



Středoškolská technika 2022

Setkání a prezentace prací středoškolských studentů na ČVUT

Alan Turing a jeho přínos pro informatiku

Barbora Šubová

První soukromé jazykové gymnázium
Brandlova 875, Hradec Králové

Prohlášení

Prohlašuji, že jsem svou práci vypracovala samostatně a použila jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů..

Nemám závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Hradci Králové dne 17. března 2022

.....

Barbora Šubová

Anotace

Práce představuje Alana Turinga jako jednu z nejdůležitějších osobností dvacátého století, jehož mozek pomohl rozhodnout o konci druhé světové války. Práce stručně shrnuje nejdůležitější životopisná data a zaměřuje se na jeden z jeho hlavních přínosů pro teoretickou informatiku - Turingovy stroje. Pro zpracování tématu jsou shrnuty a vysvětleny související pojmy z teorie jazyků a gramatik i složitosti a vyčíslitelnosti.

Cílem práce je sestavení vhodného materiálu pro pochopení funkce a významu Turingových strojů a simulace výpočtu.

Klíčová slova:

Turingův stroj, Turing, jazyky, gramatiky, složitost, vyčíslitelnost, problém zastavení, algoritmus, teoretická informatika

Annotation

The thesis presents Alan Turing as one of the most important pioneers of computer science of the twentieth century, who helped to decide the end of World War II. The thesis briefly summarizes his most important biographical dates and focuses on one of his main contributions to theoretical computer science - Turing machines. Related concepts from the theory of languages and grammars as well as complexity and computability are summarized and explained for the topic elaboration.

The aim of this thesis is to compile a suitable material for understanding the function and meaning of Turing machines and to simulate computation.

Keywords:

Turing machine, Turing, languages, grammars, complexity, computability, halting problem, algorithm, theoretical computer science

Poděkování

V první řadě děkuji panu doc. Ing. Filipu Malému, Ph.D., který mou práci vedl a provázel mě velmi cennými radami. Dále velké díky patří mému tatínkovi, díky kterému jsem většinu věcí pochopila, mé mamince, která strávila spoustu času nad pročitáním celé práce a opravováním pravopisných chyb. Velmi děkuji Mgr. Jitce Čákové a spolužákům, kteří mi pomohli poznámkami k anglické verzi anotace. Velké díky patří i mnoha dalším, jež mi během uplynulého pololetí byli mentální podporou.

Obsah

Úvod a cíl práce.....	10
Teoretická část.....	11
1 Život Alana Turinga.....	11
1.1 Dětství	11
1.2 Školní léta.....	11
1.3 Studium na Cambridgeské univerzitě.....	12
1.4 Studium na Princetonské univerzitě	12
1.5 Druhá světová válka	12
1.6 Poválečný život	13
1.7 Konec života.....	13
2 Definice pojmů objevujících se v práci.....	15
2.1 Problém	15
2.2 Vyčísitelnost	15
2.3 Spočetnost.....	16
2.3.1 Spočetná množina	16
2.3.2 Nespočetná množina	16
2.4 Algoritmus	17
2.5 Složitost	17
2.5.1 Třídy složitosti	18
2.5.2 Asymptotická složitost	19
2.6 Automat	19
2.7 Syntaxe	19
2.8 Sémantika	20
2.9 Abeceda.....	20
3 Chomského hierarchie.....	21

3.1	Třídy gramatik.....	22
3.1.1	Gramatiky typu 0.....	22
3.1.2	Gramatiky typu 1.....	23
3.1.3	Gramatiky typu 2.....	23
3.1.4	Gramatiky typu 3.....	23
3.2	Srovnávací tabulka jazyků, gramatik a strojů dle Chomského.....	24
4	Turingův stroj.....	25
4.1	Skladba Turingova stroje.....	25
4.2	Základní definice Turingova stroje	26
4.3	Church - Turingova teze	29
4.4	Konfigurace Turingova stroje.....	29
4.5	Vícepáskové Turingovy stroje.....	30
4.6	Problém zastavení.....	31
4.7	Použití v praxi	31
4.8	Speciální případ Turingova stroje	32
4.9	Alternativy Turingova stroje	32
4.9.1	RAM stroj	32
4.9.2	Lambda kalkul.....	33
4.9.3	Postův stroj.....	33
	Praktická část.....	34
5	Metodika	34
6	Řešení palindromů na jednopáskovém Turingově stroji.....	35
6.1	Sudý palindrom	36
6.1.1	Sudý palindrom končící v koncovém stavu	36
6.2	Sudý i lichý palindrom	56
6.3	Nepřijetí řetězce.....	57
6.4	Zacyklení	58

Diskuse.....	59
Závěr	61
Prameny a literatura.....	62
Zdroje obrázků.....	64
Seznam obrázků	65
Seznam tabulek	65

Úvod a cíl práce

Prvotní inspirací pro výběr tématu této práce mi byl film o Alanu Turingovi *Kód enigmy* (v originále *The Imitation Game*). Tato historická osobnost mě zaujala a chtěla jsem si o ní zjistit více informací. Tehdy mi došlo, čím vším Alan Turing přispěl tomuto světu a věděla jsem také, že porozumění jeho Turingovým strojům je jeden z nejzákladnějších kamenů teoretické informatiky, které bych se možná v budoucnu chtěla více věnovat.

Cíle mé práce jsou dva. První cíl je můj osobní – nastudovat a pochopit úvod do teoretické informatiky a připravit se tak pro další studium v oboru. Druhým cílem je usnadnit tuto cestu případnému čtenáři sumarizováním poznatků.

V teoretické části práce bude nejprve stručně představen život Alana Turinga, dále se podrobněji podíváme na jednotlivé přínosy a zvláště se zaměříme na Turingův stroj, teoretický model výpočetního stroje, na jehož podstatě stojí všechny dnešní počítače.

V praktické části si ukážeme příklad výpočtu na Turingově stroji, podrobně si popíšeme každou iteraci a odečteme výsledek.

Teoretická část

První část této práce se bude zabývat výhradně teoretickým vzhledem do problematiky, bez kterého nebylo možno zpracovat praktickou část, která je hlavním přínosem této práce.

1 Život Alana Turinga

Alan Turing byl britský matematik a průkopník kryptoanalýzy. Zajisté ho můžeme řadit k nejvýznamnějším mužům minulého století, jelikož svou genialitou přispěl do mnoha matematických oborů. Postavil jedny z nejzákladnějších kamenů dnešního digitálního světa a je považován za otce moderní počítačové vědy. Dodnes se vedou spekulace, o kolik let by se protáhla či jak by vlastně dopadla druhá světová válka, nebýt Alana Turinga a jeho týmu při dešifrování německého kódu Enigmy [1;2;3].

1.1 Dětství

Alan Turing se narodil 23. června 1912, jako druhý syn Ethel a Julia Turingových, kteří dlouhodobě pobývali v Madrasu v jižní Indii, kde Alanův otec působil jako úředník. Oba rodiče si však velmi přáli, aby se narodil na území Anglie, a tak se na pár měsíců vrátili do jejich rodné země. Chvilí po Alanově narození Julius odcestoval a Ethel ho následovala o patnáct měsíců později. Alan tak zůstal sám s bratrem v Anglii, kde byl vychováván chůvami a rodinnými přáteli [1;3].

1.2 Školní léta

Alan byl průměrným studentem, vyjma přírodních věd a matematiky. V těchto předmětech vynikal natolik, že byl přijat na Scherbone School v Dorsetu¹, chlapeckou soukromou školu, kde se seznámil s Christopherem Morcomem, stejným nadšencem pro přírodní vědy, ke kterému si vybuodoval vztah silnější než jen přátelství. Christopher

¹ Zde se dostáváme k první novinové zmínce o Alanu Turingovi, ovšem nejednalo se o žádný významný objev, ale o jeho pozoruhodný sportovní výkon, jelikož přes vyhlášenou generální stávkou, která znemožnila Turingovi dostat se do školy vlakem, ujel 100 km na kole za pouhé dva dny, jelikož nechtěl zmeškat první den školy.

bohužel 13. února 1930 zemřel na tuberkulózu, což byla pro Alana obrovská tragédie. Zvýšil však tak své úsilí o stipendium na Cambridge, jelikož to byl jejich společný sen a chtěl tak uctít jeho památku [1;2;3].

1.3 Studium na Cambridgeské univerzitě

V roce 1931 byl na první pokus přijat na King's College v Cambridgi, kde byl na základě své disertační práce v roce 1935 zvolen členem univerzitní koleje (*fellow*). Na univerzitě byl více než spokojen. Mnoho studentů zde debatovalo o řešitelnosti určitých problémů a mladý Alan Turing byl samozřejmě mezi nimi. V roce 1936 vydává článek: „On Computable Numbers, with an Application to the Entscheidungsproblem“, kde poprvé představuje koncept Turingova stroje². Ze stejného roku (1936) pochází Lambda Kalkul, teoretický model výpočetního stroje Alonza Churcha, o kterém se zmiňuje ve svém článku o nerozhodnutelnosti problému. Oba matematici (Alan Turing a Alonzo Church) totiž hledali způsob, jakým by se dal sestavit univerzální model, který by mohl popsat chování libovolného algoritmu [3;4;5;6;7].

1.4 Studium na Princetonské univerzitě

V roce 1937 Alan Turing přeplul Atlantik a usídlil se na univerzitě v New Jersey, kde později získal doktorát. V Princetonu studoval pod vedením již zmíněného Alonza Churcha. Společně dokázali, že Churchův Lambda kalkul a Turingův stroj mají stejnou výpočetní sílu. Zde se pomalu, ale jistě blížili k sestavení tzv. Church – Turingovy teze, na kterou se podrobněji podíváme ve stejnojmenné kapitole [3;5;7;8].

1.5 Druhá světová válka

Alan Turing ve druhé světové válce nepomohl úžasnou schopností bojovat, fyzickou silou nebo položením svého života za krále. Pomohl ovšem, jak nejlépe mohl, svou úžasnou inteligencí. Stal se členem týmu v Bletchely parku, kde spolupracoval na dešifrování kódu německého šifrovacího stroje Enigma [3;7;9].

² Název „Turing stroj“ se vžil později. Turing svůj stroj nazýval „Univerzální stroj“.

1.6 Poválečný život

Po druhé světové válce (1948) se na základě Turingova stroje začaly stavět první počítače řízené programem uloženým ve vnitřní paměti.

Roku 1948 se Turing navrátil k univerzitnímu životu, tentokrát však už jako profesor v Manchesteru. Jeho zájem si vybral cestu umělé inteligence. Dlouhá léta strávil hloubáním nad možností inteligence strojů. V roce 1950 publikoval ve filozofickém periodiku svou esej „Computing Machinery and Intelligence“, v níž popisoval své objevy a myšlenky. Byl to právě Alan Turing, kdo si jako první položil otázku: *Umí počítače myslet?* Na základě této myšlenky hovoříme o tzv. Turingově testu, který Turing sám nazval Imitační hra. Byl to velký krok do studia teoretické informatiky, jelikož umělá inteligence jakožto vědní disciplína vychází z jeho práce [3;10;11].

1.7 Konec života

V roce 1952 byl Alan Turing obviněn ze sexuálního deliktu, na základě jeho přiznání o sexuálním vztahu s Arnoldem Murrayem při vyšetřování vykradení jeho bytu, ze kterého byl podezírán právě Murray. Kvůli obvinění mu byl odepřen přístup k utajovaným informacím a účast ve Vládním komunikačním centru (GCHQ). Povedlo se mu však udržet si místo na univerzitě v Manchesteru [3].

Po odsouzení si mohl vybrat, zda stráví dva roky ve vězení nebo probací, která byla spojena s podstoupení hormonální „léčby“. Aby mohl dál pokračovat ve své práci, rozhodl se pro druhou možnost a po dobu jednoho roku mu byly podávány dávky syntetického estrogenu (ženský hormon) pro snížení libida [3].

7. června 1954 spáchal Alan Turing sebevraždu. Sám se otrávil kyanidem draselným tak, že jím napustil jablko a párkrát si kousl. Příčinu smrti prokázala až pitva, která vyloučila vraždu či pouhou náhodu způsobenou neopatrným zacházením s chemikáliemi.

Tak skončil život jednoho z nejvýznamnějších matematiků Alana Turinga, ale jeho příběh se táhne až do 21. století [3].

V roce 2009 mu byla jménem vlády vyjádřena omluva britským premiérem Gordonem Brownem. Omluva vyšla v novinách *The Daily Telegraph*:

„Jménem britské vlády a všech těch, kdo díky Alanově práci žijí svobodně, říkám: Je nám to líto. Zasloužil jste si něco lepšího. Není přehnané říci, že bez jeho mimořádného přispění mohly být dějiny druhé světové války velice odlišné. To, za co mu musíme být vděční, staví do ještě hroznějšího světla fakt, že se s ním jednalo tak nelidsky.“ [3]

Dne 24. prosince 2013 byla Alanu Turngovi udělena posmrtná milost královnou Alžbětou II. Britský ministr spravedlnosti dodal:

„Jeho genialita pomohla ukončit válku a zachránila tisíce životů. Jeho pozdější život byl zastíněn jeho odsouzením za homosexualitu. Tento rozsudek bychom nyní považovali za nespravedlivý a diskriminační, a proto byl rozsudek odvolán. Turing si zaslouží být uznáván za jeho přínosy ve válečném tažení a ve vědě o počítačích. Milost od královny je adekvátní hold tomuto skvělému muži.“ [3].

2 Definice pojmů objevujících se v práci

Teoretická informatika používá zažitou terminologii, bez jejíž znalosti není možné proniknout pod povrch zkoumaného tématu.

2.1 Problém

Problémem bychom také mohli rozumět úlohu, u které existuje požadavek na řešení a existuje pro ně algoritmus. Problémy dělíme na řešitelné a neřešitelné [12;13].

V případě řešení problému má smysl uvažovat, zda lze daný problém vůbec formálně zapsat a zda existuje možnost problém vyřešit výpočtem. I když všechny tyto požadavky problém splňuje, není ještě zaručeno, že bude možné výsledek získat [12;13].

Řešitelné problémy jsou takové, pro které známe (jsme schopni sestavit) algoritmus, který poskytne jednoznačný výsledek a současně je tento výsledek poskytnut v čase, v jakém nás tento výsledek stále zajímá [12;13].

Za neřešitelné považujeme takové problémy, které sice umíme zapsat, avšak jejich řešení je tak složité, že nemáme rozumnou možnost dočkat se výsledku výpočtu. Právě v případě neřešitelných problémů je zkoumána technika nejrůznějších typů optimalizací, abychom mohli získat alespoň přijatelný výsledek v akceptovatelném čase [12;13].

Problémy je podle složitosti zvykem řadit do tříd podle toho, jak složitý je nejlepší známý algoritmus. Máme za to, že problémy v dané třídě jsou na sebe vzájemně převoditelné [12;13].

Řešitelností problému se detailně zabývají matematické oblasti vyčíslitelnost a složitost, na které se zaměříme dále [12;13].

2.2 Vyčíslitelnost

Vyčíslitelnost neboli řešitelnost je jednou z charakteristik problému. Odpovídá na otázku, zda k danému problému existuje algoritmus, kterým jej lze vyřešit. Pojem algoritmus bude vysvětlen později [14;15].

Teorie vyčíslitelnosti, jakožto matematicko-informatický vědní obor, zkoumá hranice využití algoritmických postupů [14;15].

2.3 Spočetnost

Množiny si můžeme kromě konečných a nekonečných dále rozdělit na spočetné a nespočetné. Pokud je množina spočetná, bude s nejvyšší pravděpodobností i její algoritmus konečný [14;15].

2.3.1 *Spočetná množina*

Spočetnou množinu poznáme jednoduše tak, že ji můžeme zobrazit na některou podmnožinu přirozených čísel. Můžeme říci, že prvky této množiny můžeme spočítat, neboli očíslovat přirozenými čísly. Pokud nám na očíslování stačí konečný počet přirozených čísel, jedná se o konečnou spočetnou množinu. Pokud ne, jedná se o nekonečnou spočetnou množinu [16].

Příkladem můžou být celá čísla, která můžeme očíslovat následovně:

$$\begin{array}{rcl} 0 & \rightarrow & 0 \\ -1 & \rightarrow & 1 \\ 1 & \rightarrow & 2 \\ -2 & \rightarrow & 3 \\ 2 & \rightarrow & 4 \end{array}$$

2.3.2 *Nespočetná množina*

Nespočetná množina funguje přesně naopak. Nelze ji zobrazit na žádnou podmnožinu přirozených čísel a její prvky nemůžeme spočítat ani očíslovat přirozenými čísly, nezávisle na tom, zda použijeme konečný či nekonečný počet přirozených čísel [17].

Zkráceně můžeme říct, že množina je nespočetná tehdy, když neexistuje prosté zobrazení z dané množiny na množinu přirozených čísel. Příkladem by mohla být reálná čísla [17].

Pro dokázání nespočetnosti množiny reálných čísel byla použita Cantorova diagonální metoda, velmi podobná byla použita i pro dokázání problému zastavení [1].

2.4 Algoritmus

Algoritmem rozumíme určitý postup kroků, jakým je možné problém řešit. Musí splňovat následující vlastnosti:

- **elementárnost** – skládá se z konečného počtu kroků
- **konečnost** - po provedení konečného počtu kroků musí algoritmus skončit
- **obecnost** - je funkční pro proměnné na vstupu a není vázán na jejich hodnoty
 - příklad: pro sčítání dvou proměnných „ a “ a „ b “, kdy $a, b \in \mathbb{Z}$, platí stále stejný algoritmus
- **opakovatelnost výsledku** - na stejné vstupy vrací vždy stejné výstupy (bez ohledu na to, kolikrát výpočet provedeme, vždy získáme stejný výsledek)
- **determinovanost** (jednoznačnost) – musí být přesně definováno:
 - každý krok
 - v jakém pořadí za sebou kroky následují
 - podmínky, za kterých má být daný krok proveden
- **výstup** – algoritmus musí nějakým (nezáleží na tom jakým) způsobem informovat o výsledku

[18;19]

2.5 Složitost

Složitost je metrika pro porovnání efektivity algoritmů. Tak jako ve většině oblastí, i v programování existuje často více způsobů jak dosáhnout požadovaného cíle. Přirozeně je žádoucí nalézt takový algoritmus, který nalezne výsledek co nejrychleji s co nejmenšími výpočetními prostředky a výsledek je vždy správný a nesporný [19;20].

Již bylo zmíněno, že některé problémy jsou považovány za algoritmicky neřešitelné. V takových případech musíme vzít za vděk i algoritmy, které nám poskytnou alespoň

dobré řešení, a my si musíme být vědomi, že dosažené řešení pravděpodobně není optimální [19;20;21].

Příkladem by mohl být známý Problém obchodního cestujícího³ nebo Pašerákův problém, pro které nejsme schopni sestavit efektivní algoritmus [22].

Existují dva přirozené přístupy, jak složitost měřit. V případě, že zkoumáme za jak dlouho (po kolika výpočetních krocích) algoritmus poskytne výsledek, hovoříme o tzv. časové složitosti. Druhým přístupem je hodnotit kolik paměti je pro výpočet potřeba, tedy prostorová složitost [19;20;21].

2.5.1 Třídy složitosti

Při popisu složitosti problému narážíme na skutečnost, že jen samotná definice složitosti je složitý úkol. Slovní spojení „složitost je složitá“ je krásným příkladem rekurze. A rekurze je naopak jedním z nástrojů popisu složitosti. Toto téma by si zasloužilo vlastní detailní zpracování a zcela překračuje rámec této práce. Pro naši potřebu bude stačit vědet pouze toto. Některé problémy jsou si do určité míry podobné (z hlediska složitosti a řešení) a je proto výhodné je rozdělit do tříd. Po rozdělení problémů bylo zjištěno, že jsou si problémy v rámci jedné třídy natolik podobné, že jsou na sebe často převoditelné. Nejsme si zatím zcela jisti, zda jsou na sebe převoditelné v rámci jedné třídy úplně všechny problémy, ale je to velmi pravděpodobné [19;20;21].

Problém považujeme za převoditelný, pokud existuje polynomiálně složitý algoritmus, který převede jeden problém na druhý. V případě převodu polynomiálně složitého problému na jiný polynomiálně složitý problém, je efektivita tohoto problému obecně diskutabilní. Převod však může být významným pomocníkem v případě NP problémů, pro které obecně nemáme vhodné algoritmy, ale v některých případech se nám daří dosáhnout alespoň dostačujících výsledků a dosahujeme tak určité řešitelnosti některých problémů (např. plánování trasy sypacího vozu ulicemi) [19;20;21].

³ Cílem je najít hamiltonovskou kružnici (žádný vrchol nesmí navštívit dvakrát) mezi n městy (mezi kterými vede cesta o známé délce), kterými musí projít cestovatel, aby mu cesta zabrala co nejméně času.

2.5.2 *Asymptotická složitost*

Asymptotická složitost udává, jaká je nejhorší možná složitost zkoumaného algoritmu. Tedy například algoritmus s lineární složitostí poskytne výsledek v nejhorším případě k -krát později, když přivedeme k -krát delší vstup [23].

Tabulka 1 - Třídy asymptotické složitosti

Asymptotická složitost	zápis
konstantní	$O(1)$ – nezávisí na délce vstupu n
logaritmická	$O(\log n)$
lineární	$O(n)$
lineárně logaritmická	$O(n * \log n)$
Kvadratická	$O(n^2)$
Kubická	$O(n^3)$
Polynomiální	$O(n^k)$; $k > 0$
Exponenciální	$O(k^n)$; $k > 1$
Faktoriálová	$O(n!)$

2.6 Automat

Automatem rozumíme stroj, který po spuštění samostatně vykonává předem určenou činnost [24].

Automat na vstupu dostane zadání problému a pomocí algoritmu (který je představován právě konkrétním automatem) dojde k výsledku. Výsledek může být reprezentován různými způsoby, v nejjednodušší variantě je výstup reprezentován koncovým stavem, ve kterém se automat nachází [24].

2.7 Syntaxe

Soubor pravidel, do jaké struktury slova a hodnoty patří. V případě jednoduchých gramatik se může jednat o pouhé pořadí. V případě složitějších gramatik (např. kontextové gramatiky) je již třeba syntaxi chápat v kontextu toho, co bylo zapsáno dříve a co bude následovat (např. výraz v závorce) [25].

2.8 Sémantika

V případě syntaxe byla zkoumána pravidla, jak co zapsat. Sémantika naproti tomu přiřazuje zapsané informaci její význam. Intuitivně se sémantikou v přirozených jazycích setkáváme a nemusíme o ní přemýšlet. V případě umělých jazyků je nutno rozumět již zapsané informaci. Sémantika je tedy soubor pravidel, který zapsané informaci přiřazuje význam (např.: „int a=3“ znamená: proměnná „a“ je přirozené číslo a je mu přiřazena hodnota 3)⁴ [25].

2.9 Abeceda

Jedná se o konečnou množinu symbolů. V tomto místě by bylo možno definici považovat za dokonalou, ale pro lepší pochopení budeme pod pojmem abeceda rozumět takovou množinu symbolů, se kterými může uvažovaný stroj pracovat. Ze symbolů abecedy se pak tvoří slova na základě gramatických pravidel a vzniká tak jazyk. Stroje nezřídka používají ještě rozšířenou abecedu (např. doplněnou o prázdný symbol) i v tomto případě hovoříme o abecedě, protože splňuje naši prvotní definici [25].

⁴ int (integer) je v překladu přirozené číslo

3 Chomského hierarchie

Chomského práce byla napsána až v roce 1956, ale patří na toto místo ihned za definici pojmů, neboť teprve Noama Chomského napadlo tyto problémy a jazyky klasifikovat. Pro lepší pochopení problematiky je vhodné se Chomského prací seznámit v této fázi poznání, neboť usnadňuje chápání celé problematiky [25;26].

Chomskému se jako prvnímu podařilo podchytit a dokázat, že problémy a algoritmy pro jejich řešení si mezi sebou jsou často podobné. Máme za to, že problémy jsou na sebe navzájem převoditelné, ovšem pouze v rámci jedné třídy problémů. Do těchto tříd je rozdělil právě již zmiňovaný Noam Chomsky [25;26].

Pro správné pochopení hierarchického upořádání je třeba nejprve upřesnit vztah mezi slovem (w), abecedou (Σ), gramatikou (G) a jazykem (L):

Slovo (w) je řetězec znaků (posloupnost) z abecedy (Σ). To je možné formálně vyjádřit takto: $w = s^*$ pokud $s \in \Sigma$ [25;26]

Množina všech slov nad abecedou Σ je značena Σ^* [25;26].

Existuje speciální případ slova, které neobsahuje žádný znak. Takováto slova nazýváme prázdná. Obvykle jsou označovány speciálním znakem λ nebo ε [25;26].

Délku slova značíme $|w|$. Vyjadřuje počet znaků, Např.: $|ababa| = 5$.

Jazyk (L) je definován jako množina slov nad abecedou. Pro všechny jazyky platí, že obsahují množinu slov (s^*), která jsou tvořena znaky z abecedy. Formálně tedy můžeme zapsat: $L \subseteq \Sigma^*$ [25;26]

Konečně se dostáváme k samotnému pojmu **gramatika**. Gramatika je soubor pravidel, kterými se vytvářejí všechny řetězce (slova) daného jazyka. Takový stav popisujeme jako „Gramatika generuje jazyk“ [25;26].

$$G = (N, T, S, P)$$

Kde:

G.....je gramatika

N.....je pomocná abeceda, přesněji množina nonterminálů⁵

T.....je vlastní abeceda jazyka, neboli množina terminálních symbolů⁶

S.....je výchozí neterminální symbol, lze ho chápat jako symbol START

P.....je soubor pravidel, pravidla mají obecný tvar $u \rightarrow v$. kde $u, v \in (N \cup T)$

Abecedy N a T jsou disjunktní tedy $N \cap T = \{\}$. Neplatí to obecně, ale lze si představit, že $\Sigma = T$

[25;26]

3.1 Třídy gramatik

Noam Chomsky rozdělil formální gramatiky a jimi generované formální jazyky celkem do 4 tříd podle typu, které rozlišil čísly 0, 1, 2 a 3.

Pro jazyky platí vztah:

$$L_0 \supset L_1 \supset L_2 \supset L_3$$

A přirozeně pro gramatiky platí vztah:

$$G_0 \supset G_1 \supset G_2 \supset G_3$$

Intuitivně z toho tedy vyplývá, že každá regulární gramatika je současně bezkontextová, kontextová i fázová [27].

3.1.1 Gramatiky typu 0

Gramatiky typu 0 neboli obecné, fázové či neomezené gramatiky, jsou právě ty, které nás budou zajímat nejvíce. Tento typ gramatik generuje jazyky, se kterými umí pracovat Turingův stroj a může rozhodnout o její spočetnosti. Důkaz o spočetnosti je velmi

⁵ Pomocné symboly, které nepatří do abecedy generovaného jazyka, mají roli jakýchsi žolíků, které musí být během výpočtu nahrazeny dalšími symboly (terminály i nonterminály). Nejsou to finální (terminální) symboly. Vyjadřují fakt, že na tomto místě v řetězci se ještě bude pracovat. V programovacích jazycích je zvykem je označovat pojmem „proměnná“.

⁶ Ty jsou již umístěny na svém místě a nebudou ničím nahrazeny.

podobný pro dokázání problému zastavení. Tyto jazyky nazýváme rekurzivní nebo také rekurzivně vyčíslitelné (spočetné) jazyky [26;27].

3.1.2 Gramatiky typu 1

Kontextové gramatiky generují kontextové jazyky. Kontextové jazyky jsou, obecně řečeno, jednodušší v porovnání s typem 0. Pro jejich vyčíslení dostačuje lineárně omezený automat, což je zjednodušená varianta Turingova stroje. Lineárně omezený automat má oproti Turingovu stroji omezenou délku pásky. Právě technika převodu jazyka na lineárně omezený automat se používá k důkazu, že zkoumaný jazyk není rekurzivní [26;27].

3.1.3 Gramatiky typu 2

Bezkontextové jazyky generované odpovídajícími bezkontextovými gramatikami jsou vyčíslitelné ještě jednodušším typem stroje – zásobníkovým automatem. Zásobníkový automat pracuje podobně jako konečný automat, ale je rozšířen o paměťovou strukturu zvanou zásobník. Zásobník je zvláštní typ paměti, který umožňuje číst a zapisovat pouze do horního políčka. Pokud je potřeba pracovat s nižšími políčky, musí být zpracována všechna políčka nad nimi. Činnost zásobníkového automatu skutečně připomíná zásobník na papír, ze kterého vidíme pouze údaje na horním listu [26;27].

3.1.4 Gramatiky typu 3

Regulární gramatiky generují regulární jazyky, ty jsou vyčíslitelné konečnými automaty. Konečný automat je takový stroj, který na základě pravidel reaguje na vstupní symboly a přechází mezi různými stavy. Typickými znaky konečného automatu jsou jeden počáteční stav a nejméně jeden koncový stav. Mezi těmito stavy může být libovolný konečný počet přechodových stavů. U konečného automatu považujeme vstup za přijatý (platný) pokud automat dosáhl koncového stavu. Jedná se o nejjednodušší typ automatu [26;27].

3.2 Srovnávací tabulka jazyků, gramatik a strojů dle Chomského

Tabulka 2 - Srovnávací tabulka jazyků, gramatik a strojů dle Chomského

Jazyk	Gramatika	Stroj
L0 rekurzivně spočetný jazyk	Typ 0 frázové gramatiky	Turingův stroj
L1 kontextový jazyk	Typ 1 kontextové gramatiky	Lineárně ohraničený automat
L2 bezkontextový jazyk	Typ 2 bezkontextové gramatiky (bez ϵ -pravidel)	Zásobníkový automat
L3 regulární jazyk	Typ 3 regulární gramatiky	Konečný automat

[27]

4 Turingův stroj

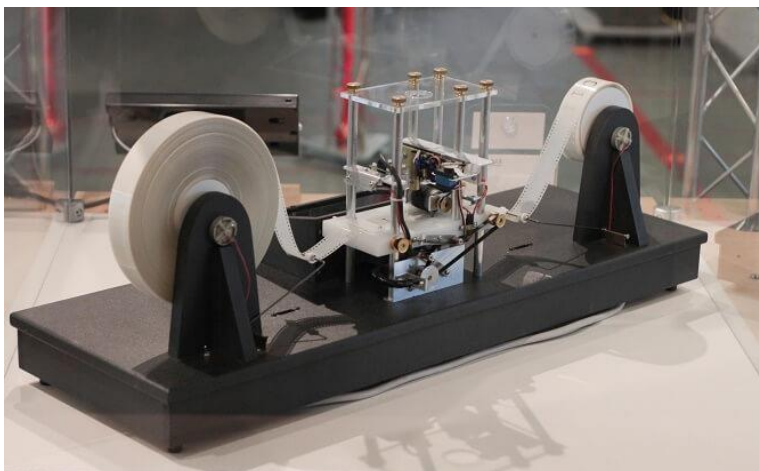
Jedná se o teoretický model výpočetního stroje nebo také model univerzálního počítače s neomezenou pamětí. Poprvé se o jeho konceptu můžeme dočíst v roce 1936, kdy Alan Turing vydal svůj článek: „On Computable numbers“ [25;28].

Turingův stroj tvoří řídicí mechanismus a vstupní tok informací – nekonečně dlouhá páska. Na rozdíl od automatů může Turingův stroj pohybovat hlavou oběma směry (doprava i doleva) a může na pásku zapisovat i z ní číst. Také je univerzální (je programovatelný). Lze dokázat, že vše, co umí vypočítat dnešní počítače, je možné vypočítat i na Turingově stroji. Ovšem ani Turingův stroj ani dnešní počítače neumí řešit všechny problémy. Jedná se o problémy, které jsou za hranicemi vyčíslitelnosti [28;29;30].

4.1 Skladba Turingova stroje

Turingův stroj uvažujeme pouze jako teoretický model. Je samozřejmě možné jej zkonstruovat, avšak je to zbytečné, neboť v praxi máme výkonnější výpočetní stroje. Význam Turingova stroje jako teoretického modelu je především v jeho jednoduchosti a představitelnosti. Stroj se skládá se ze tří částí: hlavy, pásky a přechodové funkce [28;31].

Kdybychom se přece jenom Turingův stroj pokusili sestrojít, mohl by vypadat asi následovně:



Obrázek 1 – Turingův stroj

Hlava

Hlava stroji slouží jako čtecí a zapisovací zařízení. Posunuje se po pásce a čte nebo do ní zapisuje informace [28].

Páska

Páska je „paměť“ výpočetního stroje, kterou uvažujeme neomezenou – nekonečně dlouhou. Hlava se po ní může pohybovat oběma směry (doprava i doleva). Je rozdělena na jednotlivé buňky, které jsou řazeny za sebou (sousedí spolu vždy zleva a zprava) [15;28].

Přechodová funkce

Přechodová funkce je nejdůležitější částí. Jedná se o předpis Turingova stroje. Definuje přesná pravidla, podle kterých se má Turingův stroj chovat, neboli jak má vypadat jedna iterace – změna stavu, posunutí hlavy po pásce a přepsání obsahu políček [15;32].

4.2 Základní definice Turingova stroje

Abychom se mohli ponořit do světa Turingova stroje, je nezbytné si definovat několik základních pojmů a objektů, se kterými stroj pracuje:

- Q**..... konečná množina stavů
- Σ** konečná neprázdná množina symbolů
- Γ** konečná množina symbolů obsahující množinu Σ a $Q \cap \Gamma = \emptyset$ (pásková abeceda)
- δ** přechodová funkce
- q_0** počáteční stav
- F**..... množina koncových stavů $F \subseteq Q$
- B**..... množina prázdných znaků (nějakým symbolem je vyjádřeno, že je políčko prázdné)

[15;33]

Q - Konečná množina stavů

Konečná množina stavů, do kterých se Turingův stroj může dostat. Řídící mechanismus se pak může v určitém okamžiku nacházet v libovolném stavu z této množiny. Tato množina musí obsahovat minimálně dva stavy. Jímí jsou počáteční a koncový stav, přičemž počátečním stavem logicky uvažujeme stav, ve kterém výpočetní proces začíná a koncovým, kde výpočet končí. Tím, že se stroj nachází v nějakém stavu Q_x je reprezentován kontext jeho momentálního stavu. Stav stroje tak vyjadřuje v podstatě operační paměť, neboť stroj se do tohoto stavu dostal po provedení nějaké části algoritmu během postupného zpracování vstupních hodnot na pásce [15].

Σ - Konečná množina symbolů

Konečnou (neprázdnou) množinu symbolů můžeme také nazvat vstupní abecedou. Udává nám množinu symbolů, kterými jsou zapisována vstupní data na pásku stroje.

Do této množiny můžeme a nemusíme řadit prázdný znak tzv. „blank“ (značeno „B“). Jedná se o prázdné políčko na pásce, které nám udává prostor, kam hlava může zapisovat. Pokud chce čtecí a zapisovací zařízení něco přepsat, napíše nejprve blank a poté zapíše nový symbol. Záleží na gramatice, zda je prázdný symbol významný pro zápis vstupních dat. Pokud symbol „B“ není nutný pro zápis vstupních dat, tak do množiny Σ nepatří [15].

Γ - Konečná množina páskových symbolů

Množina páskových symbolů (pásková abeceda) zcela intuitivně obsahuje všechny prvky ze vstupní abecedy Σ a rozšiřuje ji o další pomocné symboly, které Turingův stroj může během výpočtu používat (například znak „a“ kde $a \in \Sigma$ je nahrazen znakem „A“ kde „A“ $\in \Gamma$ a současně „A“ $\notin \Sigma$). Na pásce je tak odlišeno zpracované „A“ od původního nezpracovaného „a“. Stroj si tak může psát své poznámky přímo na pásku [15].

δ – Přejímová funkce

Soubor pravidel, jak se má stroj zachovat, jestliže se nachází v nějakém stavu q_x a na vstupu je symbol náležící do Γ .

Přechodové pravidlo pro jednopáskový Turingův stroj má tvar:

$$\delta: (Q - F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, N, R\}$$

Kde:

$(Q - F)$ je množina nekoncových stavů (stavy, ve kterých očekáváme, že výpočet z nich bude pokračovat)

Γ je pásková abeceda a do přechodové funkce vstupuje právě jeden symbol z této abecedy. Je to právě ten symbol, který aktuálně čte hlava Turingova stroje.

Symbol \rightarrow čteme jako: „přepisuje se“ a vyjadřujeme tím oddělení vstupu a toho, co se na základě tohoto vstupu má stát v tomto kroku. Symbol bychom také mohli chápat jako oddělovač mezi podmínkami (levá strana přechodové funkce) a příkazy (pravá strana přechodové funkce).

Q je množina stavů. V tomto případě (jelikož se nacházíme v příkazové části přechodové funkce) se jedná o stav, do kterého se stroj přesune (přepne) vykonáním prováděného příkazu. Za zmínku zajisté stojí, že do množiny těchto stavů Q patří i všechny koncové stavy (povinný je jeden), v takovém případě výpočet končí.

Γ na pravé straně symbolizuje příkaz k zápisu znaku na pásku.

$\{L, N, R\}$ symbolizují příkaz k pohybu. „L“ znamená pohyb do leva, „N“ znamená nepohybovat hlavou a symbol „R“ příkazuje pohyb do prava.

[15;28]

Při výpočtu na Turingově stroji může pravidlo přechodové funkce vypadat například takto:

$$\delta : q_0; a \rightarrow q_1; b; R$$

To znamená, že pokud je Turingův stroj ve stavu q_0 a na vstupu je znak „a“, tak se stroj přepne do stavu q_1 , na pásku zapíše znak „b“ a posune hlavu na pásce směrem do prava [15;32].

4.3 Church - Turingova teze

Jedná se o hypotézu z oboru vyčísitelnosti, pojmenována po Alanu Turingovi a Alonzu Churchovi. Zabývá se řešitelností problémů – jestli, a případně jak, můžeme problémy řešit [31;32;34].

Dle Church – Turingovy teze platí, že libovolný algoritmus je možné zapsat předpisem pro Turingův stroj, pokud platí:

1. „Algoritmus se skládá z konečného počtu instrukcí, které jsou přesně definovány použitím konečného počtu symbolů.“ [32]
2. „Algoritmus vždy vrátí výsledek po konečném počtu kroků.“ [32]
3. „Algoritmus může provádět i člověk s tužkou a papírem.“ [32]
4. „Spuštění algoritmu nepotřebuje lidskou inteligenci, s výjimkou té, která je třeba na pochopení a vykonání instrukcí.“ [32]

4.4 Konfigurace Turingova stroje

Každý deterministický Turingův stroj v každém čase musí mít přesně definovanou konfiguraci. Konfigurací v případě Turingova stroje rozumíme stav (z množiny Q), momentální stav pásky (všechny symboly z abecedy Γ , tak jak jsou na pásce za sebou uvedeny) a polohu hlavy vzhledem k pásce [32;33;34].

Za normálních podmínek se konfigurace v každé iteraci mění. Jedinou výjimkou, kdy by nedocházelo ke změně konfigurace a stroj se nenacházel v koncovém stavu, by byla práce v nekonečné smyčce, což je přirozeně nežádoucí a ukazuje na chybu přechodové funkce [32;33;34].

S pojmem konfigurace budeme běžně pracovat v praktické části. Proto bylo nutné si tento pojem definovat nyní [32;33;34].

4.5 Vícepáskové Turingovy stroje

Jak už víme, Turingův stroj se skládá ze tří částí – jedné hlavy, jedné pásky a má svou přechodovou funkci. Co by se stalo, kdyby se počet pásek a jim příslušících hlav zdvojnásobil, nebo dokonce zosminásobil? Bude stroj pracovat rychleji? Výkonněji?

Stejně otázky si kladli matematici již v minulém století a dospěli k překvapivým závěrům. K jejich pochopení si nejprve vysvětlíme, jak by takový stroj vůbec fungoval.

Uvažujme klasický jednopáskový Turingův stroj rozšířený o libovolný počet pásek a stejný počet hlav. Pro zjednodušení uvažujme, že všechny hlavy umožňují číst a zapisovat znaky na pásky.

Pro přehlednost jsou v níže uvedeném přehledu shrnuty odlišnosti od jednopáskového Turingova stroje.

Q	konečná množina stavů
Σ	konečná neprázdná množina symbolů může být pro každou pásku odlišná ⁷
Γ	v případě, že pásky pracují s různými abecedami, tak nutně musí být odlišné i páskové abecedy ⁸
δ	přechodová funkce je zcela jistě složitější
q₀	počáteční stav v obecné definici zůstává nedotčen
F	množina koncových stavů $F \subseteq Q$ zůstává nedotčena
B	množina prázdných znaků vyplývá z definice konkrétního stroje

[15]

⁷ např. Σ_A je abecedou pro první pásku a Σ_B je abecedou pro druhou pásku

⁸ např. Γ_A a Γ_B

Přidáním většího množství pásek je možno výpočet zrychlit a zefektivnit, ale zvýšením počtu pásek se nikterak nerozšíří schopnosti – nezmění se množina problémů, které jsou na Turingově stroji řešitelné. Každý vícepáskový Turingův stroj tedy můžeme převést na jednopáskový [15;28].

Vícepáskové Turingovy stroje lépe odpovídají reálným výpočetním strojům, a proto má smysl se jimi zabývat, nicméně pro potřeby důkazů v teoretické informatice se považuje důkaz na jednopáskovém Turingově stroji za plnohodnotný a dostačující [15;28].

4.6 Problém zastavení

Problém zastavení je i v tuzemské literatuře často označován anglickým názvem „Halting problem“. Problém zastavení lze popsat jako otázku, zda se Turingův stroj zastaví. Turingův stroj považujeme za zastavený, pokud dosáhl koncového stavu (stav náleží do množiny F) nebo jen stavu, ze kterého nevede vhodná přechodová funkce [35;36].

Jen dobrý algoritmus zajišťuje dosažení koncového stavu za každých okolností. Každodenní praxe v oboru informačních technologiích dokazuje, že ani v současné době s veškerým výpočetním vybavením, které máme k dispozici, se nedaří tvořit algoritmy dokonale. Teoretická informatika s využitím aparátu Turingových strojů umožňuje vnášet postupnou nápravu a takovými stavům předcházet (ošetřování výjimek). Obecně požadujeme od každého algoritmu simulovaného Turingovým strojem, aby zajišťoval, že se Turingův stroj po vykonání konečného počtu kroků zastaví v koncovém stavu. V případě, že existuje možnost, kdy se Turingův stroj nezastaví, případně se zastaví ve stavu, který nenáleží do množiny koncových stavů, tak není algoritmus úplný [35;36].

V roce 1936 Alan Turing dokázal, že neexistuje obecný algoritmus, který by dokázal rozhodnout o předloženém algoritmu, zda se na všechny vstupy zastaví. Alan Turing použil v důkazu techniku převodu na paradox [34;35;36].

4.7 Použití v praxi

Již bylo řečeno, že praktické využití Turingových strojů je pouze v teoretické informatice, hlavně v oborech vyčíslitelnost, složitost a řešitelnost. Slouží jako nástroj pro zkoumání, simulaci a důkazy otázek, kterými se teoretická informatika zabývá. Některé z nich byly

zde v práci zmíněny např.: halting problem, teorie kompilátorů (překladačů) a umělých jazyků [15;28].

U nových programovacích jazyků se také určuje tzv. Turingovská úplnost. Jedná se o kritérium, které by měl daný jazyk splňovat, aby bylo možné ho považovat za srovnatelný s ostatními již používanými programovacími jazyky. Pokud je tedy možné testovaným jazykem naprogramovat Turingův stroj, pak je turingovsky úplný a je stejně dobrý jako ostatní programovací jazyky [15;28].

Mezi Turingovsky úplné jazyky řadíme např.: C++, Python, Javu nebo Javascript a mezi nejznámější turingovsky neúplné patří jazyk HTML (jedná se pouze o značkovací jazyk) [15;28].

4.8 Speciální případ Turingova stroje

Turingův stroj je z podstaty své definice deterministický a je tak omezen pouze na řešení problému, které lze algoritmizovat. Teoretická informatika se od prvopočátků zabývá i problémy, které řešit neumí a snaží se nalézt způsob, jak poskytnout výsledek pro takovéto úlohy (např. NP úplné problémy). Pro takové případy je vyhrazen pojem „Nedeterministický Turingův stroj“ a tato práce se jím dále nezabývá [28].

4.9 Alternativy Turingova stroje

Nezávisle na Turingově stroji vznikly alternativní stroje a podle současného poznání je jejich výpočetní síla ekvivalentní s Turingovým strojem. Stroje jsou na sebe navzájem převoditelné v nejhorším případě polynomiálně složitým algoritmem.

4.9.1 *RAM stroj*

RAM - Random access machine je teoretický model počítače. Stroj se skládá z jednoho speciálního registru (akumulátoru) a neomezené paměti pro proměnné. Akumulátor obsahuje instrukce. Představuje tak určitou obdobu přechodové funkce u Turingova stroje [37].

4.9.2 *Lambda kalkul*

V roce 1936 (ve stejném roce jako Alan Turing) s tímto strojem přišel Alonzo Church, Turingův profesor na Princetonské univerzitě [38;39].

Stroj se skládá celkem ze tří prvků – proměnná (přesněji nspecifikovaná hodnota), abstrakce (forma, v jaké jsou zapsány funkce pro Lambda kalkul) a aplikace (volání funkce s jedním argumentem) [38;39].

Stroj je základem několika programovacích jazyků (např. jazyka Lisp nebo Haskell) [38;39].

4.9.3 *Postův stroj*

Postův stroj uvedl ve stejném roce (1936) polský matematik Emil Leon Post. Někdy se také nazývá Post-Turingův stroj. Tento stroj je asi nejpodobnější Turingovu stroji, přestože také vznikl nezávisle na něm [40].

Praktická část

5 Metodika

V praktické části jsou využity nastudované informace z části teoretické. Budou provedeny výpočty na jednopáskovém Turingově stroji. Bude tak ověřena jeho funkčnost a předvedena získaná znalost. Praktická část se zaměří na různé situace, které mohou nastat během výpočtu a vyhodnotí jejich dopad na výsledek.

Jako první bude předveden případ, kdy Turingův stroj skončí v koncovém stavu a řetězec na pásce bude přijat. Jednotlivé kroky budou podrobně popsány, aby bylo možné lépe pochopit průběh výpočtu.

Dále je pro ukázkou záměrně simulován Turingův stroj končící v jiném než koncovém stavu neboli nepřijetí řetězce.

Nakonec je ukázán i Turingův stroj zacyklený v nekonečné smyčce, na kterém bude předvedena důležitost správné přechodové funkce, jelikož do ní bude úmyslně zavedena zřejmá chyba. To má svůj význam pro názornost příkladu. V praxi už takové chyby často zřejmé nejsou. Jejich důsledky však bývají katastrofální pro celý výpočet.

6.1 Sudý palindrom

Při řešení palindromů na Turingově stroji, můžeme pomocí jednoduchého algoritmu ověřit, zda se jedná o palindrom či nikoli. Nejprve se podíváme na ověření sudého palindromu.

6.1.1 Sudý palindrom končící v koncovém stavu

Uvažujme Turingův stroj $T: T = (Q, \Sigma, \Gamma, \delta, q_0, F)$

$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$$

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

$$\Gamma = \{0; 1; B\}$$

$$F = \{q_7\}$$

Přechodová funkce pro řešení sudého palindromu by mohla vypadat následovně:

Tabulka 3 – Přechodová funkce č. 1

stav	0	1	B
q_0	$q_0; 0 \rightarrow q_4; B; R$	$q_0; 1 \rightarrow q_1; B; R$	$q_0; B \rightarrow q_7; B; N$
q_1	$q_1; 0 \rightarrow q_1; 0; R$	$q_1; 1 \rightarrow q_1; 1; R$	$q_1; B \rightarrow q_2; B; L$
q_2		$q_2; 1 \rightarrow q_3; B; L$	
q_3	$q_3; 0 \rightarrow q_3; 0; L$	$q_3; 1 \rightarrow q_3; 1; L$	$q_3; B \rightarrow q_0; B; R$
q_4	$q_4; 0 \rightarrow q_4; 0; R$	$q_4; 1 \rightarrow q_4; 1; R$	$q_4; B \rightarrow q_5; B; L$
q_5	$q_5; 0 \rightarrow q_6; B; L$		
q_6	$q_6; 0 \rightarrow q_6; 0; L$	$q_6; 1 \rightarrow q_6; 1; L$	$q_6; B \rightarrow q_0; B; R$
q_7	Přijato		

Zadání: Rozhodněte, zda je zadaný řetězec sudý palindrom, když je na pásce řetězec „B 1 1 0 1 0 0 1 0 1 1 B“

Pořadové číslo kroku: 0 – počáteční konfigurace																							
Výchozí stav z Q: q_0	Výstupní stav z Q: q_0																						
Čtený symbol: 1	Zapsaný symbol: 1																						
Příkaz pohybu hlavy na pásce: N																							
↓																							
B	B	B	B	B	B	B	1	1	0	1	0	0	1	0	1	1	B	B	B	B	B	B	B

Řetězec máme zapsaný na pásce. Turingův stroj zatím nemůže vědět, kolik symbolů slovo obsahuje. Musí si proto poznamenat, že u tohoto prvního znaku již byl, tím že si znak nahradí prázdným symbolem, a jelikož čteným symbolem byl symbol „1“, bude se v příštím kroku řídit příslušným pravidlem přechodové funkce.

Pořadové číslo kroku: 1																							
Výchozí stav z Q: q_0	Výstupní stav z Q: q_1																						
Čtený symbol: 1	Zapsaný symbol: B																						
Příkaz pohybu hlavy na pásce: R																							
↓																							
B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	1	B	B	B	B	B	B	B

$$\delta: q_0; 1 \rightarrow q_1; B; R$$

První symbol našeho řetězce byl tedy přepsán na prázdný znak, hlava se posunula směrem doprava a stav Turingova stroje se změnil – potřebuje si zapamatovat, jaký znak četl. V dalších krocích se musí stroj již řídit jinými příslušnými pravidly přechodové funkce.

V následujících krocích se stroj potřebuje dostat na konec řetězce, aby mohl ověřit, zda se symbol na druhé straně řetězce nachází stejný symbol v našem případě „1“. V prvním příkladu budou popsány všechny kroky Turingova stroje v zájmu názornosti.

Pořadové číslo kroku: 2																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 1	Zapsaný symbol: 1																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	1	B	B	B	B	B	B	B

$$\delta: q_1; 1 \rightarrow q_1; 1; R$$

Stav Turingova stroje q_1 zůstává zachován, symbol „1“ na pásce není přepsán, hlava stroje se pouze posune směrem doprava. Stejný děj bude probíhat v krocích 3-10 (stroj v podstatě pouze přechází na druhý konec řetězce), pravidlo přechodové funkce se ovšem bude měnit podle čteného znaku na pásce.

Pořadové číslo kroku: 3																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 0	Zapsaný symbol: 0																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	1	B	B	B	B	B	B	B

$$\delta: q_1; 0 \rightarrow q_1; 0; R$$

Pořadové číslo kroku: 4																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 1	Zapsaný symbol: 1																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	1	B	B	B	B	B	B	B

$$\delta: q_1; 1 \rightarrow q_1; 1; R$$

Pořadové číslo kroku: 5-6																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 0	Zapsaný symbol: 0																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	1	B	B	B	B	B	B	B

$\delta: q_1; 0 \rightarrow q_1; 0; R$

Pořadové číslo kroku: 7																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 1	Zapsaný symbol: 1																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	1	B	B	B	B	B	B	B

$\delta: q_1; 1 \rightarrow q_1; 1; R$

Pořadové číslo kroku: 8																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 0	Zapsaný symbol: 0																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	1	B	B	B	B	B	B	B

$\delta: q_1; 0 \rightarrow q_1; 0; R$

Pořadové číslo kroku: 9-10																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 1	Zapsaný symbol: 1																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	1	B	B	B	B	B	B	B

$\delta: q_1; 1 \rightarrow q_1; 1; R$

Pořadové číslo kroku: 11																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_2																							
Čtený symbol: B	Zapsaný symbol: B																							
Příkaz pohybu hlavy na pásce: L																								
↓																								
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	1	B	B	B	B	B	B	B

$$\delta: q_1; B \rightarrow q_2; B; L$$

Hlava Turingova stroje přečetla prázdný symbol „B“. Díky tomu stroj poznal, že došel na konec řetězce a bude muset opět změnit stav, jelikož mu to odpovídající pravidlo přechodové funkce příkazuje a pohne se doleva. Aby se jednalo o palindrom, musí dalším čteným znakem být „1“.

Pořadové číslo kroku: 12																								
Výchozí stav z Q: q_2	Výstupní stav z Q: q_3																							
Čtený symbol: 1	Zapsaný symbol: B																							
Příkaz pohybu hlavy na pásce: L																								
↓																								
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B

$$\delta: q_2; 1 \rightarrow q_3; B; L$$

Podmínka byla splněna a hlava Turingova stroje přepíše znak „1“ a označí si políčko na pásce prázdným znakem. Opět se změní i stav Turingova stroje a hlava se posune směrem doleva.

Pořadové číslo kroku: 13																									
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																								
Čtený symbol: 1	Zapsaný symbol: 1																								
Příkaz pohybu hlavy na pásce: L																									
↓																									
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B	B

$$\delta: q_3; 1 \rightarrow q_3; 1, L$$

Stroj se nyní musí propracovat zpátky na začátek řetězce. V následujících krocích (14-20) se bude posouvat vždy směrem doleva, dokud opět nedojde k prázdnému- znaku „B“ (zde je zajímavé, že narazí na znak „B“, který si sám vytvořil v kroku 1).

Pořadové číslo kroku: 14																									
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																								
Čtený symbol: 0	Zapsaný symbol: 0																								
Příkaz pohybu hlavy na pásce: L																									
↓																									
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B	B

$$\delta: q_3; 0 \rightarrow q_3; 0, L$$

Pořadové číslo kroku: 15																									
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																								
Čtený symbol: 1	Zapsaný symbol: 1																								
Příkaz pohybu hlavy na pásce: L																									
↓																									
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B	B

$$\delta: q_3; 1 \rightarrow q_3; 1, L$$

Pořadové číslo kroku: 16-17																										
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																									
Čtený symbol: 0	Zapsaný symbol: 0																									
Příkaz pohybu hlavy na pásce: L																										
↓																										
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B	B	B

$\delta: q_3; 0 \rightarrow q_3; 0, L$

Pořadové číslo kroku: 18																										
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																									
Čtený symbol: 1	Zapsaný symbol: 1																									
Příkaz pohybu hlavy na pásce: L																										
↓																										
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B	B	B

$\delta: q_3; 1 \rightarrow q_3; 1, L$

Pořadové číslo kroku: 19																										
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																									
Čtený symbol: 0	Zapsaný symbol: 0																									
Příkaz pohybu hlavy na pásce: L																										
↓																										
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B	B	B

$\delta: q_3; 0 \rightarrow q_3; 0, L$

Pořadové číslo kroku: 20																										
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																									
Čtený symbol: 1	Zapsaný symbol: 1																									
Příkaz pohybu hlavy na pásce: L																										
↓																										
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B	B	B

$\delta: q_3; 1 \rightarrow q_3; 1, L$

Pořadové číslo kroku: 21																								
Výchozí stav z Q: q_3	Výstupní stav z Q: q_0																							
Čtený symbol: B	Zapsaný symbol: B																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B

$$\delta: q_3; B \rightarrow q_0; B; R$$

Hlava Turingova stroje čte prázdný symbol, který značí, že se opět nachází na současném počátku zbývajících řetězce. Posune se směrem doprava a vrátí se do počátečního stavu. Doposud provedené kroky pro něj již nejsou důležité, nemusí si je pamatovat. Podle následujícího symbolu bude pokračovat dle příslušného pravidla přechodové funkce.

Pořadové číslo kroku: 22																							
Výchozí stav z Q: q_0	Výstupní stav z Q: q_1																						
Čtený symbol: 1	Zapsaný symbol: B																						
Příkaz pohybu hlavy na pásce: R																							
↓																							
B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B

$$\delta: q_0; 1 \rightarrow q_1; B; R$$

Turingův stroj nahradil symbol „1“ prázdným znakem, přesunul se do stavu q_1 (pamatuje si, jaký znak musí být na konci řetězce, aby se jednalo o palindrom). Posunul se směrem doprava, jak mu pravidlo přechodové funkce nařizuje. Tak bude pokračovat i v následujících krocích (23-30). Stroj nebude nic přepisovat, pouze se potřebuje dostat na konec řetězce.

Pořadové číslo kroku: 23																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 0	Zapsaný symbol: 0																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B

$\delta: q_1; 0 \rightarrow q_1; 0; R$

Pořadové číslo kroku: 24																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 1	Zapsaný symbol: 1																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B

$\delta: q_1; 1 \rightarrow q_1; 1; R$

Pořadové číslo kroku: 25-26																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 0	Zapsaný symbol: 0																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B

$\delta: q_1; 0 \rightarrow q_1; 0; R$

Pořadové číslo kroku: 27																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 1	Zapsaný symbol: 1																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B

$\delta: q_1; 1 \rightarrow q_1; 1; R$

Pořadové číslo kroku: 28																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 0	Zapsaný symbol: 0																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B

$\delta: q_1; 0 \rightarrow q_1; 0; R$

Pořadové číslo kroku: 29-30																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																							
Čtený symbol: 1	Zapsaný symbol: 1																							
Příkaz pohybu hlavy na pásce: R																								
↓																								
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B

$\delta: q_1; 1 \rightarrow q_1; 1; R$

Pořadové číslo kroku: 31																								
Výchozí stav z Q: q_1	Výstupní stav z Q: q_2																							
Čtený symbol: B	Zapsaný symbol: B																							
Příkaz pohybu hlavy na pásce: L																								
↓																								
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	1	B	B	B	B	B	B	B	B

$\delta: q_1; B \rightarrow q_2; B; L$

Stroj se dosal nakonec řetězce. Přejde do stavu q_2 a posune se směrem doleva.

Pořadové číslo kroku: 32																											
Výchozí stav z Q: q_2	Výstupní stav z Q: q_3																										
Čtený symbol: 1	Zapsaný symbol: B																										
Příkaz pohybu hlavy na pásce: L																											
↓																											
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_2; 1 \rightarrow q_3; B; L$$

Hlava Turingova stroje čte symbol „1“, a tím byla podmínka shody s počátečním znakem splněna. Původní znak je smazán a nahrazen prázdným znakem „B“ a Turingův stroj následuje odpovídající pravidlo přechodové funkce. V následujících krocích (33-38) se pouze posouvá směrem doleva a nic nepřepisuje, dokud nenarazí na prázdný znak.

Pořadové číslo kroku: 33																											
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																										
Čtený symbol: 0	Zapsaný symbol: 0																										
Příkaz pohybu hlavy na pásce: L																											
↓																											
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_3; 0 \rightarrow q_3; 0, L$$

Pořadové číslo kroku: 34																											
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																										
Čtený symbol: 1	Zapsaný symbol: 1																										
Příkaz pohybu hlavy na pásce: L																											
↓																											
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_3; 1 \rightarrow q_3; 1, L$$

Pořadové číslo kroku: 35-36																										
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																									
Čtený symbol: 0	Zapsaný symbol: 0																									
Příkaz pohybu hlavy na pásce: L																										
↓																										
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_3; 0 \rightarrow q_3; 0, L$$

Pořadové číslo kroku: 37																										
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																									
Čtený symbol: 1	Zapsaný symbol: 1																									
Příkaz pohybu hlavy na pásce: L																										
↓																										
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_3; 1 \rightarrow q_3; 1, L$$

Pořadové číslo kroku: 38																										
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																									
Čtený symbol: 0	Zapsaný symbol: 0																									
Příkaz pohybu hlavy na pásce: L																										
↓																										
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_3; 0 \rightarrow q_3; 0, L$$

Pořadové číslo kroku: 39																										
Výchozí stav z Q: q_3	Výstupní stav z Q: q_0																									
Čtený symbol: B	Zapsaný symbol: B																									
Příkaz pohybu hlavy na pásce: R																										
↓																										
B	B	B	B	B	B	B	B	B	B	0	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_3; B \rightarrow q_0; B; R$$

Hlava Turingova stroje čte prázdný znak, až doposud jsou splněny podmínky definice palindromu, zpracované znaky byly odmazány a zbývající část řetězce lze převést na původní otázku, zda je zbývající řetězec palindrom.

Pořadové číslo kroku: 40																											
Výchozí stav z Q: q_0	Výstupní stav z Q: q_4																										
Čtený symbol: 0	Zapsaný symbol: B																										
Příkaz pohybu hlavy na pásce: R																											
↓																											
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_0; 0 \rightarrow q_4; B; R$$

Hlava Turingova stroje čte první symbol zbývajících řetězce, nahradí jej prázdným znakem a „poznamená si“ jeho hodnotu pomocí stavu, do kterého se přesune (q_4).

V následujících krocích (41-45) se bude hlava Turingova stroje posouvat směrem doprava, aniž by cokoli přepisovala, dokud nedosáhne konce řetězce reprezentovaného prázdným znakem „B“.

Pořadové číslo kroku: 41																											
Výchozí stav z Q: q_4	Výstupní stav z Q: q_4																										
Čtený symbol: 1	Zapsaný symbol: 1																										
Příkaz pohybu hlavy na pásce: R																											
↓																											
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_4; 1 \rightarrow q_4; 1; R$$

Pořadové číslo kroku: 42-43																											
Výchozí stav z Q: q_4	Výstupní stav z Q: q_4																										
Čtený symbol: 0	Zapsaný symbol: 0																										
Příkaz pohybu hlavy na pásce: R																											
↓																											
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_4; 0 \rightarrow q_4; 0; R$$

Pořadové číslo kroku: 44																													
Výchozí stav z Q: q_{14}	Výstupní stav z Q: q_4																												
Čtený symbol: 1	Zapsaný symbol: 1																												
Příkaz pohybu hlavy na pásce: R																													
↓																													
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_4; 1 \rightarrow q_4; 1; R$$

Pořadové číslo kroku: 45																													
Výchozí stav z Q: q_4	Výstupní stav z Q: q_4																												
Čtený symbol: 0	Zapsaný symbol: 0																												
Příkaz pohybu hlavy na pásce: R																													
↓																													
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_4; 0 \rightarrow q_4; 0; R$$

Pořadové číslo kroku: 46																													
Výchozí stav z Q: q_4	Výstupní stav z Q: q_5																												
Čtený symbol: B	Zapsaný symbol: B																												
Příkaz pohybu hlavy na pásce: L																													
↓																													
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	0	B	B	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_4; B \rightarrow q_5; B; L$$

Byl dosažen konec řetězce a je nutné přesunout se na poslední neprázdný symbol. Přesune se proto do stavu q_5 , nic nepřepisuje a posune se doleva.

Pořadové číslo kroku: 47																												
Výchozí stav z Q: q_5	Výstupní stav z Q: q_6																											
Čtený symbol: 0	Zapsaný symbol: B																											
Příkaz pohybu hlavy na pásce: L																												
↓																												
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	B	B	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_5; 0 \rightarrow q_6; B; L$$

Byl přečten symbol „0“, který odpovídá počátečnímu znaku. Výpočet může pokračovat, jelikož existuje další přechodové pravidlo.

V následujících krocích (48-51) se Turingův stroj potřebuje dostat znovu na začátek řetězce. Bude se pohybovat doleva a nic nebude přepisovat.

Pořadové číslo kroku: 48																												
Výchozí stav z Q: q_6	Výstupní stav z Q: q_6																											
Čtený symbol: 1	Zapsaný symbol: 1																											
Příkaz pohybu hlavy na pásce: L																												
↓																												
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	B	B	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_6; 1 \rightarrow q_6; 1, L$$

Pořadové číslo kroku: 49-50																												
Výchozí stav z Q: q_6	Výstupní stav z Q: q_6																											
Čtený symbol: 0	Zapsaný symbol: 0																											
Příkaz pohybu hlavy na pásce: L																												
↓																												
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	B	B	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_6; 0 \rightarrow q_6; 0, L$$

Pořadové číslo kroku: 51																												
Výchozí stav z Q: q_6	Výstupní stav z Q: q_6																											
Čtený symbol: 1	Zapsaný symbol: 1																											
Příkaz pohybu hlavy na pásce: L																												
↓																												
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	B	B	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_6; 1 \rightarrow q_6; 1, L$$

Pořadové číslo kroku: 52																												
Výchozí stav z Q: q_6	Výstupní stav z Q: q_0																											
Čtený symbol: B	Zapsaný symbol: B																											
Příkaz pohybu hlavy na pásce: R																												
↓																												
B	B	B	B	B	B	B	B	B	B	B	1	0	0	1	B	B	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_6; B \rightarrow q_0; B; R$$

Byl dosažen prázdný znak před počátkem řetězce. Stroj opět přejde do počátečního stavu a posune se směrem doprava.

Pořadové číslo kroku: 53																												
Výchozí stav z Q: q_0	Výstupní stav z Q: q_1																											
Čtený symbol: 1	Zapsaný symbol: B																											
Příkaz pohybu hlavy na pásce: R																												
↓																												
B	B	B	B	B	B	B	B	B	B	B	B	0	0	1	B	B	B	B	B	B	B	B	B	B	B	B	B	B

$$\delta: q_0; 1 \rightarrow q_1; B; R$$

Hlava Turingova stroje přepíše počáteční symbol prázdným znakem „B“, přejde do stavu q_1 , jelikož čteným znakem byl symbol „1“.

V následných krocích (54-56) se bude hlava TS pohybovat směrem doprava a nebude nic přepisovat.

Pořadové číslo kroku: 54-55																										
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																									
Čtený symbol: 0	Zapsaný symbol: 0																									
Příkaz pohybu hlavy na pásce: R																										
↓																										
B	B	B	B	B	B	B	B	B	B	B	B	0	0	1	B	B	B	B	B	B	B	B	B	B	B	B

$\delta: q_1; 0 \rightarrow q_1; 0, R$

Pořadové číslo kroku: 56																										
Výchozí stav z Q: q_1	Výstupní stav z Q: q_1																									
Čtený symbol: 1	Zapsaný symbol: 1																									
Příkaz pohybu hlavy na pásce: R																										
↓																										
B	B	B	B	B	B	B	B	B	B	B	B	0	0	1	B	B	B	B	B	B	B	B	B	B	B	B

$\delta: q_1; 1 \rightarrow q_1; 1; R$

Pořadové číslo kroku: 57																										
Výchozí stav z Q: q_1	Výstupní stav z Q: q_2																									
Čtený symbol: B	Zapsaný symbol: B																									
Příkaz pohybu hlavy na pásce: L																										
↓																										
B	B	B	B	B	B	B	B	B	B	B	B	0	0	1	B	B	B	B	B	B	B	B	B	B	B	B

$\delta: q_1; B \rightarrow q_2; B; L$

Byl dosažen prázdný znak „B“. Hlava se posouvá o jedno pole doleva a stroj přechází do stavu q_2 .

Pořadové číslo kroku: 58																											
Výchozí stav z Q: q_2	Výstupní stav z Q: q_3																										
Čtený symbol: 1	Zapsaný symbol: B																										
Příkaz pohybu hlavy na pásce: L																											
↓																											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td> </tr> </table>		B	B	B	B	B	B	B	B	B	B	B	0	0	B	B	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	0	0	B	B	B	B	B	B	B	B	B	B	B	B	B		

$\delta: q_2; 1 \rightarrow q_3; B; L$

Posledním symbolem řetězce je „1“, podmínky byly opět splněny. Hlava Turingova stroje symbol přepíše prázdným znakem „B“.

V následujících krocích (59-60) se bude opět posouvat doleva, dokud nenačte prázdný znak „B“.

Pořadové číslo kroku: 59-60																											
Výchozí stav z Q: q_3	Výstupní stav z Q: q_3																										
Čtený symbol: 0	Zapsaný symbol: 0																										
Příkaz pohybu hlavy na pásce: L																											
↓																											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td> </tr> </table>		B	B	B	B	B	B	B	B	B	B	B	0	0	B	B	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	0	0	B	B	B	B	B	B	B	B	B	B	B	B	B		

$\delta: q_3; 0 \rightarrow q_3; 0; L$

Pořadové číslo kroku: 61																											
Výchozí stav z Q: q_3	Výstupní stav z Q: q_0																										
Čtený symbol: B	Zapsaný symbol: B																										
Příkaz pohybu hlavy na pásce: R																											
↓																											
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td><td style="padding: 2px;">B</td> </tr> </table>		B	B	B	B	B	B	B	B	B	B	B	0	0	B	B	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	0	0	B	B	B	B	B	B	B	B	B	B	B	B	B		

$\delta: q_3; B \rightarrow q_0; B; R$

Nyní se stroj dostal, z jeho pohledu, do standardní situace, kdy je na začátku řetězce. My pozorujeme, že mu zbývají pouze dvě nuly, ale stroj tento pohled nemá k dispozici, a musí proto první symbol nahradit prázdným znakem a vyhledat odpovídající symbol na konci řetězce.

Pořadové číslo kroku: 62																											
Výchozí stav z Q: q_0	Výstupní stav z Q: q_4																										
Čtený symbol: 0	Zapsaný symbol: B																										
Příkaz pohybu hlavy na pásce: R																											
↓																											
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>0</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td> </tr> </table>		B	B	B	B	B	B	B	B	B	B	B	B	0	B	B	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	B	0	B	B	B	B	B	B	B	B	B	B	B	B	B		

$$\delta: q_0; 0 \rightarrow q_4; 0; R$$

Pořadové číslo kroku: 63																											
Výchozí stav z Q: q_4	Výstupní stav z Q: q_4																										
Čtený symbol: 0	Zapsaný symbol: 0																										
Příkaz pohybu hlavy na pásce: R																											
↓																											
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>0</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td> </tr> </table>		B	B	B	B	B	B	B	B	B	B	B	B	0	B	B	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	B	0	B	B	B	B	B	B	B	B	B	B	B	B	B		

$$\delta: q_4; 0 \rightarrow q_4; 0; R$$

Pořadové číslo kroku: 64																											
Výchozí stav z Q: q_4	Výstupní stav z Q: q_5																										
Čtený symbol: B	Zapsaný symbol: B																										
Příkaz pohybu hlavy na pásce: L																											
↓																											
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>0</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td> </tr> </table>		B	B	B	B	B	B	B	B	B	B	B	B	0	B	B	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	B	0	B	B	B	B	B	B	B	B	B	B	B	B	B		

$$\delta: q_4; B \rightarrow q_5; B; L$$

Byl dosažen prázdný znak za koncem řetězce a hlava Turingova stroje se posune na poslední znak řetězce.

Pořadové číslo kroku: 65																															
Výchozí stav z Q: q_5	Výstupní stav z Q: q_6																														
Čtený symbol: 0	Zapsaný symbol: B																														
Příkaz pohybu hlavy na pásce: L																															
↓																															
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td> </tr> </table>		B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B		

$$\delta: q_5; 0 \rightarrow q_6; B; L$$

Hlava Turingova stroje přepíše poslední znak řetězce prázdným znakem, následuje pravidlo přechodové funkce a posouvá se doleva.

Pořadové číslo kroku: 66																																	
Výchozí stav z Q: q_6	Výstupní stav z Q: q_0																																
Čtený symbol: B	Zapsaný symbol: B																																
Příkaz pohybu hlavy na pásce: R																																	
↓																																	
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td> </tr> </table>		B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B		

$$\delta: q_6; B \rightarrow q_0; B; R$$

Je dosažen počátek řetězce. Tento řetězec je zajímavý tím, že je prázdný, je tedy diskutabilní, zda se jedná o řetězec, ale rozhodně se jedná o palindrom.

Pořadové číslo kroku: 67																																
Výchozí stav z Q: q_0	Výstupní stav z Q: q_7																															
Čtený symbol: B	Zapsaný symbol: B																															
Příkaz pohybu hlavy na pásce: N																																
↓																																
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td><td>B</td> </tr> </table>		B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B		

$$\delta: q_0; B \rightarrow q_7; B; R$$

Stroj je ve stavu q_0 , čte prázdný znak a odpovídající pravidlo přikazuje přejít do stavu q_7 . Stav q_7 je prvkem množiny koncových stavů, výpočet končí a stav q_7 symbolizuje výsledek. Stroj tedy přijal řetězec koncovým stavem a je tedy dokázáno, že zadaný řetězec byl sudý palindrom.

6.2 Sudý i lichý palindrom

Samozřejmě je možné navrhnout i univerzálnější algoritmus, který by poznal jak sudé, tak liché palindromy a určil by, o který se jedná.

Můžeme použít stejnou vstupní i páskovou abecedu jako v minulém případě, avšak množinu koncových stavů si rozšíříme na dva. Pokud se tedy Turingův stroj zastaví v prvním koncovém stavu (q_8), bude se jednat o sudý palindrom. Pokud ovšem po provedení poslední iterace skončí v druhém (q_9), jedná se o lichý palindrom.

Tabulka 4 - Přejchodová funkce č. 4

stav	0	1	B
q_0	$q_0; 0 \rightarrow q_2; B; R$	$q_0; 1 \rightarrow q_1; B; R$	$q_0; B \rightarrow q_8; B; N$
q_1	$q_1; 0 \rightarrow q_3; 0; R$	$q_1; 1 \rightarrow q_3; 1; R$	$q_1; B \rightarrow q_9; B; N$
q_2	$q_2; 0 \rightarrow q_4; 0; R$	$q_2; 1 \rightarrow q_4; 1; R$	$q_2; B \rightarrow q_9; B; N$
q_3	$q_3; 0 \rightarrow q_3; 0; R$	$q_3; 1 \rightarrow q_3; 1; R$	$q_3; B \rightarrow q_5; B; L$
q_4	$q_4; 0 \rightarrow q_4; 0; R$	$q_4; 1 \rightarrow q_4; 1; R$	$q_4; B \rightarrow q_6; B; L$
q_5		$q_5; 1 \rightarrow q_7; B; L$	
q_6	$q_6; 0 \rightarrow q_7; B; L$		
q_7	$q_7; 0 \rightarrow q_7; 0; L$	$q_7; 1 \rightarrow q_7; 1; L$	$q_7; B \rightarrow q_0; B; R$
q_8	Přijato – sudý palindrom		
q_9	Přijato – lichý palindrom		

6.3 Nepřijetí řetězce

Pokud se Turingův stroj zastaví v jiném než koncovém stavu, znamená to, že se jedná o jiný typ palindromu, nebo se vůbec nejedná o palindrom.

Zadání: Rozhodněte, zda je zadaný řetězec sudý palindrom, když je na pásce řetězec „B 1 1 0 1 0 0 1 0 1 0 B“

Tabulka 5 – Přejchodová funkce č. 2

stav	0	1	B
q ₀	q ₀ ; 0 → q ₄ ; B; R	q ₀ ; 1 → q ₁ ; B; R	q ₀ ; B → q ₇ ; B; N
q ₁	q ₁ ; 0 → q ₁ ; 0; R	q ₁ ; 1 → q ₁ ; 1; R	q ₁ ; B → q ₂ ; B; L
q ₂		q ₂ ; 1 → q ₃ ; B; L	
q ₃	q ₃ ; 0 → q ₃ ; 0; L	q ₃ ; 1 → q ₃ ; 1; L	q ₃ ; B → q ₀ ; B; R
q ₄	q ₄ ; 0 → q ₄ ; 0; R	q ₄ ; 1 → q ₄ ; 1; R	q ₄ ; B → q ₅ ; B; L
q ₅	q ₅ ; 0 → q ₆ ; B; L		
q ₆	q ₆ ; 0 → q ₆ ; 0; L	q ₆ ; 1 → q ₆ ; 1; L	q ₆ ; B → q ₀ ; B; R
q ₇	Přijato		

Použijeme stejný algoritmus jako v prvním případě. Výpočet tedy bude vypadat stejně jako prvně, avšak k rozdílu dojde v kroku číslo 12, kde jsme si řekli, že dalším znakem musí být „1“. Tentokrát ovšem hlava na konci řetězce čte jiný symbol.

Pořadové číslo kroku: 12																							
Výchozí stav z Q: q ₂	Výstupní stav z Q: není definován																						
Čtený symbol: 0	Zapsaný symbol: není definován																						
Příkaz pohybu hlavy na pásce: není definován																							
↓																							
B	B	B	B	B	B	B	B	1	0	1	0	0	1	0	1	0	B	B	B	B	B	B	B

Pro takový případ není ve stavu q₂ definována žádná přechodová funkce, stroj se tedy zastaví v posledním vstupním stavu, tedy v q₂, který není koncovým stavem. Nejedná se tedy o palindrom.

6.4 Zacyklení

Jak už bylo řečeno, může nastat situace, kdy se Turingův stroj nikdy nezastaví. V takovémto případě se jedná o chybu v přechodové funkci. Pro demonstraci poslouží následující příklad.

Tabulka 6 – Přechodová funkce č. 3

stav	0	1	B
q_0	$q_0; 0 \rightarrow q_2; B; R$	$q_0; 1 \rightarrow q_1; B; R$	
q_1	$q_1; 0 \rightarrow q_1; 0; R$	$q_1; 1 \rightarrow q_2; 1; R$	$q_1; B \rightarrow q_1; B; L$

Přechodová funkce obsahuje (pro zjednodušení) zcela zřejmou chybu. V případě, že Turingův stroj čte na vstupu „0“, tak se posune doprava a „0“ ponechá beze změny. Vzhledem k tomu, že nezměnil svůj stav, stačí, aby za „0“ následoval prázdný znak „B“ a ihned s další iterací se vrátí zpět - zacyklí se. Výpočet tedy nikdy neskončí.

Diskuse

V praktické části bylo provedeno několik pokusů výpočtu na Turingově stroji. Pro názornost byly vybrány jednoduché příklady výpočtů. Během zpracování práce byly provedeny další pokusy a byla prozkoumána funkčnost Turingova stroje.

Během zpracovávání teoretické části bylo zjištěno, že informatika (mylně považována čistě za technický obor) je ve skutečnosti především matematickou disciplínou, která zastřešuje i další obory. Teprve tyto podoby informatiky mají znaky přesahu technických znalostí, jako je programování, měření, optimalizace a dalších.

Praktická část se ukázala jako klíčová k pochopení principu, na kterém Turingovy stroje pracují. Byl zvolen jiný způsob zápisu jednotlivých kroků, který se v dalších publikacích zabývajících se stejnou problematikou nepoužívá.

Tento způsob zápisu byl zvolen pro jasnost a podrobnost. Je možné lépe postřehnout jednotlivé změny ve výpočtu oproti předchozím krokům, dále umožňuje přehlednější vracení se k předchozím krokům, v případě potřeby zaměření se na konkrétní části výpočtu.

Pochopení Turingova stroje je nejjednodušší cesta k pochopení všech současných počítačů. V práci je doloženo, že Turingův stroj je stejně výkonný jako všechny počítače, jaké máme všude okolo sebe.

Práce nepokryla všechny přínosy Alana Turinga, jelikož pochopení jeho dalších přínosů vyžaduje mnohem více znalostí v oboru, kterých autor zatím neměl jak dosáhnout.

Práci by ovšem bylo možno zásadně rozšířit a prohloubit. Prostor k dalšímu rozvoji je nyní již zřejmý jak v teoretické, tak v praktické části.

Teoretická část by mohla problematiku rozvinout do hloubky i šířky. Jen úvod do teoretické informatiky však zásadně překračuje rozsah této práce. Během zpracování byl pro zjednodušení zcela vynechán nedeterminismus a s ním spojené oblasti teoretické informatiky. Již během přípravy této práce bylo zřejmé, že veškerá problematika v teoretické části práce bude muset být omezena jen na nezbytně nutný rozsah, aby bylo předvedení výpočtu na Turingově stroji v rámci práce vůbec dosaženo. Práci je tedy

možno rozšířit kterýmkoli směrem v rámci teoretické informatiky, ať už by se jednalo o teorii grafů, algoritmy, složitost, teorii jazyků, gramatik i kompilátorů, nebo by mohla mířit směrem k dalším přínosům Alana Turinga a jejich podrobnějšímu nastudování.

Praktická část by mohla být dále rozvinuta směrem ke složitějším výpočtům i směrem k vícepáskovým variantám Turingova stroje.

Závěr

V práci byla zkoumána činnost Alana Turinga a jeho přínos pro informatiku. Během studia a přípravy práce bylo zjištěno, že není ani možno pochopit, natož docenit jeho přínos bez alespoň základních znalostí teoretické informatiky, ve které Alan Turing vstoupil do dějin. Práce se ale snaží tuto zeď překročit a umožnit čtenáři proniknout do vod teoretické informatiky.

Po prostudování a seznámení se s nezbytnou terminologií se práce důkladněji zaměřila na Turingův stroj, kterému je věnována celá praktická část.

V praktické části bylo úspěšně provedeno několik výpočtů na Turingově stroji. Podrobné rozepsání jednotlivých kroků pomohlo k lepšímu pochopení způsobu fungování. Dále byly nasimulovány situace, které nevedly k výsledku. Díky těmto ukázkám je možné začít chápat nejjednodušší model počítače. Hned po pochopení tohoto základu se čtenáři otevírá brána do světa všech moderních výpočetních strojů. Jak již bylo řečeno, žádný současný počítač nedovede nic víc, než by zvládl Turingův stroj. Vždyť je to právě Turingovská úplnost, čím je definována kategorie úplných programovacích jazyků.

Bylo zjištěno, že v oborech teoretické informatiky prakticky není možné narazit na odvětví, ve kterém by Turing nezanechal svou stopu. Práce se snaží související problematiku uchopit a zprostředkovává ji ve zjednodušené, ale snadno pochopitelné podobě. Největším přínosem této práce je tedy částečné prolomení bariéry mezi selským rozumem a teoretickou informatikou, která se v běžných publikacích již na úvodních stranách stává „nestravitelnou“.

Prameny a literatura

1. Alan Turing. *FDb.cz* [online]. HyperMedia, 2022 [cit. 2022-01-19]. Dostupné z: <https://www.fdb.cz/lidi-zivotopis-biografie/75390-alan-turing.html>
2. Alan Turing - betrayed by the country he saved. *YouTube* [online]. 2019 [cit. 2022-01-19]. Dostupné z: <https://www.youtube.com/watch?v=ynTAFPukXBk>
3. Alan Turing. *Osobnosti.cz* [online]. Tiscali Media, 1996–2022 [cit. 2022-01-19]. Dostupné z: <https://zivotopis.osobnosti.cz/alan-turing.php>
4. PŘIKRYL, Jan. Analýza algoritmů. Složitost. *Zolotarev.fd.cvut.cz* [online]. Praha, 2013 [cit. 2022-01-19]. Dostupné z: <https://zolotarev.fd.cvut.cz/static/mag/mag-2014-02-slides.pdf>
5. Alan Turing. *Wikipedia languages* [online]. [cit. 2022-01-19]. Dostupné z: https://www.bloges.wiki/wiki/cs/Alan_Turing
6. Alan Mathison Turing (1912-54). *King's College* [online]. [cit. 2022-01-19]. Dostupné z: <https://www.kings.cam.ac.uk/archive-centre/online-resources/online-exhibitions/alan-mathison-turing-1912-54>
7. The Alan Turing Programme at King's. *King's College* [online]. [cit. 2022-01-19]. Dostupné z: <https://www.kings.cam.ac.uk/members-and-friends/support/giving/empowering-minds/the-alan-turing-programme>
8. Alonzo Church. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-01-19]. Dostupné z: https://cs.wikipedia.org/wiki/Alonzo_Church
9. Geniální Enigma: Zašifrovaná historie. *Abc* [online]. 2021 [cit. 2022-01-19]. Dostupné z: <https://www.abicko.cz/clanek/precti-si-technika/27117/genialni-enigma-zasifrovana-historie.html>
10. *Filosofické aspekty Turingova pojetí inteligence* [online]. Liberec, 2017 [cit. 2022-01-19]. Dostupné z: https://dspace.tul.cz/bitstream/handle/15240/60733/V_16017_Pb.pdf?sequence=4&isAllowed=y. Bakalářská práce. Technická univerzita v Liberci.
11. *Turingův test* [online]. Olomouc, 2011 [cit. 2022-01-19]. Dostupné z: https://theses.cz/id/st6pmz/Turingv_test.pdf. Disertační práce. Univerzita Palackého v Olomouci.
12. *Složitost problémů* [online]. Ostrava, 2012 [cit. 2022-01-19]. Dostupné z: <http://www.cs.vsb.cz/kot/down/uti2012/uti-12-cz.pdf>
13. Typy problémů a rozhodnutí (Types of Problems and Decisions). *ManagementMania* [online]. 2016 [cit. 2022-01-19]. Dostupné z: <https://managementmania.com/cs/typy-problemu-a-rozhodnuti>
14. SIPSER, Michael. *Introduction to the theory of computation*. 2nd ed. Boston: Thomson, 2006. ISBN 0-534-95097-3.
15. *Vyčísitelnost a složitost 1* [online]. Ostrava, 2002 [cit. 2022-01-19]. Dostupné z: <http://physics.ujep.cz/~jskvor/ZS13/AFJ/xvysl1.pdf>. UČEBNÍ TEXTY OSTRAVSKÉ UNIVERZITY. Ostravská univerzita.

16. Spočetná množina. *Wikipedie: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2022-01-19]. Dostupné z: https://cs.wikipedia.org/wiki/Spo%C4%8Detn%C3%A1_mno%C5%BEina
17. Nespočetná množina. *Wikipedie: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2022-01-19]. Dostupné z: https://cs.wikipedia.org/wiki/Nespo%C4%8Detn%C3%A1_mno%C5%BEina
18. Co je to algoritmus. *Učebnice* [online]. [cit. 2022-01-19]. Dostupné z: https://popelka.ms.mff.cuni.cz/~lessner/mw/index.php/U%C4%8Debnice/Algoritmus/Co_je_to_algoritmus
19. *Složitost algoritmů* [online]. Praha, 2018 [cit. 2022-01-19]. Dostupné z: https://cw.fel.cvut.cz/old/_media/courses/b6b36dsa/dsa-3-slozitostalgoritmu.pdf. České vysoké učení technické v Praze.
20. *Základy složitosti a vyčíslitelnosti* [online]. 2018/19 [cit. 2022-01-19]. Dostupné z: <http://ktiml.mff.cuni.cz/~kucerap/NTIN090/NTIN090-sljady.pdf>
21. *Vybrané třídy složitosti* [online]. Brno, 2005 [cit. 2022-01-19]. Dostupné z: http://www.fit.vutbr.cz/~meduna/mti/2004_05/erlebach.pdf. Vysoké učení technické v Brně.
22. *Převoditelnost problémů nezávislé množiny na problém hamiltonovského cyklu* [online]. [cit. 2022-01-19]. Dostupné z: http://www.cs.vsb.cz/kot/soubory_animaci/Hrancik/IS-HC.pdf
23. *Časová složitost algoritmů* [online]. Šumperk, 2009 [cit. 2022-01-19]. Dostupné z: https://cw.fel.cvut.cz/old/_media/courses/y36alg/2009_sumperk_p10.pdf
24. *Automaty a gramatiky* [online]. Praha [cit. 2022-01-19]. Dostupné z: <http://ktiml.mff.cuni.cz/~bartak/automaty/lectures/lecture12.pdf>. Univerzita Karlova, Matematicko-fyzikální fakulta.
25. *Vyčíslitelnost a složitost* [online]. Brno [cit. 2022-01-19]. Dostupné z: <https://akela.mendelu.cz/~foltynek/VAS/OporaCZ/content/index.html#cast70982https://dewiki.top/wiki/Turingmaschine>. Mendelova univerzita v Brně.
26. Úvod do teoretické informatiky - přednáška 8: Turingovy stroje, Chomského hierarchie. *YouTube* [online]. 2021 [cit. 2022-01-19]. Dostupné z: https://www.youtube.com/watch?v=-Dag_tCfoU
27. Klasifikace gramatik, formálních jazyků a automatů přijímajících jazyky. *Kalábovi* [online]. 2012 [cit. 2022-01-19]. Dostupné z: <https://kalabovi.org/pitel:msz:gramatiky>
28. *Různé definice Turingova stroje* [online]. Praha, 2010 [cit. 2022-01-19]. Dostupné z: https://dspace.cuni.cz/bitstream/handle/20.500.11956/28499/BPTX_2009_1_11320_NSZZ027_236324_0_77984.pdf?sequence=1&isAllowed=y. Bakalářská práce. Univerzita Karlova, Matematicko-fyzikální fakulta.
29. Automat. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2022-01-19]. Dostupné z: <https://cs.wikipedia.org/wiki/Automat>
30. Here is the simplest explanation. *Stackoverflow* [online]. [cit. 2022-01-19]. Dostupné z: <https://stackoverflow.com/questions/7284/what-is-turing-complete/37247136#37247136>

31. Turing completeness. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2022-01-19]. Dostupné z: https://en.wikipedia.org/wiki/Turing_completeness
32. Church-Turingova teze. *Referaty-seminarky.cz* [online]. 2008 [cit. 2022-01-19]. Dostupné z: <http://referaty-seminarky.cz/church-turingova-teze/>
33. *Automaty a gramatiky* [online]. Praha [cit. 2022-01-19]. Dostupné z: http://kti.mff.cuni.cz/~bartak/automaty/lectures/lecture12_1x2.pdf. Univerzita Karlova, Matematicko-fyzikální fakulta.
34. *Vypočitatelnost a výpočtová složitost* [online]. [cit. 2022-01-19]. Dostupné z: <https://www.kiv.zcu.cz/~netrvalo/vyuka/ppa2-13/ekniha/online/HTML/51/default.htm>
35. Co je to nenaprogramovatelné?. *ROOT.cz* [online]. 1998 – 2022, 26. 11. 2003 [cit. 2022-01-19]. Dostupné z: <https://www.root.cz/clanky/co-je-to-nenaprogramovatelne/>
36. *Halting problem* [online]. [cit. 2022-01-19]. Dostupné z: https://p2k.unkris.ac.id/IT/en/3065-2962/halting-set_9368_p2k-unkris.html
37. RAM stroj. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2022-01-19]. Dostupné z: https://cs.wikipedia.org/wiki/RAM_stroj
38. Lambda kalkul. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2022-01-19]. Dostupné z: https://cs.wikipedia.org/wiki/RAM_stroj
39. *Kapitola 2. Lambda kalkul* [online]. [cit. 2022-01-19]. Dostupné z: <http://www.cs.vsb.cz/navrat/vyuka/flp/texty/fp/ch02.html>
40. Post–Turing machine. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2022-01-19]. Dostupné z: https://en.wikipedia.org/wiki/Post%E2%80%93Turing_machine#cite_note-1
41. Turing Machine for Palindrome. *YouTube* [online]. 12. 6. 2017 [cit. 2022-01-19]. Dostupné z: <https://www.youtube.com/watch?v=Ynd5on9g6Q8>

Zdroje obrázků

1. Alan Turing: Vynálezce Turingova stroje. *Elektrina.cz* [online]. [cit. 2022-01-19]. Dostupné z: https://www.elektrina.cz/data/images/article/616-wikimedia-nutno-uvest-autora-rocky-acosta-Turing_Machine_Model_Davey_2012.jpg

Seznam obrázků

Obrázek 1 – Turingův stroj	25
----------------------------------	----

Seznam tabulek

Tabulka 1 - Třídy asymptotické složitosti.....	19
Tabulka 2 - Srovnávací tabulka jazyků, gramatik a strojů dle Chomského	24
Tabulka 3 – Přechodová funkce č. 1	36
Tabulka 4 - Přechodová funkce č. 4.....	56
Tabulka 5 – Přechdová funkce č. 2	57
Tabulka 6 – Přechodová funkce č. 3	58