Kompletní stroj

Obrázek 3.2 zobrazuje celý diagram TS. Oproti složení částí s předchozí sekce je doplněna větev, kde symbol nalezený v kroku 1 je 1. A je skryt zamítací stav.

2.6 Zakódování TS

V této sekci demonstrujeme, jak by bylo možné zakódovat libovolný TS T do řetězce $[T]$. Existuje nekonečně mnoho způsobů jak zakódovat TS, my si uvedeme tento jeden:

neprázdná konečná množina stavů Q: Jelikož nezáleží na konkrétním označení jednotlivých stavů, předpokládejme, $Q = \{q_1, q_2, \ldots, q_n\}$. Množinu zakódujeme jako: 0^n, tedy řetězec nul o délce $n = |Q|$.



Obrázek 2.6: Přechodový diagram Turingova stroje, který rozhoduje jazyk $L = \{\alpha \in \{0,1\}^* \mid \alpha$ obsahuje stejně 0 jako 1 $\}$

vstupní abeceda Σ a pásková abeceda Γ: Jelikož nezáleží na konkrétním označení jednotlivých stavů, předpokládejme, $\Sigma \subset \Gamma = \{a_1, a_2, \ldots, a_m\}$. Množinu Γ zakódujeme jako: 0^m, tedy řetězec nul o délce $m = |\Gamma|$.

přechodové funkce δ: Jednotlivé přechody $\delta(q_i, a_j) = (q_l, a_k, \mathrm{X})$ zakódujeme jako: $0^i 1 0^j 1 0^l 1 0^k 1 0^x$, kde

$$x = \begin{cases} 1 & \text{pokud } \mathrm{X} = \mathrm{L}, \\ 2 & \text{pokud } \mathrm{X} = \mathrm{R}. \end{cases}$$

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

počáteční stav nekódujeme, uvažujeme že startovní stav je q_1.

přijímací stav $q_+ \in Q$ zakódujeme jako 0^e tak že $q_+ = q_e$.

zamítací stav $q_- \in Q$ zakódujeme jako 0^f tak že $q_- = q_f$.

Zakódování celého TS je pak zřetězení všech jeho částí oddělených třemi jedničkami 111, tedy

$$[Q]111[\Gamma]111[\delta]111[q_+]111[q_-].$$

Dále budeme potřebovat zakódovat konfiguraci TS (q_l, α, n). Zakódujeme stav q_l jako 0^l, α jako posloupnost zakódování jednotlivých symbolů α oddělených od sebe jednou jedničkou. Číslo n zakódujeme jako $0^{(n+1)}$. Celá konfigurace bude zakódována jako $[q_l]11[\alpha]11[n]$.

Značení $[\cdot]$ budeme používat pro zakódování čehokoli do řetězce nad vstupní abecedou TS.

Příklad 4. Kód TS M_1 je:

```
000000011101001110101001000100110100010000000010001001101001000000001001001100101000
100100110010001000000100010011001001001001100010100001010011000100010000000100010
110001001000100100110000101000100100110000100010000001000100110000100100000100100110
0000101000000101011000000100010010001001100000010010000001010111000000011100000000.
```

Poznámka 4. Všimněte si, že stejný Turingův stroj můžeme popsat různými kódy, například můžeme různými způsoby oindexovat stavy a symboly, nebo můžeme kódovat přechody v různém pořadí.

Protože Turingovy stroje můžeme kódovat řetězci, kterých je spočetné nekonečno, je také Turingových strojů spočetné nekonečno.

Poznámka 5. (Existence jazyků, které nejsou částečně rekurzivní) Všimněme si, že Turingových strojů je jen spočetné nekonečno, protože je možno je kódovat do řetězců, kterých je spočetně nekonečno. Jazyků nad libovolnou abecedou je však nespočetně mnoho. To tedy znamená, že

- rekurzivních jazyků je nekonečně-krát méně, než nerekurzivních jazyků.
- částečně rekurzivních jazyků je nekonečně-krát méně, než jazyků, které nejsou ani částečně rekurzivní.
- pravděpodobnost, že náhodně vybraný problém bude (částečně) řešitelný, je 0.

2.7 Úkoly k textu

1. Nakreslete přechodový diagram TS, který přijímá jazyk
$$\{\omega\omega \mid \omega \in \{0,1\}^*\}.$$
2. Nakreslete přechodový diagram TS, který vyčisluje funkci $\omega \mapsto \triangleright\omega$. Zajistěte, aby stroj zastavoval v konfiguraci $\triangleright q_+\omega$.
3. Nakreslete přechodový diagram TS, který vyčisluje funkci $\triangleright\omega \mapsto \varepsilon$ (předpokládáme, že vstupní slovo obsahuje vždy symbol $\triangleright$ a to pouze jako první symbol). Zajistěte, aby stroj zastavoval v konfiguraci q_+ (tj. hlava je nad nejlevějším políčkem pásky, každé políčko pásky obsahuje $\text{\textvisiblespace}$).
4. Nakreslete přechodový diagram TS, který vyčisluje funkci $\omega \mapsto \omega\omega$.

2.8 Řešení úkolů k textu