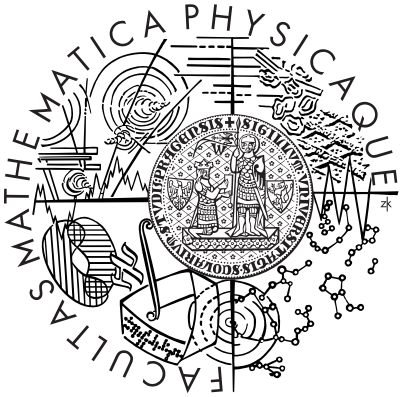
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Adéla Skoková

Různé definice Turingova stroje

Katedra algebry

Vedoucí bakalářské práce: Prof. RNDr. Jan Krajíček, DrSc.

Studijní program: Matematika, obecná matematika

2010

Ráda bych poděkovala především Prof. Krajíčkovi za téma práce a mnoho cenných rad. Dále Dr. Hladíkovi a Dr. Černému za poskytnutí literatury a dalších informací ohledně Turingových strojů. Nakonec všem ostatním, kteří mne při psaní této práce podporovali.

Prohlašuji, že jsem svou bakalářskou práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 8. května 2010 Adéla Skoková

Obsah

Název práce: Různé definice Turingova stroje
Autor: Adéla Skoková
Katedra: Katedra algebry
Vedoucí práce: Prof. RNDr. Jan Krajíček, DrSc.
E-mail vedoucího: krajicek@karlin.mff.cuni.cz

Abstrakt: V předložené práci studujeme různé varianty Turingových strojů a dokazujeme, že jsou ekvivalentní ve smyslu, že přijímají stejné jazyky. Takové stroje je možno sebou navzájem nasimulovat, neboli naprogramovat tak, aby první stroj mohl vždy ve všem zastoupit druhý stroj a naopak. V první části práce zavádíme základní definici Turingova stroje, ze které dále vycházíme. V dalších dvou kapitolách uvádíme několik alternativních definic včetně vlastní definice (Strom) a uvádíme u nich způsoby vzájemné simulace. V předposlední kapitole se věnujeme nedeterministickému Turingovu stroji a ukazujeme možnou myšlenku simulace deterministickým Turingovým strojem.

Klíčová slova: Turingův stroj: nedeterministický, vícepáskový, dvoudimenzionální; Church-Turingova teze.

Title: Various definitions of Turing machine
Author: Adéla Skoková
Department: Department of Algebra
Supervisor: Prof. RNDr. Jan Krajíček, DrSc.
Supervisor's e-mail address: krajicek@karlin.mff.cuni.cz

Abstract: In the present work we study various types of Turing machines and prove their equivalency, in the sense that they accept the same languages. Such machines can simulate each other, in the sense that we can program them to solve the same task. In the first part of this work we formulate a basic definition. Afterwards we state alternative definitions, including our own, and present mutual simulations for these. The fifth chapter is dedicated to a nondeterministic Turing machine and the idea of its simulation by a deterministic Turing machine.

Keywords: Turing machine: nondeterministic, multitape, two-dimensional; Church-Turing thesis.

Kapitola 1

Úvod

Alan Turing, britský matematik, logik a kryptolog, se narodil 26. června 1912 v Londýně, vystudoval na King's College v Cambridge a v Princetonu pod vedením Alonzo Churche. Za druhé světové války pracoval pro britské ministerstvo zahraničních věcí v dešifrovacím oddělení. V roce 1943 byl podle jeho prací sestaven jeden z prvních elektronkových počítačů, který pomáhal k dešifrování německých zpráv. Po skončení války se Turing věnoval akademické kariéře. Například umělá inteligence, jako vědní disciplína, vychází i z jeho prací. Také dnešní počítače jsou postaveny na jeho základním schématu. Turing zemřel 7. června 1954 a na jeho počest byla roku 1966 zavedena Turingova cena, která patří k nejprestižnějším oceněním v informatice.

Ve svém článku [6] zavedl výpočetní model teoretického stroje, který dnes známe pod názvem Turingův stroj. Jedná se o stroj s nekonečnou pamětí, který má pouze čtecí a zapisovací zařízení. Na tomto modelu je v současné době založeno mnoho podstatných částí teoretické informatiky.

Church-Turingova teze říká, že každý intuitivně představitelný výpočetní postup lze realizovat Turingovým strojem. Neboli můžeme říci, že každý algoritmus se dá realizovat nějakým Turingovým strojem. Tuto tezi potvrzuje mnoho výzkumů [4]. Tedy se předpokládá, že nelze sestrojit silnější model realizovatelného počítače než je Turingův stroj.

Postupně se objevily různé modifikace Turingova stroje. Předpokládáme však, že všechny mohou být naprogramovány tak, aby pro libovolný vstup daly stejný výsledek, nebo se ani jeden z nich nezastavil. Říká se, že rozeznávají stejný jazyk. Tím se budeme v této práci zabývat. Budeme prezentovat několik variant Turingových strojů, včetně vlastní varianty (Strom), a budeme ukazovat, že se tyto stroje umí nasimulovat sebou navzájem. Nebudeme se však věnovat časové ani paměťové složitosti těchto strojů a vzájemných simulací. Dále ani výstupům, které tyto stroje mohou podávat. Bude nás zajímat pouze to, zda se Turingův stroj zastaví v jednom z předem určených konečných stavů anebo se nezastaví vůbec.

Kapitola 2

Základní definice Turingova stroje

Nejprve uvedeme několik základních pojmů potřebných k definici Turingova stroje, kterou budeme nazývat *základní* a se kterou budeme dále pracovat. Vycházíme z definic v [5].

2.1 Abecedy a slova

Abecedou rozumíme konečnou neprázdnou množinu. Prvky této množiny se nazývají symboly.

Slovem Γ délky l nad abecedou Σ pak rozumíme uspořádanou l-tici symbolů z abecedy Σ, kde l je konečné přirozené číslo. *Délku slova* značíme $|\Gamma| = l$. Například si uvedeme slovo Γ délky 5 nad abecedou Σ:

$$\Sigma = \{0, 1\}$$

$$\Gamma = 01110$$

Prázdné slovo, neboli slovo délky 0, budeme dále značit Δ. Tento symbol nepatří do žádné abecedy.

Množinu všech slov nad Σ budeme značit Σ^*:

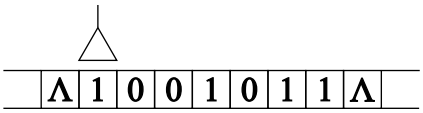
$$\Sigma^* = \{\Delta\} \cup \Sigma \cup (\Sigma \times \Sigma) \cup (\Sigma \times \Sigma \times \Sigma) \cup \cdots .$$

Jazyk nad abecedou Σ je libovolná množina slov ze Σ^*.

2.2 Turingův stroj – základní definice

Turingův stroj je jednoduchý teoretický počítač s nekonečnou pamětí, které bude v naší základní definici odpovídat oboustranně *nekonečná páska* rozdělená na jednotlivá *políčka*, někdy se také nazývají buňky, sousedící spolu vždy zleva a zprava.

Do jednotlivých políček se zapisují symboly z dané abecedy Σ, kterou budeme nazývat *pracovní abeceda* stroje. Prázdnou buňku budeme značit Λ. Platí, že Λ je symbolem pracovní abecedy, nikdy však nepatří do vstupní abecedy. *Vstupní abecedou* rozumíme abecedu Θ, jejíž slova jsou *vstupy*, na kterých Turingův stroj bude pracovat. Vždy bude platit, že vstupní abeceda je částí pracovní abecedy, tedy $\Theta \subseteq \Sigma$. Dále budeme užívat binární abecedy $\Theta = \{0, 1\}$. Možná představa Turingova stroje je uvedena na obrázku 2.1, kde hlava ukazuje na první symbol vstupu.



Obrázek 2.1: **Schéma Turingova stroje**

Hlava, neboli ukazatel, na Turingově stroji pracuje tak, že vždy ukazuje pouze na jediné z políček a neví, co se právě nachází v ostatních. Může z tohoto políčka číst i do něj zapisovat symboly z pracovní abecedy Σ. Dále se také může posunout o jedno políčko doleva nebo doprava. Zpočátku bude hlava vždy ukazovat na první políčko vstupu. Dále budeme vždy předpokládat, že vstup má nenulovou délku, tedy, že hlava zpočátku neukazuje na prázdné políčko.

Stavem q Turingova stroje budeme rozumět informaci stroje o tom, v jaké části svého programu se právě nachází. Na počátku se bude stroj nacházet ve stavu q_0. Stroj přechází do jiných stavů podle toho, co si přečte hlava na pásce a v jakém stavu se stroj právě nachází. Plní tedy funkci paměti stroje. Množinu všech stavů budeme dále značit Q.

Mějme například stroj, který zjišťuje, zda se ve vstupu z abecedy $\Theta = \{0, 1\}$ nachází řetězec tvaru 000. Hlava se zde posouvá zleva doprava. Pokud by hlava přečetla první nulu, přešel by stroj do stavu q_1, a hlava by se pak posunula na další pole. Pokud by tam našla druhou nulu, stroj by přešel do stavu q_2. Pokud by i v dalším políčku našla nulu, přešel by stroj do stavu q_{prijme}. Dostane-li se stroj do tohoto stavu, zastaví svůj výpočet a přijme daný vstup, neboli podá informaci o tom, že se v zadaném řetězci nachází podřetězec 000. Pokud by však stroj byl v jednom ze stavů q_0, q_1, q_2 a přečetl na pásce znak 1, vrátil by se do počátečního stavu q_0, ale hlava by zůstala na políčku s 1. Stroj by si nyní "nepamatoval"nic z předchozí části řetězce.

Stroj definujeme jeho **přechodovou funkcí**. Budeme jí rozumět částečnou funkci

$$\delta : Q \times \Sigma \to Q \times \Sigma \times \{L, R\},$$

kde Q je množina všech stavů Turingova stroje, Σ je pracovní abeceda stroje a $\{L, R\}$ rozumíme informaci o posunu hlavy doleva L, nebo doprava R. Tato funkce ze čteného symbolu a stavu stroje určí nový stav stroje, nebo ho ponechá stejný. Dále určí, co zapsat na pásku a kam posunout hlavu stroje. Tedy je-li $\delta(q, s) = (q', s', L)$, pak stroj ve stavu q, jehož hlava čte s, přejde do stavu q', znak s přepíše znakem s' a hlava se posune o jedno políčko doleva. Dále budeme říkat, že *ponecháme* znak, bude-li přepsán stejným znakem. Funkce není definována pro všechny dvojice $Q \times \Sigma$.

Turingův stroj se zastaví právě na těch dvojicích, pro které není definována přechodová funkce δ a naopak na těch dvojicích, pro které je definována se nezastaví. Pro stavy q_{prijme} a $q_{zamitne}$ nebude tato funkce nikdy definována a budeme je nazývat *konečné*.

Jako příklad si uveďme následující instrukci:

$$\delta(q_5, 0) = (q_1, 1, R)$$

Stroj podle této funkce přepíše symbol 0 symbolem 1, přejde ze stavu q_5 do stavu q_1 a posune hlavu o jedno políčko doprava.

Formálně Turingovým strojem rozumíme osmici:

$$(Q, \Sigma, \Theta, \Lambda, \delta, q_0, q_{prijme}, q_{zamitne}),$$

kde Q je konečná množina všech stavů Turingova stroje obsahující počáteční stav stoje q_0 a konečné stavy: q_{prijme}, $q_{zamitne}$, které budou vždy různé. Σ značí pracovní abecedu stroje, Θ vstupní abecedu[1] (Σ i Θ jsou vždy neprázdné) a platí $\Theta \subseteq \Sigma$. Dále budeme vždy brát $|\Theta| \geq 2$ a nebudeme se zabývat unárními abecedami. Znak Λ značí prázdné políčko a platí $\Lambda \in \Sigma \setminus \Theta$. δ je přechodová funkce, neboli program stroje.

Příklad: mějme Turingův stroj, který dostane na vstupu slovo z abecedy $\Theta = \{0, 1\}$ s úkolem zdvojit každý jeho znak. Například vstup bude 101 a výstup 110011. Celý výpočet je uveden v tabulce 2.1.

Mějme pracovní abecedou stroje $\Sigma = \{0, 1, \Lambda, \zeta\}$, kde Λ značí prázdnou buňku, a ζ je pomocný znak, který bude rozdělovat pásku na dvě části. Vlevo se budou nacházet již zdvojené znaky a vpravo znaky, které bude teprve potřeba zdvojit.

[1] Někdy také bývá definována jediná abeceda Σ a o vstupní abecedě se automaticky předpokládá, že je její součástí.

Zpočátku hlava ukazuje na první znak na vstupu a je ve stavu q_0. Tedy ukazuje na znak i z abecedy Θ[2]. Tento znak přepíše znakem ζ, ale i zapíše do dvou políček zleva a hlava stroje se vrátí zpět na políčko obsahující ζ.

Další kroky stroje si můžeme rozdělit do dvou bloků:

(Blok1) Stroj nyní celé slovo od znaku ζ doprava posune o políčko doprava. To udělá tak, že přejde na konec slova. Poslední znak posune o pole doprava, pak předposlední a tak postupuje, dokud nepřečte znak ζ, přepíše ho na Λ, posune se o políčko doprava a přejde do Bloku2.

Ukazuje-li hlava na políčko se znakem Λ, přejde stroj do stavu q_{prijme}, který není dále definován, a tedy výpočet končí s informací, že na pásce se nachází již zdvojnásobené slovo.

(Blok2) Pokud hlava ukazuje na jeden ze znaků vstupní abecedy Θ, zapamatuje si ho a přejde do stavu p_{11} pro 1 nebo p_{00} pro 0 a přesune se o políčko doleva, kam tento znak zapíše. Pak se posune o políčko doprava, kam opět zapíše ten samý znak. Dále se posune znovu doprava, tam zapíše znak ζ a posune se ještě o jedno políčko doprava. Výpočty stroje nyní přejdou do Bloku1.

Přechodová funkce pro tento Turingův stroj je definována takto: $i = 0, 1$; $\odot$ značí libovolný znak pracovní abecedy Σ (čte-li stroj $\odot$ a zapisuje-li ho, znamená to, že ponechá stejný znak):

$$\delta(q_0, i) = (p_i, i, L)$$

$$\delta(p_i, \odot) = (p_{ii}, \odot, L)$$

Blok1:

$$\delta(q_{posun}, \zeta) = (p_1, \Lambda, R)$$

$$\delta(q_3, i) = (q_k, i, R)$$

$$\delta(q_k, i) = (q_k, i, R)$$

$$\delta(q_k, \Lambda) = (q_{posun}, \Lambda, L)$$

Ve stavu q_{posun} je stroj připraven posunout celé slovo o políčko doprava, nejedná-li se o posun znaku ζ. Pak se vrátí o dvě políčka doleva:

$$\delta(q_{posun}, i) = (q_{posun-i}, i, R)$$

$$\delta(q_{posun-i}, \odot) = (q_{zpet}, i, L)$$

$$\delta(q_{zpet}, \odot) = (q_{posun}, \odot, L)$$

[2]Nebudeme předpokládat prázdné vstupy. Je možno je vždy snadno ošetřit, například takto: $\delta(q_0, \Lambda) = (q_{zamitne}, \Lambda, L)$

Blok2:

$$\delta(q_1, i) = (p_{ii}, i, L)$$
$$\delta(p_{ii}, \odot) = (p_{i2}, i, R)$$
$$\delta(p_{i2}, \odot) = (p_\zeta, i, R)$$
$$\delta(p_\zeta, \odot) = (q_3, \zeta, R)$$
$$\delta(q_3, \Lambda) = (q_{posun}, \Lambda, L)$$
$$\delta(q_1, \Lambda) = (q_{prijme}, \Lambda, L).$$

Nyní si celý postup ukážeme na vstupu 101 v tabulce 2.1, která je uvedena na konci kapitoly. Každý řádek tabulky značí Turingův stroj po jednom průchodu přechodovou funkcí. Tedy po přepsání symbolu původního políčka, posunu hlavy na nové políčko, přečtení nového políčka a přechodu do nového stavu. Políčko, na nějž právě ukazuje hlava, je označeno ♭. Pro názornost v prvním sloupci uvádíme stav, ve kterém se stroj právě nachází. Toto políčko však není součástí pásky. Stav stroje nepatří mezi znaky, které by se mohly na pásce vyskytovat.

Poznámka: Stav stroje není přímo znakem, který by patřil do pracovní abecedy Turingova stroje. Pokud by se ale v pracovní abecedě nacházel například symbol q_0, kterým běžně značíme počáteční stav stoje, bylo by lepší počáteční stav označit jinak. Samotný stroj však nikdy nespojuje dohromady znaky a stavy, tedy by si dvojí význam znaku q_0 vždy vyložil správně.

V literatuře je možné se setkat i s dalšími příklady různých Turingových strojů [2, 5].

Definujme dále některé pojmy související s Turingovými stroji. Na libovolném vstupu se může stát právě jedna z těchto možností:

- stroj se zastaví v jednom ze stavů q_{prijme} nebo $q_{zamitne}$
- stroj se zastaví v jiném stavu
- stroj se nezastaví

Halting problem, neboli problém zastavení stroje, je otázka, zda se Turingův stroj T, jehož přechodovou funkci známe, zastaví na daném vstupu x. Definujme si množinu dvojic stroj a vstup, na kterém se stroj zastaví:

$$\mathbf{HP} := \{[T, x] \mid T \text{ se na } x \text{ zastaví}\}$$

Alan Turing dokázal, že neexistuje obecný algoritmus řešící problém zastavení pro všechny vstupy všech programů [6].

Věta: Neexistuje Turingův stroj Q takový, že na každém vstupu skončí a platí:

- $Q([T, x])$ přijme právě když $[T, x] \in \mathbf{HP}$
- $Q([T, x])$ zamítne právě když $[T, x] \notin \mathbf{HP}$

Důkaz může čtenář nahlédnout v [1].

Řekneme, že Turingův stroj *rozhoduje* jazyk L, pokud se na každém vstupu zastaví a skončí v jednom z konečných stavů (q_{prijme}, či $q_{zamitne}$). Právě na vstupech z jazyka L přijme a na všech ostatních vstupech zamítne. Takový stroj vždy dá odpověď.

Řekneme, že Turingův stroj *přijímá* jazyk L, pokud se zastaví právě na vstupu z jazyka L. Takový stroj se nezastaví na žádném jiném vstupu. Tedy odpověď dostaneme pouze, pokud je vstup z jazyka L. Dokud probíhá výpočet stroje, nelze obecně říci, zda je vstup z jazyka L. Takový stroj tedy zřejmě nerozhoduje L.

Řekneme, že jazyk L je *Turingovsky rohodnutelný*, resp. *přijímatelný*, pokud existuje Turingův stroj, který jazyk L rozhoduje, resp. přijímá.

q_0	Λ	Λ	Λ	$\flat$ **1**	0	1	Λ	Λ	Λ
p_1	Λ	Λ	$\flat$ Λ	1	0	1	Λ	Λ	Λ
p_{11}	Λ	$\flat$ Λ	Λ	1	0	1	Λ	Λ	Λ
p_{12}	Λ	1	$\flat$ Λ	1	0	1	Λ	Λ	Λ
p_ζ	Λ	1	1	$\flat$ **1**	0	1	Λ	Λ	Λ
q_3	Λ	1	1	ζ	$\flat$ **0**	1	Λ	Λ	Λ
q_k	Λ	1	1	ζ	0	$\flat$ **1**	Λ	Λ	Λ
q_k	Λ	1	1	ζ	0	1	$\flat$ Λ	Λ	Λ
q_{posun}	Λ	1	1	ζ	0	$\flat$ **1**	Λ	Λ	Λ
$q_{posun-1}$	Λ	1	1	ζ	0	1	$\flat$ Λ	Λ	Λ
q_{zpet}	Λ	1	1	ζ	0	$\flat$ **1**	1	Λ	Λ
q_{posun}	Λ	1	1	ζ	$\flat$ **0**	1	1	Λ	Λ
$q_{posun-0}$	Λ	1	1	ζ	0	$\flat$ **1**	1	Λ	Λ
q_{zpet}	Λ	1	1	ζ	$\flat$ **0**	0	1	Λ	Λ
q_{posun}	Λ	1	1	$\flat$ ζ	0	1	1	Λ	Λ
p_1	Λ	1	1	Λ	$\flat$ **0**	1	1	Λ	Λ
p_{00}	Λ	1	1	$\flat$ Λ	0	1	1	Λ	Λ
p_{02}	Λ	1	1	0	$\flat$ **0**	1	1	Λ	Λ
p_ζ	Λ	1	1	0	0	$\flat$ **1**	1	Λ	Λ
q_3	Λ	1	1	0	0	ζ	$\flat$ **1**	Λ	Λ
q_k	Λ	1	1	0	0	ζ	1	$\flat$ Λ	Λ
q_{posun}	Λ	1	1	0	0	ζ	$\flat$ **1**	Λ	Λ
$q_{posun-1}$	Λ	1	1	0	0	ζ	1	$\flat$ Λ	Λ
q_{zpet}	Λ	1	1	0	0	ζ	$\flat$ **1**	1	Λ
q_{posun}	Λ	1	1	0	0	$\flat$ ζ	1	1	Λ
p_1	Λ	1	1	0	0	Λ	$\flat$ **1**	1	Λ
p_{11}	Λ	1	1	0	0	$\flat$ Λ	1	1	Λ
p_{12}	Λ	1	1	0	0	1	$\flat$ **1**	1	Λ
p_ζ	Λ	1	1	0	0	1	1	$\flat$ **1**	Λ
q_3	Λ	1	1	0	0	1	1	ζ	$\flat$ Λ
q_{posun}	Λ	1	1	0	0	1	1	$\flat$ ζ	Λ
p_1	Λ	1	1	0	0	1	1	Λ	$\flat$ Λ
q_{prijme}	Λ	1	1	0	0	1	1	$\flat$ Λ	Λ

Tabulka 2.1: **Postup Turingova stroje**

Kapitola 3

Jednodušší modifikace Turingových strojů

V této kapitole uvedeme jiné definice Turingových strojů, se kterými je možno se setkat v [2, 3, 5]. Vždy ukážeme, že uvedené stroje se mohou navzájem simulovat. Tedy přesněji máme definice A a B a chceme ukázat:

1. Pro každý Turingův stroj podle definice A existuje Turingův stroj podle definice B takový, že přijímá nebo rozhoduje stejný jazyk.

2. Pro každý Turingův stroj podle definice B existuje Turingův stroj podle definice A takový, že přijímá nebo rozhoduje stejný jazyk.

Budeme dále říkat, že takové stroje jsou ekvivalentní. Jsou-li dva stroje ekvivalentní, potom na všech možných vstupech podají stejný výsledek (přijmou, nebo zamítnou), nebo se ani jeden nezastaví. Oba stroje vždy budou přijímat a rozeznávat stejný jazyk.

Při modifikaci některé z předchozích definic nebudeme opět vypisovat celou formální definici, ale pouze tu část, která bude pozměněna.

Znakem $\Leftarrow$ budeme dále myslet důkaz, že některá z předchozích definic (např. přímo základní definice) jde převést (nasimulovat) na naši novou definici a $\Rightarrow$ bude naopak znamenat převod nové definice na předchozí. Jméno definice, se kterou budeme dokazovat ekvivalenci, bude vždy uvedeno na začátku důkazu.

3.1 Turingův stroj rozšířený o N

Mějme Turingův stroj, který kromě příkazů L a R má navíc ještě příkaz N. Tento příkaz bude znamenat, že hlava zůstane na stejném políčku (nikam se neposune).

Přechodová funkce bude v takovém případě vypadat takto:

$$\delta : Q \times \Sigma \to Q \times \Sigma \times \{L, N, R\}$$

Ekvivalence se základní definicí: $\Leftarrow$ Pokud N nepoužijeme v definici přechodové funkce, získáme okamžitě předchozí definici. Základní definice je speciální případ této definice.

$\Rightarrow$ Krok N bude vždy nahrazen kroky: doleva L, kde hlava nebude přepisovat dané políčko, a zpět R.

3.2 Vícekrokový Turingův stroj

Jedná se o Turingův stroj s jedinou modifikací: namísto $\{L, R\}$ máme v přechodové funkci přímo $\{nL, nR\}$, pro konečně mnoho $n \in \mathbb{N}^+$. Hlava při jednom průchodu přechodovou funkcí udělá, místo jednoho kroku daným směrem, přímo n kroků.

Můžeme mít například Turingův stroj, který ve své přechodové funkci používá pouze $\{2L, 2R, 3L, 5R\}$. Hlava tohoto stroje tedy při jednom výpočtu může dělat pouze dva, nebo tři kroky doleva, resp. dva, nebo pět kroků doprava.

Ekvivalence se základní definicí: $\Leftarrow$ Pokud v celé přechodové funkci položíme všechna $n = 1$, získáme okamžitě stroj dle základní definice.

$\Rightarrow$ Pokud se bude v přechodové funkci vícekrokového stroje nacházet:

$$\delta(q_1, a) = (q_2, b, nL), \quad n \in \mathbb{N}$$

definujeme přechodovou funkci základního stroje:

$$\delta(q_1, a) = (q_{2,n}, b, L)$$

$$\delta(q_{2,i+1}, a) = (q_{2,i}, a, L), \quad i = N-1, \ldots, 2, 1, \ a \in \Sigma$$

$$\delta(q_{2,1}, a) = (q_2, b, L), \ a \in \Sigma.$$

Oba stroje přečetly znak a, přepsali ho znakem b a posunuly hlavy o n míst doleva. Analogicky definujeme i pohyb doprava pro libovolný stav. Tímto způsobem okamžitě dostaneme simulaci vícekrokového stroje.

Příklad: Mějme Turingův stroj, který obdrží-li na vstup 8-bitové číslo, správně rozhodne, zda je liché.

Jelikož předpokládáme 8-bitové číslo, stačí pro vícekrokový stroj definovat přechodovou funkci pro $i = 0, 1$ takto:

$$\delta(q_0, i) = (q_1, i, 7L)$$

$$\delta(q_1, 0) = (q_{zamitne}, 0, R)$$
$$\delta(q_1, 1) = (q_{prijme}, 1, R)$$

Základní stroj dostane pro stejný postup instrukce:

$$\delta_z(q_j, i) = (q_{j+1}, i, R),\ j = 0, 1, \ldots, 6$$

$$\delta_z(q_7, 1) = (q_{prijme}, 1, R)$$
$$\delta_z(q_7, 0) = (q_{zamitne}, 0, R)$$

Oba stroje vykonaly stejný počet kroků hlavou, ale vícekrokový stroj použil přechodovou funkci pouze dvakrát, zatímco základní stroj ji použil sedmkrát.

Základní stroj by v tomto případě mohl mít i jednodušší přechodovu funkci. Mohl by najít konec slova a pak se vrátit na poslední znak, podle kterého by určil výsledek.

$$\delta_z(q_0, 0) = (q_0, 0, R)$$
$$\delta_z(q_0, 1) = (q_0, 1, R)$$
$$\delta_z(q_0, \Lambda) = (q_1, \Lambda, L)$$
$$\delta_z(q_1, 0) = (q_{zamitne}, 0, R)$$
$$\delta_z(q_1, 1) = (q_{prijme}, 1, R)$$

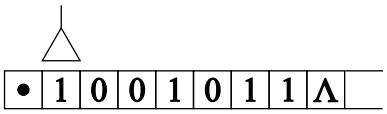
V tomto případě budeme volat funkci δ_z devětkrát, ale její výhodou je, že by mohla být užita i pro řetězec libovolné délky.

Vícekrokový Turingův stroj se hodí pro konstantní zrychlení, protože ubyly kroky, během kterých se hlava jen posouvá přes pole, která během výpočtu nemění a která neovlivní stav stroje. Nebude tedy potřeba definovat n (a více) příkazů (a stavů), místo kterých bude jen jeden příkaz. Tato definice je názornější a časově úspornější.

3.3 Turingův stroj s jednosměrně nekonečnou páskou

Mějme Turingův stroj jako v základní definici, s jediným rozdílem: páska bude pouze jednosměrně nekonečná. Bude začínat políčkem se symbolem $\bullet$, který nepůjde přepsat, ani z něj nebude hlava moci postoupit doleva. Přechodová funkce bude pro tento znak vždy definována takto:

$$\delta(q, \bullet) = (p, \bullet, R),\ \text{kde } q \in Q \setminus \{q_{prijme}, q_{zamitne}\},\ p \in Q.$$



Obrázek 3.1: **Stroj, jehož páska začíná symbolem •**

Vstup se běžně uvádí od prvního zapisovacího políčka doprava. Představu si můžeme vytvořit například podle obrázku 3.1.

Ekvivalence s vícekrokovým Turingovým strojem: ⇒ do pracovní abecedy přidáme znak • a v prvním kroku ho zapíšeme před vstup. Hlava pak postoupí na první políčko vstupu a dále bude přechodová funkce definována stejně jako u jednopáskového stroje.

⇐ Očíslujeme pole nového stroje:

$$j, i, 0, -1, 1, -2, 2, \ldots$$

kde j je políčko ve kterém se nachází symbol •. Do políčka s indexem i zapíšeme znak ∘. Tyto dva znaky stroj nebude přepisovat. Z políčka s indexem i bude hlava postupovat pouze podle instrukce R a z políčka s indexem j podle $3R$. Na ostatních políčkách se bude hlava pohybovat vždy o dvě políčka podle $2L, 2R$ ve smyslu L, R. Tedy na políčkách s kladným a nulovým indexem bude postupovat $2L$ ve smyslu L a $2R$ ve smyslu R stroje s oboustranně nekonečnou páskou. Na políčkách se záporným indexem pak opačně: $2L$ ve smyslu R a $2R$ ve smyslu L. Políčka i, j pak slouží pouze jako regulační políčka, která otáčí tyto smysly pohybu. Tímto jsme získali stroj, který pracuje stejně jako stroj s oboustranně nekonečnou páskou číslovaný:

$$\ldots, -2, -1, 0, 1, 2, \ldots$$

Vzhledem k políčkům i, j si stroj nemusí počítat index políčka, na kterém se právě nachází. Jeho přechodová funkce se pak kromě změn u L, R rozšíří pouze takto:

$$\delta(q, \circ) = (q, \circ, R)$$

$$\delta(q, \bullet) = (q, \bullet, 3R), \text{ kde } q \text{ je libovolný stav z } Q \setminus \{q_{prijme}, q_{zamitne}\}.$$

3.4 Vícepáskový Turingův stroj

Mějme Turingův stroj, jehož paměť se nachází na více páskách, jejichž počet je vždy předem pevně daný. Každá páska má svou vlastní hlavu. Všechny hlavy ovlivňují stav stroje, čímž mezi sebou komunikují. Pokud hlava na jedné pásce čte 0, ví o této

informaci i všechny hlavy na ostatních páskách a podle toho dále počítají. Tomu odpovídá definice přechodové funkce:

$$\delta : Q \times \Sigma^n \to Q \times \Sigma^n \times \{L, R\}^n,$$

kde n je počet pásek stroje.

Například výraz

$$\delta(q_i, a_1, \ldots, a_n) = (q_j, b_1, \ldots, b_n, L, R, \ldots, L)$$

znamená, že pokud je stroj ve stavu q_i a hlavy čtou na každé z pásek symboly v daném pořadí $a_1, \ldots, a_n$, pak má stroj přejít do stavu q_j, zapsat postupně na každou z pásek symbol $b_1, \ldots, b_n$ a pohnout každou z hlav podle zápisu (první doleva, druhou doprava, ..., poslední doleva).

Případně můžeme uvažovat i vícekrokový vícepáskový Turingův stroj, kde důkaz ekvivalence je analogický důkazu u vícekrokové a základní definice. Budeme tedy rovnou dokazovat pro vícekrokové stroje.

Ekvivalence s vícekrokovým Turingovým strojem: $\Leftarrow$ Jednopáskový stroj je speciální případ pro $n = 1$.

$\Rightarrow$ Podle [2]: Nechť má vícepáskový Turingův stroj právě n pásek, kde $n \in \mathbb{N}$. Políčka těchto pásek očíslujeme

$$\ldots, -2_i, -1_i, 0_i, 1_i, 2_i, \ldots, \ i = 1, \ldots, n, \text{ značí i-tou pásku.}$$

Pásku vícekrokového stroje očíslujeme:

$$\ldots, -2, -1, 0, 1, 2, \ldots.$$

Tuto pásku rozdělíme modulo n, tedy například $k, k \leq n$, vezmeme z naší pásky políčka s indexy:

$$\ldots, -2n + k, -n + k, k, n + k, 2n + k, \ldots$$

do kterých vypíšeme postupně pásku k vícepáskového stroje. Hlava pak postupuje po k-té pásce tak, že postupuje vždy o n polí ve smyslu nL, nR, tedy po indexech k modulo n. Informaci o pozicích hlav na jednotlivých páskách potom bude potřeba zajistit pomocí přechodové funkce.

$\Rightarrow$ Podle [5]: Další možností je zavést si do pracovní abecedy symboly, které budou značit vždy začátek ♡ a konec ♣ pásek a s jejich pomocí můžeme rozdělit pásku základního stroje. Příklad pro dvě pásky je uveden v tabulce 3.1. Pomocné políčko mezi ♣ a ♡ slouží jako paměť, kam si stroj bude zapisovat poznámky z právě opouštěné pásky.

♡	1. páska	♣	pomocné políčko	♡	2. páska	♣

Tabulka 3.1: **Rozdělení pásky základního stroje**

Pokud hlava přejde na další pásku, zapíše na pomocné políčko znak, který povede stroj k tomu, aby se vrátil na správné políčko pásky. Tato informativní políčka musí být ohraničena z obou stran, aby nedošlo k přepisu dat na pásce. V případě úplného využití prostoru mezi ♡ a ♣. Pokud by došlo k tomu, že by se kapacita vyhrazeného místa pro libovolnou z pásek úplně zaplnila, uložila by hlava dočasnou informaci do pomocného políčka a posunula by zbytek celé pásky základního stroje doprava (resp. doleva), aby vzniklo další místo a platilo, že každá z pásek má nekonečný prostor k zapisování.

Příklad: Mějme Turingův stroj, který provádí operaci sčítání dvou čísel v binárním zápisu.

Můžeme vzít třípáskový Turingův stroj. Na první dvě pásky zapíšeme čísla v binárním tvaru, která chceme sečíst. Na třetí pásku bude stroj během výpočtu postupně zapisovat jednotlivé cifry výsledku.

Na 1. a 2. pásce nejprve obě hlavy dojdou na poslední znak obou čísel (např. najdou první prázdný znak Λ) a vrátí se o pole doleva. Pak budou postupovat po jednom políčku doleva až dojdou na konec jednoho z nich (nebo i obou čísel najednou). Před kratší číslo připíše stroj znaky 0, dokud nedojde na konec delšího čísla. Následně ještě připíše před obě čísla 0. Pak hlavy opět dojdou na konec obou čísel, třetí hlava na své pásce také postoupí o stejný počet kroků doprava a stroj přejde do stavu q_1.

Dále postupujeme školským algoritmem: 1. a 2. hlava přečtou číslice a 3. hlava zapíše druhou číslici součtu a první si bude pamatovat. Ve stavu q_1 se bude 3. hlava chovat takto: pro $0+0$ a $1+1$ zapíše 0, pro $0+1$ a $1+0$ zapíše 1 a posune se o pole doleva. Pokud navíc nastane případ $1+1$, přejde stroj do stavu q_2, jinak zůstane v q_1. Stav q_2 bude značit stav, kdy přičítáme ještě jednu jedničku z předešlého výpočtu. V tomto stavu se bude stroj chovat takto: pro $0+0$ a $1+1$ zapíše 1, pro $0+1$ a $1+0$ zapíše 0 a posune se o pole doleva. V případě $0+0$ přejde stroj zpět do stavu q_1, jinak zůstane v q_2.

Pokud obě hlavy na prvních dvou páskách dojdou na prázdný znak, stroj se zastaví a podá výstup z 3. pásky.

Na jednopáskovém stroji bychom museli zapsat čísla za sebe například tak, že bychom předpokládali maximální délku čísel k, mezi nimi by vždy byl uveden pracovní znak α, který by je odděloval, a hlava by pak postupovala o k kroků mezi jednotlivými částmi pásky a za posledním α by počítala součet.

Vícepáskový Turingův stroj je jednou z jeho nejdůležitějších verzí a vycházejí z ní i složitější modely reálných strojů. Dovoluje uchovávat několik různých údajů najednou s rychlým přístupem k nim. Je také názornější a přehlednější pro vnějšího pozorovatele, ale i úspornější na časovou složitost.

3.5 Turingův stroj s více hlavami

Mějme Turingův stroj podle základní definice, který má namísto jedné hlavy více hlav na stejné pásce. Tento stroj je svým způsobem analogický vícepáskovému Turingovu stroji. Má stejnou přechodovou funkci, ale pouze jednu pásku. Každá z hlav má informace jen o políčku, na které právě ukazuje, a o stavu, ve kterém se nachází stroj. Může ale nastat, že se dvě a více hlav dostanou na stejné políčko. Je potřeba určit prioritu jednotlivých hlav, neboli, že jedna hlava pracuje jako první, skončí a pak teprve začne pracovat další. Je také možné u některých hlav zvolit, že mohou jen číst a u jiných, že mohou jen zapisovat. Například můžeme mít stroj se třemi hlavami, kde první hlava může pouze číst, druhá může číst i zapisovat a třetí pouze zapisovat.

Ekvivalence se základním Turingovým strojem: $\Leftarrow$ Jedná se o stejný případ s počtem hlav rovným jedné.

$\Rightarrow$ Definujeme přechodovou funkci tak, aby jediná hlava základního stroje vykonala postupně to, co 1. hlava, pak přešla na místo 2. a zapsala dle instrukcí to samé, co 2. hlava, pak se přesunula na místo další hlavy atd. Pak se vrátila opět na místo 1. hlavy. To budeme stále opakovat.

3.6 Turingův stroj s více abecedami

Pro vícepáskové stroje je navíc možno definovat více abeced. Například vstupní abecedu Θ_1 pro první pásku, výstupní Θ_2 pro druhou, nebo případně pro každou pásku jinou abecedu.

Ekvivalence s vícepáskovým Turingovým strojem: $\Leftarrow$ Budeme počítat pouze s vstupní a pracovní abecedou. Pak se bude jednat o vícepáskový Turingův stroj definovaný výše.

$\Rightarrow$ Naopak všechny abecedy, které nová definice užívá, sloučíme v pracovní abecedu Σ. Vstupní a výstupní, pak sloučíme také v jednu abecedu, která bude odpovídat abecedě Θ dle základní definice.

Více abeced se opět hodí pro přehlednost. Případně zajímá-li nás více různých informací ohledně výstupu a podobně. Jedná se však hlavně o formální definici, stroj jinak pracuje úplně stejně jako vícepáskový Turingův stroj.

Kapitola 4

Méně uvažované modifikace Turingových strojů

V této kapitole se budeme věnovat modifikacím Turingova stroje, které jsou méně obvyklé. Simulace některým z předchozích strojů mohou být méně názorné a časově náročnější než předchozí simulace. Při dokazování ekvivalence budeme postupovat jako v předchozí kapitole.

4.1 2D Turingův stroj

Prostorový, neboli 2D Turingův stroj má paměť myšlenou jako nekonečnou plochu, rozdělenou na jednotlivá políčka [3]. Ta mají sousedy ze čtyř různých stran. Můžeme si ji představit jako nekonečný čtverečkový papír, po němž hlava postupuje vždy po sousedních čtverečcích. Postup je tedy možný jedním ze směrů: doleva L, doprava R, nahoru U a dolů D.

Přechodová funkce 2D stroje je tvaru:

$$\delta : Q \times \Sigma \to Q \times \Sigma \times \{L, R, U, D\} \ .$$

Pro přehlednost zavedeme systém souřadnic (x, y) analogicky jako u $\mathbb{Z}^2$ bez os. Souřadnice x značí horizontální posun odpovídající L a R a y vertikální odpovídající U a D, kde $x, y \in \mathbb{Z} \setminus \{0\}$.

Paměť si rozdělíme na čtyři kvadranty. První kvadrant bude mít obě složky kladné, druhý bude mít první složku x zápornou a druhou složku y kladnou. Třetí kvadrant bude mít obě složky záporné a čtvrtý první složku x kladnou a druhou složku y zápornou.

Pro názornost si uveďme tabulku 4.1, kde ukazujeme souřadnice několika políček kolem počátečního políčka, které budeme dále vždy značit $(1, 1)$.

	$(-2,2)$	$(-1,2)$	$(1,2)$	$(2,2)$	
	$(-2,1)$	$(-1,1)$	$(1,1)$	$(2,1)$	
	$(-2,-1)$	$(-1,-1)$	$(1,-1)$	$(2,-1)$	
	$(-2,-2)$	$(-1,-2)$	$(1,-2)$	$(2,-2)$	

Tabulka 4.1: **Souřadnice paměti 2D Turingova stroje**

Ekvivalence s Turingovým strojem s jednosměrně nekonečnou páskou:
$\Leftarrow$ Budeme počítat pouze s možnostmi posunu hlavy $\{L, R\}$ po políčkách $(n, 1)$, kde $n \in \mathbb{N}^+$. V políčku $(-1, 1)$ bude znak konce pásky, který nebude možno přepsat a ze kterého nebude hlava nikdy postupovat doleva.

S čtyřpáskovým Turingovým strojem: $\Rightarrow$ Zpočátku budeme uvažovat pouze políčka (x, y) z prvního kvadrantu, kde $x, y \in \mathbb{N}^+$.

Nejprve definujeme funkci $f_1 : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, která očísluje všechna políčka prvního kvadrantu paměti 2D stroje takto:

$$f_1(x, y) = \frac{(x + y - 2)(x + y - 1)}{2} + x$$

Neboli číslujeme po diagonálách vedoucích doleva dolů:

- $f_1(1, 1) = 1$
- $f_1(1, 2) = 2,\ f_1(2, 1) = 3$
- $f_1(1, 3) = 4,\ f_1(2, 2) = 5,\ f_1(3, 1) = 6$
- ...

Tím získáme očíslování každého políčka prvního kvadrantu, jak je ukázáno v tabulce 4.2.

Pro celou plochu 2D stroje, kde $x, y \in \mathbb{Z} \setminus \{0\}$ bude očíslování vypadat takto: Vezmeme absolutní hodnotu souřadnic x a y, spočítáme $4 \cdot f_1(|x|, |y|) + q$, kde $q = 0, 1, 2, 3$ podle kvadrantu ve kterém se nacházíme. Bude tedy platit:

1. kvadrant bude číslován $4 \cdot f_1(|x|, |y|) + 0$
2. kvadrant bude číslován $4 \cdot f_1(|x|, |y|) + 1$
3. kvadrant bude číslován $4 \cdot f_1(|x|, |y|) + 2$

		4	8	13	
		2	5	9	
	$(-1,1)$	1	3	6	
	$(-1,-1)$	$(1,-1)$			

Tabulka 4.2: **Očíslování prvního kvadrantu funkcí** f_1

4. kvadrant bude číslován $4 \cdot f_1(|x|, |y|) + 3$

Z toho můžeme určit funkci f, která nám očísluje celou plochu paměti 2D stroje:

$$f(x,y) = 4 \cdot f_1(|x|, |y|) + q = 2 * (|x| + |y| - 2)(|x| + |y| - 1) + |x| + q$$

Je tedy zřejmé, že z indexu políčka snadno zjistíme v jakém jsme kvadrantu, pokud si spočítáme $f(x,y)$ mod 4.

Čtyřpáskový Turingův stroj bude používat dvě pásky pouze k tomu, aby určil polohu hlavy 2D stroje. Jedné z nich tedy budeme pracovně říkat x a druhé y. Na pásce x budeme mít první souřadnici a na pásce y druhou. Zpočátku bude na obou zapsána 1. Tedy posuny 2D stroje: L a R budou odpovídat -1 a $+1$ na pásce x, U a D budou odpovídat $+1$ a -1 na pásce y.

Z těchto dvou pásek je pak snadné si pomocí funkce f počítat na třetí pásce, o kolik políček se bude muset posunout hlava na čtvrté pásce.

Na čtvrtou pásku uvedeme vstup zapsaný postupně podle očíslovaných políček funkcí f. Na tuto pásku tedy převedeme všechny výpočty 2D stroje. Pokud se hlava 2D stroje posune o políčko, hlava na této pásce se posune o tolik políček, kolik určí funkce f, kterou si stroj počítá na třetí pásce.

Z počátku budou hlavy na páskách x a y ukazovat na 1, na třetí pásce bude zapsána 1 a na čtvrté pásce bude hlava ukazovat na počátek vstupu, který se v 2D stroji nacházel na políčku $(1,1)$.

Tím jsme získali čtyřpáskový Turingův stroj, který simuluje 2D Turingův stroj.

Lze rovněž uvažovat 2D Turingův stroj s více hlavami nebo více paměťmi, analogicky jako vícepáskový stroj (případně kombinace pásek a ploch coby paměti).

4.2 Turingův stroj – Strom

Verze Turingova stroje, kterou budeme dále nazývat Strom, má paměť myšlenou jako binární strom. Jedná se o nekonečnou pásku s počátkem, kde všechna políčka,

kromě počátečního, sousedí se třemi políčky. Přechodová funkce je tvaru:

$$\delta : Q \times \Sigma \to Q \times \Sigma \times \{O, LS, PS\},$$

kde O značí posun hlavy na předchůdce (nebo také otce) políčka, na které hlava ukazuje. LS značí posun hlavy na levého syna (následníka) a PS na pravého syna.

V počátečním políčku bude zapsán znak $\bullet$, který nebude možno přepsat. Z tohoto políčka se dá postoupit pouze na levého syna posunem hlavy podle LS. V prvním políčku bude zapsán nepřepsatelný znak z toho důvodu, abychom ošetřili případy, kdy by stroj chtěl postoupit z kořene do jeho předchůdce. Jako kořen bude sloužit levý syn políčka se znakem $\bullet$. Dále z každého políčka budeme moci postupovat vždy na otce a na oba následníky, podle toho, jak bude definovaná přechodová funkce.

Získáme tím pásku s počátkem ve tvaru binárního stromu s jedním políčkem nad kořenem, která se v každém vrcholu větví do dvou nekonečných binárních stromů. Políčko nad kořenem se znakem $\bullet$ zavádíme proto, abychom snadno získali model jednosměrného Turingova stroje. Možnou představu si můžeme vytvořit pomocí obrázku 4.1.

Vstup budeme zapisovat vždy od kořene do levé větve stromu. Šlo by také říci, že při zápisu vstupu by hlava postupovala pořád takto: zapiš znak a přejdi do levého syna LS.

Ekvivalence s Turingovým strojem s jednosměrně nekonečnou páskou: $\Leftarrow$ Máme jednopáskový stroj s přechodovou funkcí:

$$\delta : Q \times \Sigma \to Q \times \Sigma \times \{L, R\} \ .$$

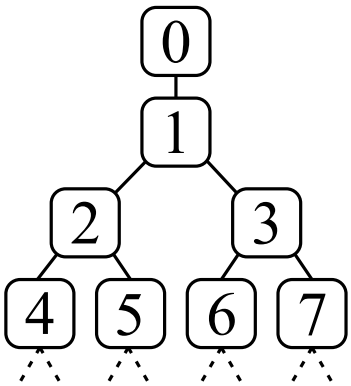
Přechodovou funkci Turingova stroje – Stromu definujeme pouze na $\{O, LS\}$, čímž získáme přechodovou funkci:

$$\delta : Q \times \Sigma \to Q \times \Sigma \times \{O, LS\}$$

která bude až na značení pohybu hlavy ekvivalentní přechodové funkci jednopáskového stroje. Neboli postupujeme pouze po jedné větvi stromu, tedy z každého políčka pouze do dvou směrů.

S dvoupáskovým Turingovým strojem: $\Rightarrow$ Nyní chceme ukázat, že výpočet stroje Stromu, můžeme počítat i pomocí dvoupáskového stroje. Pásku Stromu očíslujeme postupně po úrovních stromu (viz obrázek 4.1), počáteční políčko se znakem $\bullet$ bude mít číslo 0, kořen bude mít číslo 1, jeho následníci 2 a 3, jejich následníci $4, 5, 6, 7$ atd.

Je tedy zřejmé, že každý levý následník má po očíslování index rovný dvojnásobku svého předchůdce a pravý následník má index rovný dvojnásobku plus jedna. Naopak



Obrázek 4.1: **Očíslování binárního stromu po úrovních**

každý předchůdce má index rovný dolní celé části poloviny svého syna. Pokud například jsme na políčku očíslovaném $k \in \mathbb{N}^+$. Pak jeho levý syn má číslo $2k$ a pravý $2k+1$ a jeho předek $\lfloor \frac{k}{2} \rfloor$.

Dvoupáskový stroj dostane na první pásce vstup, který bude seřazen po úrovních stromu, tedy $1, 2, 4, 8, \ldots$ podle očíslování Stromu. Druhá páska pak bude sloužit k výpočtům, o kolik se má posunout hlava na první pásce podle výše uvedených pravidel. Z počátku bude na této pásce zapsána 1 a hlava na první pásce bude ukazovat na znak zapsaný v kořeni stromu.

Turingův stroj s větvenou páskou by se mohl užívat například pro snazší zápis stromů ve smyslu grafů (tedy graf o jedné komponentě, který nemá cykly). Jak však zapsat libovolný strom jako binární strom?

Poznámka: Každý strom lze zapsat do binárního stromu.

Tato poznámka obecně [3] známá a uvádíme ji pro úplnost.

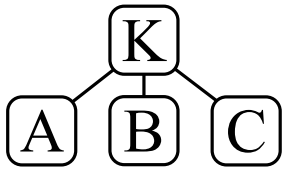
Převod vypadá takto: kořen přepíšeme do kořene. Pro přehlednost budeme mluvit o otcích a synech v binárním stromu a předchůdcích a následnících ve stromu.

Nechť máme libovolný vrchol, který má k následníků, kde $k \in \mathbb{N}$.

- Jeho první následník bude v binárním stromu zapsán na místě jeho levého syna.
- Jeho druhý následník bude v binárním stromu zapsán na místě pravého syna jeho levého syna, tedy jako pravý syn jeho prvního následníka
- Jeho třetí následník bude v binárním stromu zapsán na místě pravého syna druhého následníka.
- ...

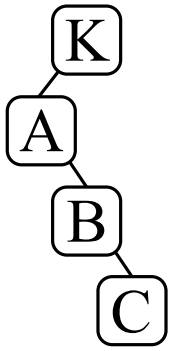
- Jeho k-tý syn bude v binárním stromě zapsán na místě pravého syna jeho $k-1$-ního následníka.

Ukažme si to na příkladně pro $k = 3$:



Obrázek 4.2: **Strom tvaru: kořen s třemi syny**

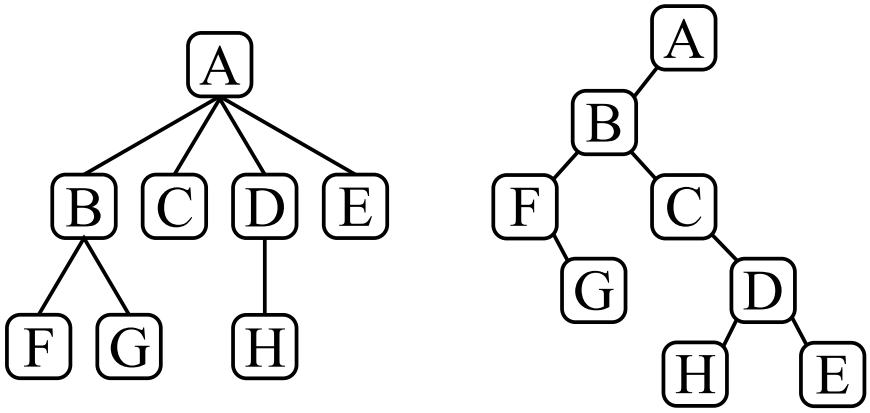
Máme strom, kde kořen K má tři následníky v pořadí A, B, C, viz obrázek 4.2. Při převodu na binárního strom bude kořen K mít levého syna A a nebude mít pravého syna. Vrchol A pak bude mít pouze pravého syna B a ten bude mít zase pouze pravého syna C. Tento binární strom je uveden na obrázku 4.3.



Obrázek 4.3: **Binární strom vzniklý ze stromu z obrázku 4.2**

Tedy převedeme-li libovolný strom na binární strom, bude mít každý vrchol své syny zapsané v pravé větvi levého syna. Své bratry bude mít zapsané ve své pravé větvi. Kořen nebude mít žádného pravého syna, protože kořen nemá žádné prvky na stejné úrovni. U každého vrcholu bude záležet na tom, zda se jedná o levého syna, potom jeho otec je jeho předchůdcem, nebo o pravého syna, potom jeho otec je v původním stromu na stejné úrovni.

Příklad: Na obrázku 4.4 vidíme převod stromu s libovolným počtem následníků do binárního stromu. Písmena v obou stromech značí vrcholy, které mají stejný význam určený stromem vlevo.



Obrázek 4.4: **Strom nalevo převedený do binárního stromu napravo**

Tímto způsobem jde libovolný strom zapsat do binárního stromu a tedy i námi definovaný Turingův stroj by mohl pracovat s libovolným stromem.

Kapitola 5

Nedeterministický Turingův stroj

Nedeterministický Turingův stroj budeme definovat pomocí základního Turingova stroje s tím rozdílem, že pro libovolnou dvojici $Q \times \Sigma$ může existovat více než jeden přechod do $Q \times \Sigma \times \{L, R\}$. Přechodová funkce bude tvaru

$$\delta : Q \times \Sigma \to \mathcal{P}(Q \times \Sigma \times \{L, R\}),$$

kde $\mathcal{P}$ značí potenční množinu. Přechodová funkce nyní nevede nejvýše do jediného možného prvku množiny $Q \times \Sigma \times \{L, R\}$, ale přímo do celé její podmnožiny a stroj může používat jakoukoliv z těchto instrukcí. Máme tedy více možností, kudy se může výpočet stroje ubírat. Vede-li některá z možností postupu stroje k přijímajícímu stavu, řekneme, že nedeterministický Turingův stroj přijímá na daném vstupu.

Nedeterministický Turingův stroj je rozšířením deterministického Turingova stroje, se kterým jsem se setkali v předchozích kapitolách. Zatímco deterministický stroj má předem přesně dáno, jak bude počítat, nedeterministický stroj může počítat více způsoby. Deterministický stroj ve stavu q po přečtení znaku a bude vždy postupovat stejně. Nedeterministický stroj přečte znak a ve stavu q a potom se rozhodne, jak dále počítat podle možností z jeho přechodové funkce (jít směrem L, či R, přejít do stavu q, nebo do jiného stavu).

Podrobněji o nedeterminismu pojednává například [5].

Všechny možné výpočty nedeterministického stroje můžeme zapsat do stromu. Každá hrana bude značit jednu možnou volbu podle přechodové funkce. Jedna větev tedy bude značit jeden možný způsob, jakým by stroj mohl pracovat. Důležité je, existuje-li jedna konečná větev, která vede k přijímajícímu stavu.

Vezměme si například graf $G = (V, E)$, kde V je množina vrcholů a E množina hran grafu G. Ten může být reprezentován například na 2D stroji svou incidenční maticí. Zvolíme si dva vrcholy: $v_1, v_2 \in V$. Budeme zjišťovat, jestli mezi nimi existuje cesta. Začneme ve vrcholu v_1 a stavu stroje q_0. Pokud z v_1 vede více hran, je

více možností, kudy by mohl Turingův stroj postupovat. Nedeterministický stroj se tedy rozhodne pro jednu z hran a přejde na vrchol na druhém konci. Z něj pak postupuje stejně, jako ve vrcholu v_1, nejedná-li se přímo o vrchol v_2, kde jeho výpočet končí úspěchem a stroj přejde do stavu q_{prijme}. Existuje-li cesta mezi v_1 a v_2, pak je možnost, že si stroj pokaždé vybere tu správnou hranu, kterou se dále vydat, a dojde až do vrcholu v_2. Pak říkáme, že nedeterministický stroj přijímá na daném vstupu. Tento výpočet by se dal zapsat do stromu, kde v kořeni by byl vrchol v_1. Jeho synové by pak byly vrcholy, se kterými je v_1 spojen hranou. Pokud se v grafu nachází cyklus, pak takový strom může mít i nekonečné větve, ale pokud existuje cesta mezi v_1 a v_2, bude mít větev konečné délky, která povede k přijímajícímu stavu. Pak by výpočet stroje skončil úspěchem.

Věta: Ke každému nedeterministickému Turingovu stroji existuje ekvivalentní deterministický Turingův stroj takový, že oba přijímají stejný jazyk.

Deterministický Turingův stroj můžeme také zařadit mezi nedeterministické stroje. $\mathcal{P}$ obsahuje vždy maximálně jedinou možnou cestu dál. Otákou je, jak napsat přechodovou funkci deterministického Turingova stroje tak, aby pracoval stejně jako nedeterministický? Tedy aby se zastavil, nebo nezastavil vždy stejně jako nedeterministický stroj.

Ideou důkazu je, že deterministický stroj bude postupně procházet všechny větve možného postupu nedeterministického stroje po úrovních, podobně jako u Turingova stroje – Stromu. Například je možné použít třípáskového deterministického stroje, který na první pásce dostane vstup, ten zkopíruje na druhou pásku a pomocí třetí pásky zkouší možnosti, kudy by se mohl ubírat výpočet nederministického stroje. Našel-li by deterministický stroj přijímající stav, přijal by daný vstup. V opačném případě by se nezastavil.

Podrobný důkaz je možné nalézt v [5].

Narozdíl od deterministického stroje, který má předem pevně stanoveno, jak bude na základě čteného symbolu postupovat, má nedeterministický stroj více možností jak postupovat. Můžeme tedy říci, že deterministické Turingovy stroje jsou součástí nedeterministických strojů(viz [4]).

Naše simulace ukazuje, že náročnost výpočtu roste s tím, na jak nízké úrovni je přijímací stav. Tato náročnost je exponenciálně větší. Je stále otevřenou otázkou, zdali tato náročnost nejde zmenšit.

Kapitola 6

Závěr

Dle Church-Turingovy teze nezáleží na tom, kterou z variant Turingova stroje zvolíme pro naše výpočty. Jak jsme v této práci ukázali, každý z námi definovaných Turingových strojů, včetně Stromu, tuto tezi potvrzuje.

Turingovy stroje slouží k formálnímu modelování počítače a jejich varianty se hodí k různým využitím. Běžně rozlišujeme pouze to, zda je Turingův stroj nedeterministický, nebo není. Všechny definice z kapitol 2 – 4 jsou obecně nazývány deterministické. Takto definované stroje mají na daném vstupu vždy jednoznačně určeno, jak a co budou dále počítat.

Někdy nám záleží i na časové a prostorové složitosti výpočtu, které měříme počtem pohybů hlavy a počtem políček pásky hlavou navštívených, a vyjadřujeme je jako funkci délky vstupu. Dva stroje mohou počítat tutéž funkci, ale složitost jejich výpočtu se může výrazně lišit. Tyto otázky jsou předmětem tzv. teorie vypočetní složitosti, viz. například [5].

Užíváme-li pojem Turingova stroje, můžeme mít na mysli libovolnou definici z kapitol 2 – 4, není-li předem pevně stanoven jeden určitý typ definice.

Literatura

[1] Arora S., Barak B. *Computational Complexity: A Modern Approach*, Cambridge University Press, 2009

[2] Černý M.: *Výpočty* Technická zpráva, Vysoká škola ekonomická Praha, 2009.

[3] Demuth O., Kryl R., Kučera A.: *Teorie algoritmů I.* SPN, Praha 1984

[4] Holub Š.: *Složitost pro kryptografii*, Učební text, MFF UK, 2004

[5] Sipser M.: *Introduction to the theory of Computation*, PWS Publishing company, 1997.

[6] Turing A.: *On Computable Numbers, with an Application to the Entscheidungsproblem* Proc. London Math. Soc. 42 (1936), pp. 230–265.