FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

| | |
|---|---|
| **Název:** | Cantor, Gödel, Turing a paradox lháře |
| **Student:** | Martin Slávik |
| **Vedoucí:** | RNDr. Kateřina Trlifajová, Ph.D. |
| **Studijní program:** | Informatika |
| **Studijní obor:** | Teoretická informatika |
| **Katedra:** | Katedra teoretické informatiky |
| **Platnost zadání:** | Do konce letního semestru 2020/21 |

## Pokyny pro vypracování

Cantorova věta o nespočetnosti reálných čísel, Gödelova věta o neúplnosti a problém zastavení Turingova stroje jsou tři negativní matematické věty, všechny jsou postaveny na diagonalizaci. V jejich základě je princip sebevztažnosti, jehož prvním a nejjednodušším vyjádřením je paradox lháře. Bylo by zajímavé zjistit, jaká mezi těmito třemi důkazy existuje souvislost.

Student se seznámí s Cantorovou větou a jejími důsledky v matematice, Gödelovou větou a problémem zastavení Turingova stroje a jeho souvislostmi. Srozumitelně vše popíše a bude se zabývat otázkou, v čem se jednotlivé důkazy liší a v čem si jsou podobné. Použije uvedenou literaturu a vyhledá si i další materiály.

## Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Jan Janoušek, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 1. února 2020

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

# Cantor, Gödel, Turing a paradox lháře

***Martin Slávik***

Katedra teoretické informatiky
Vedoucí práce: RNDr. Kateřina Trlifajová, Ph.D.

3. června 2020

# Poděkování

Rád bych poděkoval vedoucí RNDr. Kateřině Trlifajové, Ph.D. za trpělivost, vstřícnost a cenné rady při tvorbě této práce.

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principu při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisu, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 3. června 2020 . . . . . .

### Odkaz na tuto práci

Slávik, Martin. *Cantor, Gödel, Turing a paradox lháře*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

# Abstrakt

Tato bakalářská práce se zabývá vysvětlením jednotlivých problémů, zasazením do souvislostí a rozborem důkazů. Cantorova věta, Gödelova věta a problémem zastavení Turingova stroje jsou tři důležitá negativní tvrzení z různých oborů matematiky a informatiky. Spojuje je fakt, že se v jejich důkazech nějakým, avšak rozdílným způsobem využívá metoda diagonalizace. Ve všech případech lze jádro důkazu schematicky ukázat na nekonečné čtvercové tabulce. Pomocí „negace diagonály“ se vytvoří nový prvek, jenž v tabulce nemůže být obsažen a z jehož existence vyplývá negativní výsledek těchto tří vět. Způsob vytvoření tabulky a formální popis „negace diagonály“ je však odlišný. Jednotlivé důkazy nelze přímočaře mezi sebou převádět. Avšak přes Richardův paradox lze ukázat, jak se z Cantorovy věty odvodí Gödelova věta s použitím paradoxu lháře. Ukazuje také podobnost mezi universálním Turingovým strojem a Gödelovou formulí, která je založena na stejném principu jako je paradox lháře.

**Klíčová slova** Diagonální lemma, Diagonální metoda, Paradox lháře, Gödelova věta, Cantorova věta, Cantorův paradox, Turingův stroj, Problém zastavení

# Abstract

This bachelor's thesis deals with the explanation of individual problems, setting them in context and analyzing how to prove them. Cantor's theorem, Gödel's theorem, and the Halting problem are three important negative statements from various fields of mathematics and computer science. They are united by the fact that the diagonalization method is used in their proof, even though in different ways. In all cases, the core of the proof can be shown schematically on an infinite square table. The "diagonal negation" creates a new element which cannot be included in the table and whose existence results in a negative result of these three theorems. However, the way the table is created and the formal descriptions of the "diagonal negation" are different. The individual pieces of evidence cannot be transferred directly into each other. However, with Richard's paradox, it is possible to show how Gödel's theorem is derived from Cantor's theorem using the Liar's paradox. It also shows the similarity between the universal Turing machine and the Gödel's formula, which is based on the same essence as the Liar's paradox.

**Keywords** Diagonal lemma, Diagonal method, Liar's Paradox, Gödel's theorem, Cantor's theorem, Cantor's paradox, Turing Machine, Halting problem

# Obsah

# Seznam tabulek

# Úvod

V této práci se budeme zabývat třemi tématy ze tří různých oborů matematiky a informatiky. Všechna tři tvrzení přinášejí negativní výsledky, které ovlivnily následující vývoj daných oborů. Opírají se o dva principy, které se navzájem ovlivňují: sebevztažnost a diagonalizaci.

Vzniká tedy otázka, zda princip sebevztažnosti a použití diagonální metody v těchto třech tvrzeních je pouze náhoda, anebo zda tyto tři problémy a jejich řešení spolu nějakým způsobem souvisí?

Budeme se zabývat následujícími tvrzeními:

**Cantorova věta** je důležitou větou v terorii množin. Ukazuje, že mohutnost potenční množiny je vždy větší než mohutnost původní množiny. Její důkaz je formálně jednoduchý. Dále zjistíme, jak funguje *Cantorův diagonální argument*, který vyplývá z Cantorovy věty, resp. Cantorova věta je tedy jeho zobecněním. Byl použit pro důkaz nespočetnosti reálných čísel. Dále nás, jako první, seznámí s diagonálou.

**Gödelova věta** je věta z matematické logiky. Tato kapitola bude o něco obsáhlejší. Ukážeme, jak se tato věta dokazuje. Také si něco povíme o známém Richardově paradoxu. Samotná Gödelova věta byla dokázána v roce 1931 a hned poté se proslavila svým výsadním postavením v celé moderní matematické logice. Svou roli hraje v celé matematice, zejména v aritmetice a v teorii množin. Gödelovi se s její pomocí podařilo dokázat, že nelze „sesbírat“ všechny axiomy, aby axiomatická teorie byla kompletní, neboli úplná.

**Problém zastavení** Turingova stroje bude posledním tématem. V jeho důkazu se dá také použít Cantorův diagonální argument, proto budeme zkoumat, jak tento problém z teorie vyčíslitelnosti souvisí s tím z teorie množin (Cantorova věta) a s druhým (Gödelova věta) z oboru logiky. Problém zastavení, jak je patrné z jeho názvu, řeší, zda je možné obecně rozhodnout, zda pro libovolný počítačový program (také algoritmus) existuje Turingův stroj, který po jeho přijetí zastaví, tedy ukončí výpočet s nějakým výsledkem. Nebo naopak bude v nekonečném cyklu a tedy stále počítat. Odpověď na tento problém

našel Alan Turing v roce 1936.

**Shrnutí** a porovnání proběhne v samostatné kapitole, v níž se budeme zabývat otázkou podobnosti daných problémů a jejich důkazů.

## Cíl práce

Cílem práce je vytvořit matematický dokument, který popisuje (resp. objasňuje) následující věty a jejich důkazy:

**Cantorova věta** a její důsledky v matematice

**Gödelova věta** a její důsledky v matematické logice

**Problém zastavení Turingova stroje** a jeho vliv na informatiku.

Všechna tato tvrzení jsou negativními matematickými tvrzeními. Celá práce má za úkol zkoumat tyto slavné problémy. Poté bude hledat podobnosti a rozdíly, ve kterých se od sebe liší. Stěžejní oblastí bude Kapitola o Gödelově větě, jež je zpravidla zahrnována mezi nejtěžší oblasti disciplíny zvané *logika*.

Kapitola 1

# Cantorova věta

S porovnáváním velikostí nekonečných souborů byl už odedávna problém a to je jedním z důvodů, proč až do konce 19. století nebylo aktuální nekonečno[1] přijato do matematiky. Už Galileo Galilei jej demonstroval s použitím přirozených čísel $\mathbb{N}$ a jejich druhých mocnin $Q$, když srovnával velikost těchto dvou souborů.

Existují dva pohledy.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1. | $\mathbb{N}$ | 1, | 2, | 3, | 4, | 5, | ... |
| | $Q$ | 1, | | | 4, | | ... |

---

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2. | $\mathbb{N}$ | 1, | 2, | 3, | 4, | 5, | ... |
| | $Q$ | 1, | 4, | 9, | 16, | 25, | ... |

1. Druhých mocnin je méně, jelikož jsou jen výběrem z přirozených čísel (jejich podmnožinou),

2. je jich stejně, protože ke každému přirozenému máme jeho druhou mocninu.

Je zřejmé, že oba principy srovnávání, z nichž jeden tvrdí, že jich je stejně, a druhý, že jich je různě, nemohou platit zároveň [1].

Uvažme první princip. Můžeme například definovat, že $A$ je vlastní podmnožinou $B$ právě tehdy, když $A$ má méně prvků. Avšak už pro konečné množiny se může stát, že chceme porovnat množiny, které vůbec v relaci „býti podmnožinou“ nejsou, např. $U = \{1, 2, 3, 4\}$ a $V = \{a, b, c, d, e\}$. Označením

[1] Problém byl (už od Aristotela) s existencí nekonečných souborů. Otázka zněla, zda vůbec nekonečné soubory existují. Považujeme-li nekonečně věcí za jeden soubor (jeden hotový soubor), jedná se o aktuální nekonečno. Přijmeme-li, že takové nekonečné soubory existují, vyvstává další otázka, jakým způsobem lze srovnávat jejich velikost. A zde se objevuje Galileův paradox.

$|U| = 4$, čímž míníme, že $U$ obsahuje 4 prvky a $|V| = 5$ takže $|U| < |V|$ a přitom $U \not\subset V$. Tedy počet prvků těchto konečných množin není srovnatelný. Také neurčíme například vztah mezi mezi $S$ (sudými) a $L$ (lichými čísly).

Druhý princip, jenž přijal Georg Cantor, je silnější než ten první. Je to vidět už na příkladu se sudými a lichými čísly. Je založený na pojmu zobrazení. Dvě množiny mají stejný počet prvků, když mezi nimi existuje bijekce, funkce $f$, která každému prvku z jedné množiny $A$ přiřadí unikátní prvek z druhé množiny $B$ a zároveň vyčerpá všechny prvky $B$, formálně:

$$(\forall x_1, x_2 \in A)\Big((x_1 \neq x_2) \Rightarrow \big(f(x_1) \neq f(x_2)\big) \wedge (\forall y \in B)(\exists z \in A)\big(y = f(z)\big)\Big).$$

## 1.1 Mohutnosti množin

### 1.1.1 Bijekce

**Definice 1.1.1.** Množina je konečná, pokud je možné vytvořit bijekci

$$f : S \to \{1, \ldots, n\} \text{ pro } n \in \mathbb{N}.$$

Zjistit velikost konečné množiny je jednoduché. Pro množinu $\mathbb{M} = \{1, 2, 3\}$ platí, že počet jejích prvků $|\mathbb{M}| = 3$. Ale jestliže se budeme ptát na množinu všech sudých čísel $S = \{2, 4, 6, 8, \ldots\}$, kolik je $|S|$? Těžko říci přesně, jediná odpověď, která připadá v úvahu je nekonečno. Tím zde ale máme problém, jak pak dvě nekonečné množiny porovnat. Takže otázka, která z množin $S$ nebo přirozená čísla $\mathbb{N} = \{1, 2, 3, 4, \ldots\}$ je větší, je problematická, protože porovnávat $\infty > \infty$ nelze. My ale vytvoříme bijekci, která nám pomůže, tedy funkci která každému jednomu prvku z první množiny přiřadí jiný prvek z druhé. Následující funkce $f(x) = 2x$ tento požadavek splňuje a proto tvoří bijekci mezi přirozenými a sudými čísly. Bijekce říká, že ke každému jednomu patří jiný. Z toho vyplývá, že je jich stejný počet, a že jsou ekvivalentní.

$$S \approx \mathbb{N}$$

**Definice 1.1.2.** Pro dvě množiny $A$ a $B$ platí:

- $A$ a $B$ jsou ekvivalentní ($A \approx B$) právě tehdy, když existuje bijekce mezi prvky množiny $A$ a prvky množiny $B$.
- Existuje-li prosté zobrazení z množiny $A$ do množiny $B$, pak řekneme, že $A$ je subvalentní $B$, zapisujeme $A \preceq B$.
- Pokud platí $A \preceq B$ a neplatí $A \approx B$, pak zapisujeme $A \prec B$.

Množiny $A$ a $B$ jsou ekvivalentní právě tehdy, když existuje bijekce z $A$ do $B$. To znamená, že je možné vytvořit dvojici $[a, b]$, kdy $a \in A$ a $b \in B$, prvky

této dvojice z obou množin odebrat a opakovat (tvořit a odebírat) s $A$ a $B$, kterým byl prvek odebrán do té doby, než z obou množin zůstanou prázdné množiny. Pokud nelze obě množiny vyprázdnit v jednom kroku, není možné vytvořit bijekci. Pokud již prázdné byly, je to v pořádku, a prohlásíme je za ekvivalentní.

Ukázka na sudých $S$ a lichých $L$ číslech, kde intuice napovídá, že by jich mělo být stejně. Po vytvoření dvojic $[1,2]$, $[3,4]$, $[5,6]$,... je možné napsat, že každá dvojice je $[l, l+1]$ nebo $[s-1, s]$. Tím je vytvořena bijekce mezi množinami $S$ a $L$ a lze tak říci, že jsou stejně velké $S \approx L$ ($|\,S\,| = |\,L\,|$).

### 1.1.2 Přirozená čísla

Je snadné určit, že nějaká množina je ekvivalentní s přirozenými čísly. Stačí její prvky uspořádat a očíslovat.

**Tvrzení 1.1.1.** *Tyto množiny jsou ekvivalentní s přirozenými čísly:*

1. *celá čísla* $\mathbb{Z}$
2. *sudá čísla* $S$ *(resp. lichá čísla* $L$*)*
3. *uspořádané dvojice přirozených čísel* $\mathbb{N} \times \mathbb{N}$
4. *racionální čísla* $\mathbb{Q}$

*Poznámka* 1.1.1. Takovýto přístup obsahuje rozřešení Galileova paradoxu. $\mathbb{N} \approx Q$. Použijme bijekci, která každému přirozenému číslu přiřadí jeho druhou mocninu, $f(x) = x^2$.

*Důkaz.*

1. $\mathbb{Z}$ – neformálně naznačíme důkaz takto: přirozená čísla $\mathbb{N}$ zapíšeme do posloupnosti $(1, 2, 3, 4, 5, \ldots)$ a vytvoříme zobrazení na posloupnost takovouto $(0, 1, -1, 2, -2, 3, \ldots)$.
2. $S$ (resp. $L$) – lze si pomocí funkce $f(x) = 2x$ (resp. $f(x) = 2x - 1$) tvořit bijekci mezi přirozenými a sudými (lichými) čísly.
3. $\mathbb{N}\times\mathbb{N}$ – očíslování uspořádaných dvojic $(i, j)$ předpisem $\frac{(i+j-2)(i+j-1)}{2} + j$ vzniká $\{(1,1),(2,1),(1,2),(3,1),(2,2),(1,3),(4,1),\ldots\}$
4. $\mathbb{Q}$ – podobně jako předchozí příklad. Namísto $(i, j)$ píšeme $\frac{i}{j}$, pak dvojice které reprezentují stejnou hodnotu ($\frac{1}{1}$ a $\frac{2}{2}$) zapsat pouze jednou. Tak vznikne $\mathbb{Q}^+$ a celé $\mathbb{Q}$ vznikne jako v prvním bodě pro $\mathbb{Z}$

□

*Poznámka* 1.1.2. Pokud existuje bijekce mezi množinou $A$ a přirozenými čísly, potom nazýváme $A$ souborem spočetným, neboli lze jej očíslovat přirozenými čísly. Tedy všechny uvedené množiny jsou spočetné.

### 1.1.3 Mohutnosti množin

**Definice 1.1.3.** Množina $A$ je konečná, jestliže existuje bijekce $A$ na nějaké přirozené číslo $n \in \mathbb{N}$. Pak definujeme, že mohutnost množiny $A$ je $n$,

$$|A| = n.$$

Mohutnost přirozených čísel se nazývá spočetná, označuje se symbolem $\aleph_0$.

$$|\mathbb{N}| = \aleph_0.$$

**Definice 1.1.4.** (Srovnání mohutností množin.) Nechť $A$, $B$ jsou dvě množiny, $|A|, |B|$ označují jejich mohutnosti.

- $(A \approx B) \Leftrightarrow (|A| = |B|)$
- $(A \preceq B) \Leftrightarrow (|A| \leq |B|)$
- $(A \prec B) \Leftrightarrow (|A| < |B|)$

Podle věty 1.1.1 jsou množiny sudých čísel, celých čísel i racionálních čísel ekvivalentní s množinou přirozených čísel. Jsou tedy také spočetné.

**Věta 1.1.2.** *Platí:*

- *Pro každé dvě množiny $A$, $B$, platí právě jeden ze vztahů*

$$|A| = |B|, |A| < |B|, |A| > |B|$$

- *Je-li $B \subset A$, pak $|B| \leq |A|$ (zkratka za: $|B| < |A|$ nebo $|B| = |A|$)*

*Důkaz.* Důkaz není triviální, a proto odkáži na knihu [2], s. 166. Jestliže přijmeme axiom výběru, pak lze každou množinu dobře uspořádat, tedy každá množina má své kardinální číslo. Kardinální čísla lze dobře uspořádat. Tedy mohutnosti každých dvou dobře uspořádaných množin lze porovnat. □

### 1.1.4 Mohutnost reálných čísel

Otázka je, zda je množina reálných čísel $\mathbb{R}$ spočetná, tj. zda $\mathbb{R} \approx \mathbb{N}$. Prozatím budeme uvažovat pouze interval reálných čísel $(0, 1)$ Jinak řečeno, jestli je možné vytvořit jednoznačné zobrazení z $\mathbb{N}$ do $(0, 1)$ neboli reálná čísla z intervalu $(0, 1)$ očíslovat přirozenými čísly. K vyvrácení předpokladu $\mathbb{N} \approx (0, 1)$ se používá *Cantorova diagonální metoda.*

**Věta 1.1.3.** *Platí:* $\mathbb{N} \not\approx (0, 1)$.

*Důkaz.* Postupujeme sporem. Předpokládáme, že interval reálných čísel $(0, 1)$ je spočetný, tedy jej můžeme uspořádat a očíslovat přirozenými čísly.

| $i =$ | | 1 | 2 | 3 | 4 | ... |
|---|---|---|---|---|---|---|
| $r_1 =$ | 0, | **1** | 4 | 3 | 8 | ... |
| $r_2 =$ | 0, | 7 | **5** | 1 | 6 | ... |
| $r_3 =$ | 0, | 6 | 3 | **4** | 7 | ... |
| $r_4 =$ | 0, | 1 | 6 | 8 | **0** | ... |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ |
| $r_x = d =$ | 0, | 1 | 5 | 4 | 0 | ... |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ |
| $d' =$ | 0, | 8 | 4 | 5 | 9 | ... |

Tabulka 1.1: Spočetnost reálných čísel

1. Čísla zapíšeme do seznamu
2. Každé číslo lze „zapsat“ v desetinném rozvoji $0, \ldots$
3. Pro ilustraci vytvoříme číslo $d$ konstrukcí přes diagonálu, aby bylo vidět, o které cifry se jedná. V čísle $r_i$ na pozici $i$ se nachází $\alpha$, $d$ je tedy složeno tak, že na $i$-tou pozici zapíšeme $\alpha$.
4. Poté vytvoříme číslo $d'$. Do čísla $d'$ vložíme $\beta = 9 - \alpha$ na pozici $i$. Pro příklad v předchozím bodě by bylo $d' = 0,8459\ldots$.
5. Číslo $d'$ je zřejmě reálné (jelikož všechny desetinné rozvoje reprezentují reálné číslo) z intervalu $(0, 1)$.
6. Mělo by tedy existovat $r_n = d'$ pro nějaké $n$, jelikož jsme předpokládali, že v posloupnosti $(r_1, r_2, r_3, \ldots)$ jsou všechna reálná čísla z intervalu $(0, 1)$.
7. Avšak kvůli konstrukci v bodě 4 je číslo $d'$ alespoň v jedné cifře rozdílné od všech čísel $r$ v posloupnosti. Tedy číslo $d'$ se v posloupnosti $(r_1, r_2, r_3, \ldots)$ nevyskytuje.
8. To je spor s předpokladem, že v seznamu se nachází všechna reálná čísel z intervalu $(0, 1)$.

Odtud plyne, že interval $(0, 1)$ není spočetný.[3] □

*Poznámka* 1.1.3. Konstrukční předpis v důkazu výše (1.1.4) pro $d'$ je obměnitelný. Stejně tak i číselná soustava. Ukážeme pro dvojkovou soustavou:

Z důkazu 1.1.4 plyne, že reálná čísla na intervalu $(0, 1)$ mají větší mohutnost než přirozená čísla. Tudíž i reálných čísel je nutně více než přirozených 1.1.2

| $i =$ | | 1 | 2 | 3 | 4 | ... |
|---|---|---|---|---|---|---|
| $r_1 =$ | 0, | **1** | 1 | 0 | 1 | ... |
| $r_2 =$ | 0, | 0 | **1** | 1 | 0 | ... |
| $r_3 =$ | 0, | 1 | 1 | **0** | 0 | ... |
| $r_4 =$ | 0, | 1 | 1 | 0 | **1** | ... |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ |
| $d =$ | 0, | 1 | 1 | 0 | 1 | ... |
| $d' =$ | 0, | 0 | 0 | 1 | 0 | ... |

Tabulka 1.2: Spočetnost reálných čísel ve dvojkové soustavě

**Věta 1.1.4.** *Mohutnost intervalu* $(0, 1)$ *je stejná jako mohutnost* $\mathbb{R}$ *tedy*

$$\mathbb{R} \approx (0, 1)$$

*Důkaz.* $f(x) = tan((x - \frac{1}{2})\pi)$ je předpisem funkce (bijekce) z intervalu $(0, 1)$ do celých reálných čísel $\mathbb{R}$. □

**Věta 1.1.5.** *Nechť* $M \subseteq N$*. Potom množina* $M$ *má mohutnost menší nebo rovnou než množina* $N$*.*

$$M \subseteq N \Rightarrow M \preceq N$$

*Důkaz.* Mějme množinu $M \subseteq N$. Uvážíme-li identické zobrazení, pak nutně $M \preceq N$.

□

**Věta 1.1.6.** *Mohutnost přirozených čísel je menší než reálných.*

$$\mathbb{N} \prec \mathbb{R}$$

*Důkaz.* Víme, že $\mathbb{N} \subset \mathbb{R}$, identita je prosté zobrazení $\mathbb{N}$ do $\mathbb{R}$. Je tedy $\mathbb{N} \preceq \mathbb{R}$. Avšak podle vět 1.1.4 a 1.1.3 dostáváme, že $\mathbb{R} \not\approx \mathbb{N}$. □

### 1.1.5 Mohutnost potenční množiny

Zobecněním předchozích tvrzení je Cantorova věta.

**Věta 1.1.7** (Cantorova věta)**.** *Pro libovolnou množinu* $x$ *má potenční množina* $\mathcal{P}(x)$ *obsahující všechny podmnožiny množiny* $x$ *větší mohutnost než* $x$*.*

$$x \prec \mathcal{P}(x)$$

*Důkaz.* Nechť $X$ je libovolná množina a $\mathcal{P}(X)$ množina všech podmnožin $X$ (tzv. potenční množina). Tvrzení, že $\mathcal{P}(X)$ má větší mohutnost než $X$, je ekvivalentní tomu, že neexistuje prosté zobrazení z $X$ do $\mathcal{P}(X)$, které by bylo na (surjektivní). Sporem:

1. Pro spor předpokládejme, že existuje prosté zobrazení $f : X \to \mathcal{P}(X)$, které je na. Tedy pro každý prvek $A \in \mathcal{P}(X)$ existuje nějaké $x$ tak, že $f(x) = A$.

2. Nyní definujme podmnožinu $Y \subseteq X$, že $Y = \{x \in X : x \notin f(x)\}$

3. Y obsahuje ty prvky $X$, které nejsou ve svém obrazu daném zobrazením $f$. $Y$ je zřejmě podmnožina $X$ a tedy musí existovat $y \in X$ tak, že $Y = f(y)$. Mohou tedy nastat dvě možnosti:

   a) $y \in Y$, to je ale spor s definicí $Y$, podle které $y \notin f(y)$, ale $f(y) = Y$,

   b) $y \notin Y$, jenže pak z definice $Y$ plyne $y \in f(y)$ a podle předpokladu $Y = f(y)$ musí platit $y \in Y$, což je opět spor.

Existence zobrazení $f : X \to \mathcal{P}(X)$, které je na, vede ke sporu a tedy $\mathcal{P}(X)$ má vždy větší mohutnost než $X$.

| | $f(x_1)$ | $f(x_2)$ | $f(x_3)$ | $f(x_4)$ | $\dots$ | $Y$ | $\dots$ |
|---|---|---|---|---|---|---|---|
| $x_1 =$ | **1** | 1 | 0 | 1 | $\dots$ | 0 | $\dots$ |
| $x_2 =$ | 0 | **1** | 1 | 0 | $\dots$ | 1 | $\dots$ |
| $x_3 =$ | 1 | 1 | **0** | 0 | $\dots$ | 0 | $\dots$ |
| $x_4 =$ | 1 | 1 | 0 | **1** | $\dots$ | 1 | $\dots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ |
| $y =$ | 0 | 0 | 1 | 0 | $\dots$ | **?** | $\dots$ |

Tabulka 1.3: Cantorova věta

□

**Důsledek 1.** *Existuje nekonečně mnoho nekonečných mohutností.*

*Důkaz.* Potenční množina přirozných čísel má větší mohutnost než přirozená čísla. Potenční množina potenční množiny má zase větší mohutnost atd.

$$\mathbb{N} \prec \mathcal{P}(\mathbb{N}) \prec \mathcal{P}(\mathcal{P}(\mathbb{N})) \prec \dots$$

□

## 1.2 Cantorův paradox

Paradox s nímž přišel Georg Cantor v roce 1899 se týkal množiny všech množin a její potenčních množiny.

**Paradox 1.2.1** (Cantorův paradox)**.** *Mějme universální množinu všech množin $V$, pak její $\mathcal{P}(V)$ má zároveň menší nebo rovnou a zároveň větší mohutnost než $V$.*

*Důkaz.* Uvažujme o množině $V$ všech množin. Každá její podmnožina je jejím prvkem. Neboli $\mathcal{P}(V) \subseteq V$ a tedy $\mathcal{P}(V) \preceq V$.

Zároveň však podle Cantorovy věty má její potenční množina $\mathcal{P}(V)$ větší mohutnost než samotná $V$, $V \prec \mathcal{P}(V)$. A to je spor.

□

## 1.3 Význam

Lékem na paradoxy teorie množin, jako je ten zmíněný výše, bylo vytvoření axiomatické teorie množin, která vyloučila objekty podobné *množině všech množin* i další problematické množiny a začala je nazývat třídami, pro které již vlastnost, jako mít potenční množinu, nemá smysl a tudíž ani není definována.

Z Cantorovy věty plyne, že pro každou nekonečnou množinu existuje množina s řádově větší mohutností. To znamená, že máme různě veliká nekonečna. Byla vytvořena teorie množin, která zkoumá vlastnosti a vztahy mezi nekonečnými kardinálními čísly značenými $\aleph$. Vznikly i další možnosti, jak vytvořit kardinální čísla. V knize [2] se čtenář může dovědět více.

Avšak ani v axiomatickém systému teorie množin nelze určit, zda existuje nějaká množina, jejíž mohutnost by byla větší než mohutnost $\mathbb{N}$ a menší než mohutnost $\mathbb{R}$. Cantor se domníval, že tomu tak není, a že mohutnost reálných čísel je první následující za mohutností přirozených čísel neboli $\mathbb{R} \approx \aleph_1$. To se nazývá hypotéza kontinua.

Ukázalo se, že tato hypotéza není z axiomů teorie množin ani dokazatelná ani vyvratitelná. Jedná se o tzv. nezávislé tvrzení.

Kapitola 2

# Gödelova věta

## 2.1 Hilbertův program

V době kdy vznikl Gödelův důkaz (1931) si matematici dali za úkol sepsat všechna pravidla (axiomy), která platí, a ze kterých se bude dát vše odvodit. S tímto úkolem přišel David Hilbert, který formuloval cíle tzv. *Hilbertova programu*.

Hilbertův program měl formalizovat matematiku, jinými slovy zavést formální systém, který bude mít svůj formální jazyk, ve kterém bude možné zapsat veškerá matematická tvrzení a budou platit pravidla podle kterých bude možné pracovat s těmito tvrzeními. Tento systém bude:

- Úplný – každé pravdivé tvrzení bude možné v rámci systému dokazatelné. Jinými slovy každá formule bude dokazatelná či vyvratitelná důkazem její negace.

- Bezesporný (konzistentní) – bude obsahovat důkaz, že nelze zkonstruovat kontradikce. Tedy neplatí formule i její negace zároveň. Taktéž by tento důkaz měl obsahovat konečný počet formulí o konečných matematických objektech.

- Rozhodnutelný – měl by být k dispozici algoritmus k rozhodnutí zda dané tvrzení v systému platí, či naopak.

- Uzavřený – důkaz, že výsledky získané s použitím „ideálních objektů“ je možné dokázat i bez použití takových „ideálních objektů“.[4]

A právě Kurt Gödel dokázal, že takový cíl, kdy je systém zároveň bezesporný i úplný, je neuskutečnitelný.

## 2.2 Paradoxy

Matematická logika se zabývá nejrůznějšími tématy a jedním z nich jsou paradoxy. Paradox je slovo pocházející z řeckého *paradoxos*, složeného z předpony *para-* a kmene *doxa*. *Doxa* se běžně překládá jako mínění, tvrzení o něčem, i když může mít význam zdání, *para-* naproti tomu označuje něco vedle, při nebo proti něčemu.

Paradoxy tedy byly známy již ve starém Řecku, kde měly několik funkcí. První jako ukazatel nespolehlivosti našich smyslů při poznávání světa, další jako hříčky pro pobavení, anebo byly vysvětlovány jako podivné a tajemné úkazy, možná vedoucí k průniku do hlubšího stupně našeho poznání.

Paradox je tedy tvrzení, které si alespoň zdánlivě protiřečí. Odborněji: pokud můžeme v nějaké teorii formulovat paradox, znamená to, buď že je sporná nebo že je nějakým způsobem neúplná.

Logický paradox, o kterém je další sekce, má vlastnost autoreference, tj. schopnost „mluvit o sobě samém“, která, jak uvidíme, bude užitečná.

## 2.3 Paradox lháře

První nejstarší výskyt *paradoxu lháře* byl z antiky. Byl údajně vysloven Eubulidem z Milétu. Paradox lháře je příkladem autoreferenčního paradoxu a je možné jej obměnit v různých dalších variantách, ale zdálo se, že na moderní logiku se nevztahuje, protože se nenachází v „sémanticky uzavřeném jazyce“, takže nemůže vypovídat sám o sobě.

**Paradox 2.3.1.** *Eubulides: „Muž říká, že lže. Je to co řekl pravda nebo lež?“*

Podle principu „vyloučeného třetího“ tento muž buď mluví pravdu anebo nemluví pravdu, tedy lže.

- Jestliže mluví pravdu, pak lže, což je spor.
- Jestliže lže, pak platí negace toho, co říká, tedy není pravda, že lže neboli mluví pravdu, což je zase spor.

Za předpokladu, že věta je buď pravdivá nebo nepravdivá. Obě varianty vedou ke sporu, tedy se jedná o paradox.

Alfred Tarski zjistil, že tento paradox se nachází pouze v „sémanticky uzavřených jazycích“, jsou to takové jazyky, které vlastním větám dovolují vypovídat o pravdivostní hodnotě jiných vět, nebo dokonce i o své vlastní. Podobně jako v tomto paradoxu, nachází se zde predikát pravdivosti (mluví pravdu $\times$ lže) a zároveň vypovídá o vlastním tvrzení (vedlejší větou).

## 2.4 Richardův paradox

Následující text v celé této kapitole vychází z knihy Nagela a Newmana Gödelův důkaz [5].

Mluvíme-li o nějakém formálním systému, např. matematice, je třeba zásadním způsobem rozlišit tvrzení matematická ($1 + 1 = 2$) od tvrzení o matematice, tedy od tvrzení metamatematických („$1 + 1 = 2$“ je jednoduchá rovnice). Pak je ovšem třeba zodpovědět otázku, jakým způsobem metamatematická tvrzení (a tedy naše úvahy a důkazy o tomto systému) formulovat – není totiž zase tak obtížné zdánlivě vyvodit spor v nějakém formálním systému, jestliže způsob, jakým o tomto systému budeme mluvit (například přirozený jazyk), bude mít mnohem silnější vyjadřovací schopnosti než systém samotný (což zřejmě přirozený jazyk splňuje).

Příkladem takového zdánlivého sporu je tzv. *Richardův paradox*, ten se většinou uvádí ve zkrácené podobě, která je nepřesvědčivá. Tou je *paradox sta slov*.

**Paradox 2.4.1** (Paradox sta slov)**.** *Česká abeceda obsahuje konečně mnoho písmen. Proto českých slov majících méně než sto písmen je také konečně mnoho. Stejně tak i všech českých vět obsahujících méně než sto slov (s méně než sto písmeny) je konečně mnoho. Jen některé z těchto vět definují jednoznačně nějaké přirozené číslo (například „Padesát šest“ nebo „Třetí mocnina největšího dvanácticiferného prvočísla“). Tedy všech vět v češtině majících méně než sto slov, z nichž každé obsahuje méně než sto písmen české abecedy, které definují nějaké přirozené číslo, je jen konečně mnoho. Všech přirozených čísel je však nekonečně mnoho. Proto musí existovat přirozené číslo, které žádnou větou splňující výše popsané podmínky definovat nelze, a tedy existuje nejmenší takové přirozené číslo. Pak ovšem věta „Nejmenší přirozené číslo, které není možné definovat pomocí věty o méně než sto slovech, z nichž každé má méně než sto písmen české abecedy“ je větou o méně než sto slovech, z nichž každé má méně než sto písmen české abecedy, která toto číslo definuje. Tedy číslo, náležející mezi čísla nedefinovatelná, je zároveň definováno právě touto větou.*[6]

Samotný *Richardův paradox* zní takto:

**Paradox 2.4.2** (Richardův paradox)**.** *Uvažujme libovolný přirozený jazyk a množinu přirozených čísel* $\mathbb{N}$*. Dále předpokládejme, že v daném přirozeném jazyce lze o přirozených číslech mluvit a přiřazovat jim tak různé vlastnosti, jako např. „být sudé“, „být prvočíslo“, „mít jako poslední cifru jedničku“ apod. Jestliže vycházíme z toho, že každý přirozený jazyk obsahuje pouze konečně mnoho slov, pak každá taková formulace vlastnosti čísel je také konečná a můžeme všechna tato tvrzení o číslech lexikograficky uspořádat(abecedně seřadit) a tím každému tvrzení přiřadit nějaké pořadí.*

*Shodou okolností se však může (ale nemusí) stát, že číslo (pořadí) určitého tvrzení bude mít vlastnost, kterou toto tvrzení formuluje – například že na 4. místě bude tvrzení „být sudé“. O takovém číslu prohlásíme, že není Richardovské a obráceně ostatní čísla tvrzení (ta, která nemají vlastnost popisovanou tímto tvrzením) označíme jako Richardovská.*

*Tím jsme ovšem opět korektně formulovali vlastnost přirozených čísel („být Richardovské“). Takové tvrzení tedy nutně je zahrnuto v našem uspořádání a má nějaké pořadí $n$. Ptejme se nyní, zda pro $n$ platí, že je Richardovské. Jestliže ano, pak nemá vlastnost, kterou popisuje $n$-té tvrzení, tedy nemá vlastnost „být Richardovské“ a tedy není Richardovské. Naopak, pokud by $n$ nebylo Richardovské, tak má vlastnost, kterou popisuje $n$-té tvrzení, tedy má vlastnost „být Richardovské“ a tedy je Richardovské. Zkrátka $n$ je Richardovské právě tehdy, když není Richardovské a spor je na světě.*

| | $x =$ | 1 | 2 | 3 | 4 | … | $n$ | … |
|---|---|---|---|---|---|---|---|---|
| 1) | „číslo $x$ je liché“ | **1** | 0 | 1 | 0 | … | ? | … |
| 2) | „číslo $x$ je sudé“ | 0 | **1** | 0 | 1 | … | ? | … |
| 3) | „číslo $x$ je kladné“ | 1 | 1 | **1** | 1 | … | 1 | … |
| 4) | „číslo $x$ má jako poslední cifru jedničku“ | 1 | 0 | 0 | **0** | … | ? | … |
| ⋮ | | | | | | | | |
| $n$) | „číslo $x$ je Richardovské“ | 0 | 0 | 0 | 1 | … | **???** | … |

Tabulka 2.1: Richardův paradox

Klíčový problém paradoxu spočívá v odlišení přirozeného jazyka (metajazyka) od jazyka speciálního, určeného pro mluvení o objektech nějaké užší oblasti našeho zájmu. Je povoleno mluvit přirozeným jazykem o jazyce speciálním, ne však mluvit přirozeným jazykem o jazyce přirozeném ani mluvit speciálním jazykem o jazyce speciálním či přirozeném. Dá se to představit jako úrovně a je zakázáno hovořit o úrovni stejné či vyšší. Richardův paradox zase mylně předpokládá, že v původních všech vlastnostech se již nacházela vlastnost „být Richardovským číslem“, jenže to není pravda. Tu jsme si dodefinovali. Navíc definice trpí nekonečným odkazem na sebe samu, jenž nikdy nemůžeme ohodnotit, neboli se ptáme zas na definici, kterou chceme právě definovat: používá se pro definici soubor definicí, ve které se sama nachází.

„Být Richardovským“ bylo dodefinováno v jazyce vyššího řádu. Taková vlastnost nikdy nemohla být ve výčtu vlastností patřících do matematiky (matematika samotná neobsahuje vlastnost „být Richardovským“). A právě tomuto se Gödel geniálně vyhnul právě způsobem popsaným v kapitole 2.6 s názvem *Schéma Gödelova argumentu*.

## 2.5 Gödelova numerace

Gödelova numerace neboli číslování je postup, kterým se k jakékoli správně sestavené formuli přiřadí jedinečné číslo (Gödelovo číslo), tak, že z něj lze zpětně „dekódovat“ přesný tvar původní formule. Kurt Gödel v podstatě vytvořil zobrazení formulí do přirozených čísel a zpět. A navíc při výpočtu těchto čísel zjistíme, že lze přiřazovat číslo i symbolu a hlavně i **posloupnostem formulí**. Posloupnost formulí totiž reprezentuje důkaz. Bude klíčové, aby také měl své Gödelovo číslo.

Určení Gödelova čísla pro celou formuli (tj. posloupnost takových symbolů) probíhá tak, že každému symbolu se přiřadí po řadě prvočíslo umocněné na Gödelovo číslo daného symbolu a číslo celé formule je součinem těchto umocněných prvočísel. Tento postup je jednoznačný a díky faktorizaci (tj. rozkladu čísla na součin) se dají zpětně rekonstruovat formule z Gödelových čísel.

Postup není fixně daný, existují obměny v definicích. Hlavní je princip, který Gödel využil.

Potřeba bude zakódovat překlad, k tomu nám pomohou tabulky 2.2, 2.3 a ilustrace v tabulce 2.4, inspiraci pro tyto tři tabulky jsem našel také v [5].

| Symbol | Gödelovo číslo | Význam |
|---|---|---|
| $\neg$ | 1 | negace |
| $\vee$ | 2 | nebo |
| $\Rightarrow$ | 3 | implikace |
| $\exists$ | 4 | existenční kvantifikátor |
| $=$ | 5 | rovnost |
| 0 | 6 | nula |
| $s$ | 7 | následník |
| ( | 8 | interpunkční znaménko |
| ) | 9 | interpunkční znaménko |
| , | 10 | interpunkční znaménko |
| $+$ | 11 | sčítání |
| $\times$ | 12 | násobení |

Tabulka 2.2: Gödelovo číslování – Konstanty

| Číselné proměnné | Gödelovo číslo |
|---|---|
| $x$ | 13 |
| $y$ | 17 |
| $z$ | 19 |

číselné proměnné patří k prvočíslům větším než 12.

Tabulka 2.3: Gödelovo číslování – Proměnné

Nejlépe se dá číslování vyložit na příkladu.

Vezměme formuli $(x = 0)$. Převedeme ji následujícím způsobem.

$$\begin{array}{ccccc} ( & \mathrm{x} & = & 0 & ) \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 8 & 13 & 5 & 6 & 9 \end{array}$$

Je však potřeba formuli přiřadit pouze jedno číslo, nikoli posloupnost. Bude jím násobek po sobě jdoucích prvočísel umocněných na daná Gödelova čísla.

$$f = 2^8 \times 3^{13} \times 5^5 \times 7^6 \times 11^9$$

Tím máme číslo formule, ještě je možné spočíst slibované číslo posloupnosti formulí, používaných v důkazech. Postupuje se stejně, jako když jsme spojovali čísla symbolů do čísla formule, jen teď spojujeme čísla formulí do čísla posloupnosti formulí. Pro případ dvou po sobě jdoucích formulí s čísly $m$ a $n$ by vztah výsledného čísla $l$ byl $l = 2^m \times 3^n$.

Ukázali jsme si, že je možné „zakódovat“ každý symbol, formuli i posloupnost formulí. Zbývá ještě ukázat „cestu zpět“. K tomu nám pomůže faktorizace a tabulka Gödelovo číslování – Postup.

| |
|---|
| $243\ 000\ 000$ |
| $64 \times 243 \times 15\ 265$ |
| $2^6 \times 3^5 \times 5^6$ |
| $\begin{array}{ccc} 0 & = & 0 \\ \downarrow & \downarrow & \downarrow \\ 6 & 5 & 6 \end{array}$ |
| $0 = 0$ |

Tabulka 2.4: Gödelovo číslování – Postup

Je nutné si povšimnout, že užitím Gödelova očíslování lze mluvit o aritmetických formulích opět prostřednictvím aritmetiky samotné. Takto lze například vyjádřit, že formule z našeho příkladu s Gödelovým číslem f zahrnuje symbol „=“ tak, že prvočíselný rozklad čísla f obsahuje právě pátou mocninu prvočísla pět, což lze aritmeticky vyjádřit. Gödel založil svůj důkaz právě na tom, že ukázal, jak lze aritmeticky konstruovat složitější formule vyjadřující se o (ne)dokazatelnosti nějakého tvrzení.

### 2.5.1 Dokazatelnost

Gödel geniálně používal termín *dokazatelnost* namísto přímých tvrzení. Tak jako existuje aritmetické tvrzení „obsahuje číslici“, tak zadefinoval tvrzení je dokazatelné z posloupnosti formulí. Máme tedy $dem(x, y)$*, jenž tvrdí: „Posloupnost formulí s číslem x je důkazem formule s číslem y.“ Fakt, proč je možné toto definovat byl odvozen z jistého vztahu, stejně tak, jako byl zadefinován výpočet čísla posloupnosti $l = 2^m \times 3^n$. Existuje jistá relace mezi $l$ a $n$, i když nijak jednoduchá.

*Poznámka* 2.5.1. Pokud kterékoli číslo vložené jako argument do funkce *dem* není Gödelovým číslem, funkce vrací hodnotu 0 znamenajíc, že nic „nedemonstruje“.

### 2.5.2 Nahrazování

Poslední záležitostí, kterou je třeba podat před samotným jádrem Gödelova důkazu, je poslední funkce s názvem *sub*. Její argumenty jsou tři: Gödelovo číslo formule, číslo proměnné a dosazené číslo. Funkce $sub(x, 17, z)$ má za úkol vzít formuli s Gödelovým číslem $x$ dosadit za proměnnou $y$ (ta má Gödelovo číslo 17) číslo $z$. Na příkladu by to vypadalo takto: $x$ je číslo formule $(\exists x)(x = sy)$, která tvrdí, že $y$ ná bezprostředního následníka $x$, $z$ bude rovno 0 (a stále je 17 Gödelovým číslem proměnné $y$), pak číslo $s = sub(x, 17, z)$ reprezentuje formuli $(\exists x)(x = s0)$ a je spočetné. Funkce *sub* se bude často používat ve tvaru $s = sub(x, 17, x)$ jejíž význam je, že do formule s číslem $x$ dosaď za proměnnou $y$ číslo formule $x$.

## 2.6 Schéma Gödelova argumentu

Začátkem bude obecný rozbor v pěti krocích, aby bylo možné sledovat celý postup.

Prvním krokem je zkonstruovat formuli $G$, která tvrdí: „Formule $G$ je nedokazatelná.“ Ta bude použita podobně jako se používá v *Richardově paradoxu (2.4.2)* tvrzení „číslo x není Richardovské“. Zde se tedy bude používat formule s (Gödelovým) číslem $g$ je nedokazatelná.

Druhým krokem je to, co také vyplývá z *Richardova paradoxu*, že formule G je dokazatelná právě tehdy, když je dokazatelná její negace $\neg G$. To zní jako další paradox, ale je zapotřebí tento fakt spojit s „bezesporností“ systému. A tím pádem z toho plyne, že: Pokud je systém bezesporný, pak je formule $G$ formálně nerozhodnutelná. To znamená, že není rozhodnutelná $G$ ani $\neg G$.

*„dem“ je zde zkratka z německého „Demonstration“, tj. důkaz

Co více, třetím krokem bude ukázáno, že formule $G$ i přes to, že není formálně rozhodnutelná, je i tak pravdivá.

Předposledním krokem bude ukázat, co vyplývá z kroků předchozích a tím je, že systém není úplný, jelikož obsahuje formuli $G$, která není rozhodnutelná v rámci systému, ale přesto víme, že je pravdivá. Jako kdyby v systému nějaký axiom chyběl, na to však Gödel odpovídá tím, že systém je *podstatně neúplný*.

Finálním krokem k završení je ukázat konstrukci formule $A$, která tvrdí, že systém je bezesporný. Podařilo se dokázat formuli $A \Rightarrow G$, $G$ platí (ze třetího bodu). Avšak formule $A$ dokazatelná není. To znamená, že bezespornost systému nemůže být dokázána žádnou posloupností odvození v rámci daného systému. Tím se myslí, že ať se udělají jakákoli odvození, dokázat bezespornost systému v něm samém je nemožné.

## 2.7 Gödelova věta

**Věta 2.7.1** (Gödelova věta)**.** *Každý formální systém zahrnující alespoň aritmetiku přirozených čísel buď není bezesporný, nebo není úplný.*

*Důkaz.* Gödel ukázal, jak lze s pomocí jím definovaného očíslování sestrojit formuli $G$, která (sama o sobě) tvrdí, že „$G$ není dokazatelná“.

Konstrukce je následující:

Vezměme formuli $(\exists x)(dem(x, y))$ znamenající: „Formule s číslem y je dokazatelná.“

Vytvoříme její negaci $\neg(\exists x)(dem(x, y))$.

Pokročíme dál a rozšíříme za y dosadíme formuli $sub(y, 17, y)$, dohromady $\neg(\exists x)(dem(x, sub(y, 17, y)))$.

V mezikroku definujeme $n$. Hodnota $n$ je rovna Gödelovu číslu této formule: $\neg(\exists x)(dem(x, sub(n, 17, n)))$.

Využijeme funkci *sub*, k tomu abychom nahradili výskyty $y$ za číslo formule $sub(n, 17, n)$, tak dostaneme formuli $\neg(\exists x)(dem(x, sub(n, 17, n)))$, která tvrdí: „Formule s číslem $n$ po nahrazení všech proměnných $y$ za číslo $n$ nemá žádný důkaz.“

Tím máme slíbenou formuli $G$: $\neg(\exists x)(dem(x, sub(n, 17, n)))$ s významem: „Formule s číslem $n$ po dosazení proměnných $y$ číslem $n$ nemá důkaz.“ Tato formule má všechny proměnné kvantifikované, takže lze o ní rozhodnout zda je pravdivá, či ne, zároveň o sobě tvrdí, že není dokazatelná.

Číslo $g$ je tedy rovno $sub(n, 17, n)$.

Jádro jeho postupu spočívá v tom, že přesně ukázal, jak lze najít číslo $g$, jenž je Gödelovým číslem formule $G$, která pro rekapitulaci tvrdí: „Formule s Gödelovým číslem $g$ není dokazatelná“, a jak tuto formuli způsobem uvedeným v předchozím odstavci vyjádřit, tedy $\neg(\exists x)dem(x, g)$. Co ovšem plyne z $G$?

- Je-li $G$ dokazatelná, pak platí $G$, tj. že $G$ není dokazatelná.
- Naopak, není-li $G$ dokazatelná, pak platí to, co $G$ tvrdí.

Výsledkem je, že $G$ je dokazatelná právě tehdy, když $\neg G$ je dokazatelná, tedy spor.

Určitě platí, že systém obsahuje $G$ nebo $\neg G$, pak ale není bezesporný, ale jestliže bezespornost požadujeme, pak nesmí daný systém obsahovat $G$ ani $\neg G$. Formule $G$ je v daném systému nerozhodnutelná, neboť neexistuje žádný důkaz formule $G$. Tím se dostáváme ale k tomu, že formule $G$ je pravdivá, neboť přesně to $G$ sama o sobě tvrdí, že není dokazatelná. To nás vede k faktu, že jsme našli pravdivou formuli, jenž není v systému dokazatelná, proto můžeme tvrdit, že systém je neúplný.

Zatímco opačně platí, že pokud je systém úplný, pak obsahuje $G$ i $\neg G$ a tedy není bezesporný, tímto je *Gödelova věta* dokázána.

□

*Poznámka* 2.7.1. Kromě toho Gödel také dokázal, že takový systém je tzv. *podstatně neúplný*. Tím je myšleno, že neexistuje konečně mnoho axiomů, kterých by mohlo být přidáno, aby byl systém úplný. Plyne to z podstaty konstrukce formule $G$, která nezávisí na systému jako takovém, takže v případě že přidáme původní $G$ mezi axiomy, tvoříme nový systém ve kterém stále je možné vytvořit $G'$, pro kterou bude platit výše uvedený spor.

## 2.8 Důsledky

První Gödelova věta nachází omezení každého formálního systému zahrnujícího aritmetiku. Proto se dotýká mnoha vědních oblastí zahrnujících logiku, matematiku, informatiku i filosofii. Pro informatiku má zásadní význam v teorii vyčíslitelnosti, protože klade omezení na jakoukoli výpočetní sílu používanou pro algoritmizaci úloh. Tou nejznámější úlohou, na kterou se podíváme i v další kapitole, je tzv. „problém zastavení“.

Kapitola 3

# Turingův stroj – problém zastavení

Turingův stroj ($TS$) je teoretický model počítače používající se v teorii vyčíslitelnosti pro výpočty. Nás bude zajímat problém zastavení, jelikož je dalším negativním matematickým tvrzením, které říká, že $TS$ nemůže rozhodnout jestli jiný $TS$ zastaví a tedy nám vrátí nějaký výsledek, ať už je správný nebo ne, a nebo nezastaví a tedy se výsledku nikdy nedočkáme.

Čtenáři, kteří znají pojem Turingův stroj a terminologii s ním spojenou, mohou pokračovat sekcí *Univerzální TS*.

Veškeré vysvětlované pojmy v této kapitole jsou vybrané z [7].

## 3.1 Základní pojmy

**Algoritmem** míníme mechanizovaný postup instrukcí, které lze realizovat na jakémkoli $TS$.
**Abecedou** máme na mysli konečnou a neprázdnou množinu. Její prvky se nazývají symboly.
**Slovem** $w$ o délce $l$ nad abecedou $\Sigma$ rozumíme uspořádanou $l$-tici symbolů z abecedy $\Sigma$, kdy $l$ je konečné přirozené číslo. Délku slova značíme $|w| = l$. Například slovo $w$ délky 5 nad abecedou $\Sigma$:

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

$$w = 11011$$

Prázdné slova, nebo také slova délky 0, značíme $\varepsilon$.

Množina všech slov nad abecedou se dále značí $\Sigma^*$:

$$\Sigma^* = \{\varepsilon\} \cup \Sigma \cup (\Sigma \times \Sigma) \cup (\Sigma \times \Sigma \times \Sigma) \cup \dots$$

**Jazyk** $L$ nad abecedou $\Sigma$ je libovolná množina slov ze $\Sigma^*$.
**Konečný automat** je teoretický výpočetní model používaný v informatice pro studium formálních jazyků. Jeho činnost spočívá v tom, že se může nacházet pouze v jednom stavu v jeden moment, přecházet mezi stavy v závislosti na čteném symbolu. Konečný automat obsahuje konečnou množinu stavů. Je výpočetně jednoduchým modelem.

### 3.1.1 Turingův stroj (TS)

Turingův stroj je teoretický model počítače, který navrhl Alan M. Turing v roce 1936. Skládá se z

- procesorové jednotky, tvořené konečným automatem,
- programu, který je vymezen pravidly (tzv. přechodovou funkcí)
- a pásky, která není omezená na počtu informací, jenž pojme.

Turingův stroj vznikl za účelem simulovat výpočetní sílu počítače a aby bylo možné jej teoreticky pojmout a tvořit důkazy o počítačích. K tomu abychom mohli tvrdit, že počítač (počítající nějaký algoritmus, potažmo tedy algoritmus nebo jakýkoli jiný „rozumný“ výpočetní model) a Turingův stroj mají stejnou výpočetní sílu, potřebujeme větu známou pod jménem Church-Turingova teze.

**Věta 3.1.1** (Church-Turingova teze)**.** *„Ke každému algoritmu existuje ekvivalentní Turingův stroj.“*

*Důkaz.* Kvůli neformální definici pojmu *algoritmus* nemůže být tato teze nikdy dokázána, lze ji však vyvrátit, podaří-li se sestrojit stroj, který bude umět řešit problémy, které Turingův stroj řešit neumí (např. již zmiňovaný problém zastavení TS). □

**Definice 3.1.1.** TS je reprezentován sedmicí $(Q, \Sigma, G, \delta, q_0, B, F)$, kde:

- $Q$ je konečná množina vnitřních stavů,
- $\Sigma$ je konečná vstupní abeceda,
- $G$ je konečná pracovní abeceda ($\Sigma \subset G$),
- $\delta$ je zobrazení z $(Q \setminus F) \times G$ do $Q \times G \times \{-1, 1\}$, je to tedy přechodová funkce, která určuje chování stroje. V závislosti na stavu, ve kterém se stroj právě nachází. Určuje do kterého nového stavu má stroj přejít, jaký symbol má zapsat na místo původního a kam má posunout čtecí hlavu.
- $q_0 \in Q$ je počáteční stav,
- $B$ je prázdný symbol ($Blank, B \in G \setminus \Sigma$),
- $F \subseteq Q$ je množina koncových stavů.

### 3.1.2 Modifikace TS

Existuje mnoho druhů modifikací TS, avšak podstatné zůstává, že veškeré modifikace zachovávají stejnou výpočetní sílu původního TS. Mezi jednoduché modifikace například patří přidání 0 do množiny posunů, takže pak nový TS má na výběr posunout se vlevo, vpravo a nebo zůstat stát na místě. Jednoduchý neformální důkaz je takový, že nově vytvořená instrukce pro TS pouze slučuje dvě instrukce, které posunou jedním směrem tam a zas zpět. Proto jediná změna je, že se musí vykonat méně instrukcí k dosažení stejného cíle. Proč bychom toto potřebovali dělat, když už to základní TS umí, odpovědí je například pro přehlednost. Další z modifikací jsou například:

TS s posunem o n míst na pásce,

vícepáskový TS,

TS s více abecedami,

TS s více hlavami, atd.

Důkazy k modifikacím výše lze nalézt v práci [8].

## 3.2 Univerzální TS

Abychom vytvořili univerzální TS je potřeba vytvořit jednoznačné kódování. Kódování je možné realizovat následujícím způsobem.

Seřaďme a očíslujme jednotlivé konstrukční prvky TS ($\Sigma$, $Q$ a $G$).

Dále zakódujme pohyb po pásce jako $d_1 = -1$ a $d_2 = 1$. Potom je možné zakódovat hodnoty přechodové funkce $\delta(q_h, x_i) = (q_j, x_k, d_m)$ do binárního řetězce $0^h10^i1^j10^k10^m$. Vzor 11 bude sloužit k oddělení jednotlivých prvků a 111 k uvození a zakončení celého kódování.

Nadále budeme označovat kód stroje $M$ jako $<M>$ a kód slova $w$ jako $<w>$. Kvůli převeditelnosti můžeme psát pouze $w$.

Znak # bude sloužit jako oddělovač pro případ, že chceme sloučit a zřetelně oddělit kód stroje od vstupu stroje.

Kvůli tomuto kódování lze sestrojit Univerzální TS, značme $U$, takový, že (jazyk přijímaný $U$) $L(U) = \{<M>\#<w> \mid M$ akceptuje slovo $w\}$. U má tři pásky. Výpočet $U$ je následující:

- $U$ ověří, zda je slovo požadovaného formátu, nevyhovující zamítá.
- Simuluje výpočet stroje $M$ pro slovo $w$.
- Na první pásce uchovává a nemění kód $M$.
- Druhá páska obsahuje kopii pásky stroje $M$ – nad ní se také provádí simulace $M$.

- Třetí páska vlastní kód aktuálního stavu $M$.
- $U$ pro $<M>\#<w>$ zamítá právě tehdy, když zamítá $M$ pro $w$.
- $U$ pro $<M>\#<w>$ přijímá právě tehdy, když přijímá $M$ pro $w$.
- $U$ pro $<M>\#<w>$ cyklí právě tehdy, když cyklí $M$ pro $w$.[7]

## 3.3 Rozhodnutelnost

**Definice 3.3.1.** Formální jazyk $L$ je **rekurzivně spočetný**, jestliže pro něj existuje TS, který všechna slova z tohoto jazyka přijímá (akceptuje). Slova, která nejsou z tohoto jazyka, pak může buď odmítat a nebo se může stroj zacyklit. Tím se liší od rekurzivního jazyka, u kterého TS vždy zastaví, to znamená, že buď akceptuje nebo zamítne, i když mu je dáno slovo z doplňku jazyka. Pokud takový TS M existuje, říkáme, že TS $M$ přijímá jazyk $L$, značíme $L(M)$.

**Tvrzení 3.3.1.** *Existuje jazyk, který není rekurzivně spočetný*

*Důkaz.* K důkazu věty se použije *Cantorova diagonální metoda*, s níž jsme se setkali v důkaze nespočetnosti množiny reálných čísel.

Mějme abecedu $\{0,1\}^*$. Každý řetězec nad touto abecedou lze chápat jako kód nějakého Turingova stroje, nebo také jako kód slova. Pro dané slovo $x_n \in \{0,1\}^*$ označme $M_n$ Turingův stroj s kódem $x_n$ a $w_n$ slovo s kódem $x_n$. Abychom mohli pokračovat musíme seřadit řetězce z $\{0,1\}^*$. (To lze podle délky a následně s pravidlem, že 0 předchází 1. Lze vytvořit tabulku a zapsat 1 pokud stroj $M_m$ akceptuje slovo $w_w$, 0 v opačném případě. Podle tabulky lze

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $w_5$ | $w_6$ | $\dots$ |
|---|---|---|---|---|---|---|---|
| $M_1$ | **1** | 0 | 1 | 1 | 0 | 1 | $\dots$ |
| $M_2$ | 0 | **1** | 1 | 1 | 0 | 1 | $\dots$ |
| $M_3$ | 1 | 1 | **0** | 0 | 1 | 1 | $\dots$ |
| $M_4$ | 0 | 0 | 0 | **1** | 0 | 1 | $\dots$ |
| $M_5$ | 1 | 0 | 0 | 1 | **0** | 0 | $\dots$ |
| $M_6$ | 0 | 0 | 0 | 1 | 1 | **0** | $\dots$ |
| $\vdots$ | | | $\vdots$ | | | | $\ddots$ |
| $M_d$ | 1 | 1 | 0 | 1 | 0 | 0 | $\dots$ |
| $M_{d'}$ | 0 | 0 | 1 | 0 | 1 | 1 | $\dots$ |
| $\vdots$ | | | $\vdots$ | | | | $\ddots$ |

následně konstruovat diagonálu a z ní získat jazyk $D'$. Takže jazyk $D'$ bude obsahovat taková slova $w_{d'}$, která nejsou akceptovaná stroji $M_{d'}$. Tím máme jazyk, který nepřijímá žádný Turingův stroj. Protože kdyby přijímal, je zde spor:

Jazyk $D'$ je akceptován strojem $M_m$, pak tedy akceptuje slovo $w_m$, jestliže je obsaženo v $D'$, pak to ale znamená, že $M_m$ neakceptuje $w_m$, podle pravidel vzniku $D'$, což je spor.

Naopak $D'$ je akceptován, $M_m$ neakceptuje slovo $w_m$, jelikož není obsaženo v $D'$, ale to znamená, že je obsaženo v $D'$, jelikož se zde nacházejí slova, která $M_m$ neakceptuje, takže také spor.

To znamená, že jazyk $D'$ akceptován nějakým $M_m$, jestliže obsahuje slovo $w_m$ právě tehdy, když jej neobsahuje, a naopak.

V tomto případě máme neplatné předpoklady, že všechny jazyky jsou rekurzivně spočetné.

□

## 3.4 Problém zastavení

S pomocí kódování a předchozího tvrzení můžeme řešit problém zastavení.

**Definice 3.4.1.** Problém zastavení je definován jako problém rozhodnout, zda daný TS $M$ nad daným vstupem $w$ zastaví. Tomuto problému odpovídá jazyk

$$PZ = \{<M>\#<w> \mid \text{výpočet } M \text{ na } w \text{ je konečný}\}$$

**Věta 3.4.1.** *Problém zastavení není rozhodnutelný.*

Ukážeme si, že jazyk PZ nelze přijímat TS.

*Důkaz.* Provedeme opět sporem. Předpokládáme rozhodnutelnost $PZ$, tedy existenci TS $T$ takového, že $L(T) = PZ$.

Zkonstruujeme stroj $N$, který pro vstup $x$ pracuje takto:

Simuluje výpočet stroje $T$ na vstupu se svým kódem $<N>$ pro vstup $x$

$N$ akceptuje právě tehdy, když $T$ zamítne vstup $<N>\#<x>$

$N$ zamítne právě tehdy, když $T$ akceptuje vstup $<N>\#<x>$

Stroj $N$ nad vstupem pak nechá stroj $T$ simulovat $<N>\#<x>$.

$T$ zamítne právě tehdy, když akceptuje, a akceptuje právě tehdy, když zamítne. Dostáváme tedy spor, tudíž stroj $T$ nemůže existovat.

Pokud stroj $T$ nemůže existovat, pak ani nelze přijímat Jazyk $PZ$ a tedy ani nelze rozhodnout s pomocí TS problém $PZ$. □

Kapitola 4

# Srovnání

Čtenář si mohl povšimnout, že v každé kapitole byla alespoň jedna tabulka a v každé byla zvýrazněná diagonála, na které byl postaven důkaz nebo dané tvrzení. To poukazuje na první nejvýraznější podobnost. Budeme zjišťovat, jestli konstrukce této diagonály je stejná, nebo zda to, že zde vidíme diagonálu, je pouze náhoda.

Zdá se, že by nám zde mohlo pomoci diagonální lemma, ale bohužel diagonální lemma vypovídá pouze o formulích, to znamená, že ke Gödelově větě se dokonale hodí, ale už s Cantorovou větou je to složitější, nemluvě o Turingových strojích.

Diagonální lemma zhruba tvrdí, že ke každé formuli $\varphi(x)$ s jednou volnou proměnnou jazyka, ve kterém je obsažena základní aritmetika, existuje formule $\psi$, která nese význam „mám vlastnost $\varphi$“. Připomeňme kapitolu *Gödelova věta*, ze které víme, že má smysl pro každou formuli $\varphi$ uvažovat číslo $\overline{\varphi}$, které k dané formuli jednoznačně patří.

Přesné znění diagonálního lemma lze nalézt ve studijním textu [9].

**Definice 4.0.1** (Diagonalizace)**.** Nechť *diag* je zobrazení, jehož definiční obor i obor hodnot náleží $\mathbb{N}$. Pak existuje formule $Diag(x, y)$ taková, že pro každé $x$ z definičního oboru funkce *diag* platí

$$Diag(x, y) \equiv y = \overline{diag(x)}$$

*Důkaz.* Potřebujeme funkci $diag(n) = \psi(\overline{\psi})$, kde $\overline{\psi} = n$.

Neformálně $diag(x)$ přijímá číslo $x$, jenž je číslem formule $X$ a vrací uzavřenou formuli $N$ takovou, která vznikla dosazením $x$ do formule $X$. Takže $X(x) \equiv N$ a zároveň platí $\overline{N} = x$.

□

**Věta 4.0.1** (Diagonální lemma)**.** *Nechť $\varphi(x)$ je formule aritmetiky s jednou volnou proměnnou x. Pak existuje uzavřená formule $\psi$ taková, že*

$$\psi \equiv \varphi(\overline{\psi})$$

*Důkaz.* Nechť $\varphi(x)$ je dáno. Funkce *diag* přiřazuje každé formuli s jednou volnou proměnnou $\delta(x)$ uzavřenou formuli $\delta(\overline{\delta})$. ($\delta(x)$ i $\delta(\overline{\delta})$ jsou čísla!) Díky diagonalizaci existuje formule $Diag(x, y)$ taková, že

$$Diag(\delta, y) \equiv y = \overline{diag(\delta)}$$

Pak pro každé $\delta$ (formuli s jednou volnou proměnnou) položme

$$A(x) \equiv (\exists v)(Diag(x, v) \wedge \varphi(v))$$

a $\psi \equiv diag(A)$. Pak lemma lze dokázat následující ekvivalencí:

$$\psi \equiv A(\overline{A}) \equiv (\exists v)(Diag(\overline{A}, v) \wedge \varphi(v)) \equiv (\exists v)(v = \overline{diag(A)} \wedge \varphi(v)) \equiv$$

$$\varphi(\overline{diag(A)}) \equiv \varphi(\overline{A(\overline{A})}) \equiv \varphi(\overline{\psi})$$

□

**Tvrzení 4.0.2.** *Pokud v teorii $\mathcal{T}$ je reprezentovatelná diag pro jakoukoli formuli $B(x)$ s jednou volnou proměnnou $x$, pak $\mathcal{T}$ obsahuje formuli $G$ takovou, že $G \Leftrightarrow B(\overline{G})$*

*Důkaz.* Předpokládejme, že $D$ reprezentuje *diag* v $\mathcal{T}$. Nechť $F$ je formule $(\exists y)(D(x, y) \wedge B(y))$. Vyberme $G \equiv (\exists x)(x = \overline{F} \wedge F)$ za diagonalizaci formule $F$ a pak $n = \overline{F}$ a $g = \overline{G}$, budou Gödelovými čísly příslušných formulí. Mimochodem podle definice diag víme, že platí $diag(\overline{F} = \overline{G})$ a proto musí být $D(n, g)$ reprezentovatelná v $\mathcal{T}$.

Dále $G \equiv (\exists x)(x = n \wedge (\exists y)(D(x, y) \wedge B(y)))$ je to samé, jako $(\exists y)(D(n, y) \wedge B(y))$. Kvůli tomu jak *diag* funguje a správnosti $D(n, g)$ se dá také zapsat takto: $D(n, g) \wedge B(g)$ což musí být ekvivalentní $B(g)$.

Proto $G$ je logicky ekvivalentní s $B(\overline{G})$ v teorii $\mathcal{T}$ a proto platí $G \Leftrightarrow B(\overline{G})$. □

Intuitivně už je vidět směr, kterým bychom chtěli použít diagonální lemma v našich třech problémech. Jenže komplikace nastává v tom, že každý z problémů hovoří o trochu jiných objektech. Turingovy stroje pro problém zastavení, množiny pro Cantorovu větu, pouze pro Gödelovu větu to je snadnější, jelikož ta hovoří také o formulích.

Při správném nastavení $\varphi(x)$ dostaneme $\psi$ jako číslo Gödelovy formule $G$.

Přímočaře lze vzít formuli $N(x)$ s jednou volnou proměnnou, která tvrdí: „Formule s číslem $x$ není dokazatelná“ a dosadit ji za $\varphi(x)$.

$$\varphi(x) \equiv N(x)$$

Tím dostaneme výsledek rovnou $\psi$, jež zde je Gödelovým číslem formule $N(\psi)$, tedy formule, která tvrdí: „Formule s číslem $\psi$ (tedy tato formule) není dokazatelná.“ A to je přesně to samé, co tvrdí slavná Gödelova formule $G$.

## 4.1 Od Cantora přes Richarda ke Gödelově větě

Zde si ukážeme alternativní způsob odvození Gödelovy věty za použití Cantorovy věty, Richardova paradoxu. Idea postupu čerpá inspiraci ze článku [10].

Pokud je tedy možné odvodit Gödelovu větu pomocí Cantorovy věty, pak jistě tyto problémy spolu musí souviset. Pokud totiž lze použít Cantorovu větu jako nástroj k odvození Gödelovy věty, pak mají kus jádra svého problému totožný.

Začneme tím, že si připomeneme Richardův paradox definující Richardovské číslo $g$. Řešením tohoto paradoxu, jak na konci článku sám Richard uvádí, je, že „definice“ $g$ není definicí, protože trpí nekonečnou závislostí. Pro vlastnost být Richardovským, nemůže být Richardovo číslo definováno, přestože pro ostatní vlastnosti může. To znamená, že máme na výběr, buď pokračovat v jazyce o úroveň výše, anebo přijmout jistá omezení tohoto jazyka a pokračovat v něm, i když budou chybět pravdivostní ohodnocení pro některé případy.

Richardův krok lze použít bezprostředně s Cantorovou diagonální metodou, která je odvozena hned z Cantorovy věty. Ta dokazuje, že množina má menší mohutnost než její potenční množina. Postupujeme sporem. Předpokládáme, že existuje funkce $f$, která je bijekce, definiční obor je $S$ a obor hodnot je potenční množina množiny $S$. Tedy každému $x \in S$ funkce $f$ přiřadí podmnožinu $f(x) \subseteq S$. Definujme podmnožinu $T \subseteq S$ tak, že:

$$x \in T \Leftrightarrow x \notin f(x).$$

Protože $f$ je bijekce, musí existovat prvek $n \in S$ tak, že $T = f(n)$. Kontradikce:

$$n \in f(n) \Leftrightarrow n \notin f(n).$$

*Poznámka* 4.1.1. Jestliže $S$ bude množina všech množin a $f$ zvolíme jako identitu, dostáváme Russelův paradox. Definujme podmnožinu $T \subseteq S$ tak, že:

$$x \in T \Leftrightarrow x \notin x.$$

Paradox pak spočívá v tom, že se ptáme, jak to je s množinou $T$. Dostáváme odpověď

$$T \in T \Leftrightarrow T \notin T.$$

Pak lze Richardův nápad vyjádřit jako funkci $f$, která každému číslu $x$ (a číslo $x$ zároveň odpovídá nějaké vlastnosti) přiřazuje množinu všech čísel, které mají tuto vlastnost.

Pak existuje množina $T$, pro kterou platí

$$x \in T \Leftrightarrow x \text{ je Richardovské.}$$

Toto vysvětlení je sice nejvíce názorné, ale stále trpí tím nedostatkem, že funkce $f$ není dobře definovaná, stále se používá „být Richardovský“, jež bylo vysvětleno, že nelze ani formálně popsat.

Formálnější přizpůsobení lze vytvořit tímto způsobem. Začneme tak, že $S$ bude množina přirozených čísel, které reprezentují tvrzení o množinách přirozených čísel, a množina všech formulí s pouze jednou volnou proměnnou: $\phi_0(v), \ldots, \phi_n(v), \ldots$, jejichž čísla právě obsahuje $S$. Můžeme pak definovat $T$:

$$(1) \quad n \in T \Leftrightarrow n \notin \text{ množina definovaná pomocí } \phi_n(v),$$

kde $n$ je proměnná z přirozených čísel. Pak $T$ nemůže být definovaná žádnou formulí z $\phi(v)$. Je to podobný problém jako Richardův, když tvrdíme, že (1) je definicí ve stejném jazyce. Pro přiblížení pravé strany (1), v pravdivostních hodnotách dostáváme:

$$n \in \text{ množina definovaná pomocí } \phi_n(v),$$

když $\phi_n(v)$ platí o čísle $n$. Je to ekvivalentní k tvrzení, že $\phi_n(\overline{n})$ platí, kde $\overline{n}$ reprezentuje pojmenování $n$ ve formálním jazyce. A pokud je množina definovatelná pomocí $\phi_m(v)$, pak $\phi_m(\overline{n})$ je vyjádřením levé strany (1). Proto předpoklad, že množina je definovatelná v rámci formálního jazyka, vede k $m$ takovému, že pro všechna $n$ je následující tvrzení pravdivé:

$$\phi_m(n) \Leftrightarrow \neg\text{Platí}(\overline{\phi_n(\overline{n})}),$$

kde $\overline{\phi_n(\overline{n})}$ popisuje tvrzení v uvozovkách. Pak pro $n = m$ dostáváme:

$$(2) \quad \phi_m(m) \Leftrightarrow \neg\text{Platí}(\overline{\phi_m(\overline{m})})$$

Takto dostáváme tvrzení podobné paradoxu lháře. Nicméně problém se sémantickými paradoxy je použití slova, „definuje“ u Richarda, „platí“ (reps. neplatí) pro lháře. Gödela napadlo použít namísto toho slovo „dokazatelné“, které se již v daném jazyce dá definovat, pokud máme dostatečně silný jazyk. A od roku 1931 byl formální jazyk natolik silný, aby bylo toto tvrzení možné vytvořit. Při nahrazení ve (2) „platí“ za „dokazatelné“ dostaneme:

$$(3) \quad \phi_m(m) \Leftrightarrow \neg\text{Dokazatelné}(\overline{\phi_m(\overline{m})})$$

V případě tvrzení (2) je to tak, že oba předpoklady, že buď je toto tvrzení pravdivé nebo nepravdivé, vedou ke kontradikci. Zatímco v případě tvrzení (3):

1. Pokud platí, vede ke kontradikci: Jestliže $\phi_m(\overline{m})$ je dokazatelné, pak platí, jelikož „Dokazatelným“ rozumíme, že existuje důkaz a tedy lze dokázat platnost tvrzení. Pokud platí levá strana musí platit i pravá a máme kontradikci.

2. Pokud $\neg\phi_m(\overline{m})$ je dokazatelné (jakožto negací pro (3)), pak platí, protože negace na pravé straně také musí platit. Tím je myšleno, že platí Dokazatelné$(\overline{\phi_m(\overline{m})})$ a tedy $\phi_m(\overline{m})$ také platí.

Takto jsme dostali tvrzení které platí a zároveň platí i jeho negace. Jenže to nevede ke kontradikci. Pouze k tomu, že $\phi_m(\overline{m})$ je nerozhodnutelná formule. Konstrukce pravé strany (3) vyžaduje skladbu jazyka, důkazy tvrzení být definovatelný v rámci jazyka a aritmetizaci, která přidělí čísla formulím.

Takto jsme viděli, které části spolu souvisí. Cantorova diagonální metoda vyplývá z Cantorovy věty, poté je použita s myšlenkou Richardova paradoxu a paradoxu lháře tak, abychom zkonstruovali Gödelovu formuli $G$.

## 4.2 Problém zastavení TS

Změňme trochu postup dokazování sporu v důkazu zastavení, aby bylo lépe vidět, se kterým problémem je tento příbuzný. Inspirací byl text [11].

Za vstupní abecedu zvolíme tu samou, do které kódujeme univerzální TS, tedy $\sum = \{0, 1\}^*$. Stále pro spor předpokládáme, že jsme schopni rozhodnout $PZ$, jenž odpovídá „ano“, pokud stroj zastaví a naopak pokud nezastaví.

Pak stroj $N$ bude vypadat následovně: Nechť je stroj $N$ chovající se jako opak Stroje přijímající $PZ$.

Zastavující, když $PZ$ rozhodne, že program cyklí.

Cyklící, když $PZ$ rozhodne, že program zastaví.

dostáváme spor, pokud předpokládáme, že $PZ$ je rozhodnutelný stroj $N$ pro vstup $< N >$ zastavuje právě tehdy, když cyklí.

| | $<M_1>$ | $<M_2>$ | $<M_3>$ | $<M_4>$ | $<M_5>$ | ... | $<N>$ | ... |
|---|---|---|---|---|---|---|---|---|
| $M_1$ | **1** | 0 | 1 | 1 | 0 | ... | 0 | ... |
| $M_2$ | 0 | **1** | 1 | 1 | 0 | ... | 1 | ... |
| $M_3$ | 1 | 1 | **0** | 0 | 1 | ... | 1 | ... |
| $M_4$ | 0 | 0 | 0 | **1** | 0 | ... | 0 | ... |
| $M_5$ | 1 | 0 | 0 | 1 | **0** | ... | 0 | ... |
| $\vdots$ | | | $\vdots$ | | | | | $\ddots$ |
| $T$ | 1 | 1 | 0 | 1 | 0 | ... | ??? | ... |
| $N$ | 0 | 0 | 1 | 0 | 1 | ... | ??? | ... |
| $\vdots$ | | | $\vdots$ | | | | | $\ddots$ |

Tabulka 4.1: Diagonální metoda a Turingův stroj

Pak stroj $N$, jenž stojí na tom, že obsahuje $T$, který umí rozhodnout zda $N$ zastaví, nemůže existovat.

Vidíme, že stroj $N$ je jakýmsi lhářem, kdy tvrdí vždy opak výsledku jiného stroje, v tomto případě stroje $T$. tímto máme první podobnost, nicméně pro další podobnost, bych srovnal Univerzální Turingův stroj $N$ s Gödelovo formulí $G$. Podobnost se nachází ve vlastnosti, že obsahuje negaci, tu přirovnáváme

ke vlastnosti lháře, již zmíněnou ve stroji $N$. Dále stroj $N$ přijímá vstup, v konstrukci $G$ jsme použili formuli s volnou proměnnou.

Klíčový moment v důkazu mají také stejný, kdy $N$ má přijmout za vstup vlastní kód. Zatímco $G$ vznikla způsobem, kdy do formule bylo vloženo její číslo. Pro představu Turingův stroj funguje jako funkce s jednou proměnnou, popřípadě formule s jednou volnou proměnnou, o které vypovídá.

Důležité je, aby TS $U$ a formule $F$ měly stejnou mohutnost. Víme, že jich je obou spočetně – TS $U$ z lexikografického uspořádání jejich kódu, formulí $F$ zase podle Gödelova čísla. Prohlásíme tedy množiny všech TS $U$ a všech formulí $F$ za ekvivalentní.

# Závěr

Cílem této práce bylo seznámit se třemi zmíněnými větami z teorie množin, matematické logiky a informatiky, prostudovat jejich důkazy a vzájemně je srovnat. Bylo třeba se seznámit s danou problematikou, naučit se zacházet s důkazovými prostředky a proniknout k jádru důkazů. Ukázalo se, že se skutečně v jádře jedná o podobnou metodu, i když je v jednotlivých případech rozdílně použitá.

Přínosem bylo objevení mechanizmů, na kterých je založené fungování zadaných problémů. Všechny pojednávané problémy využívají nekonečných struktur, a proto hraje zásadní roli využitá definice pro jejich porovnání. První předpoklad je založen na vlastnosti, která je zaručena v Cantorově teorii množin, že každé dvě nekonečné spočetné množiny mají stejný počet prvků. Další vlastností kterou předpokládáme je, že pro dvě množiny stejné nekonečné mohutnosti existuje diagonála. Ospravedlňujeme to předpokladem spočetnosti, tedy, že ve své podstatě konstruují pomyslný čtverec, ve kterém jsme tuto diagonálu schopni nalézt.

K použití diagonální metody jsou důležité dva předpoklady.

1. Existence zobrazení mezi dvěma spočetnými množinami, což je v případě Cantorovy věty předpoklad bijekce mezi množinou a potenční množinou. V případě Gödelovy věty to je relace $Dem(x, y)$, která nese význam, že tvrzení s číslem $y$ má důkaz s číslem $x$.

2. Možnost popsat prvky na diagonále vyčíslitelnou funkcí. (To například v Richardově paradoxu není možné, proto se jedná trochu o podvod). Právě proto je Gödelova věta tak složitá, protože musí tento problém řešit.

Ve všech třech případech se jedná o významné věty, které měly značný dopad. Pochopitelně o tom existuje velké množství literatury. Na Cantorově větě je založena nekonečná hierarchie kardinálních čísel. Gödelova věta lze zobecnit do Diagonálního lemma. Z něj lze odvodit Tarského větu o nedefinovatelnosti

pravdy a Kleeneho rekurzivní větu. Diagonální lemma je blízké Větě o pevném bodě, z níž plyne řada dalších tvrzení v algebře, analýze i diskrétní matematice. Také jsem nijak důkladně nezkoumal problém vyčíslitelnosti, který s Gödelovou větou úzce souvisí. Mým úkolem ale nebylo zabývat se pozadím a dopady těchto vět, což by mohl být úkol na celý život, ale spíše proniknout do detailu, porozumět a srovnat.

Menším osobním cílem bylo, rozšířit si obzory vědomostmi z jiných oborů. Řekl bych, že tento cíl byl úspěšně splněn. Práce celkově byla zajímavá, ale nepředpokládám, že bych v této práci pokračoval. Problémy navazující na práci již zasahují do složitějších odvětví logiky a teorie množin.

# Bibliografie


1. TRLIFAJOVÁ, Kateřina. Bolzano's Infinite Quantities. *Foundations of Science*. 2018, vol. 23, no. 4, s. 681–704.
2. BALCAR, Bohuslav; ŠTĚPÁNEK, Petr. *Teorie množin*. Academia, 2001.
3. Cantorova diagonální metoda. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2020-02-20]. Dostupné z: `https://cs.wikipedia.org/wiki/Cantorova_diagon%C3%A1ln%C3%AD_metoda`.
4. ZACH, Richard. Hilbert's Program. In: ZALTA, Edward N. (ed.). *The Stanford Encyclopedia of Philosophy* [online]. Fall 2019. Metaphysics Research Lab, Stanford University, 2019 [cit. 2020-04-10]. Dostupné z: `https://plato.stanford.edu/archives/fall2019/entries/hilbert-program/`.
5. NAGEL, Ernest; NEWMAN, James Roy; HOFSTADTER, Douglas R. *Gödelův důkaz*. Vysoké učení technické v Brně: VUTIUM, 2006. ISBN 80-214-3174-1 [126 s.].
6. CHAITIN, G.J. The berry paradox: Complex Systems and Binary Networks. In: LÓPEZ-PEÑA, R.; WAELBROECK, H.; CAPOVILLA, R.; GARCÍA-PELAYO, R.; ZERTUCHE, F. (eds.). Springer, Berlin, Heidelberg: Lecture Notes in Physics, 1995. Dostupné také z: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/cplx.6130010107`.
7. PAVLISKA, Viktor. Vyčíslitelnost a složitost 1 [online]. 2002 [cit. 2020-04-30]. Dostupné z: `https://www1.osu.cz/home/habibal/kurzy/vysl1.pdf`.
8. SKOKOVÁ, Adéla. *Různé definice Turingova stroje*. Praha, 2010. Dostupné také 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 v Praze, Matematicko-fyzikální fakulta. Vedoucí práce Jan KRAJÍČEK.

9. HÁJEK, Petr; ŠVEJDAR, Vítězslav. MATEMATICKÁ LOGIKA [online]. 1994 [cit. 2020-04-22]. Dostupné z: http://www1.cuni.cz/~svejdar/papers/mate94.pdf.

10. GAIFMAN, Haim. Naming and Diagonalization, from Cantor to Godel to Kleene. *Logic Journal of the IGPL*. 2006, roč. 14, č. 5, s. 709–728.

11. GORDON, Dov. Turing Machines, diagonalization, the halting problem, reducibility [online]. 2015 [cit. 2020-04-17]. Dostupné z: https://cs.gmu.edu/~gordon/teaching/cs600/turing_machines.pdf.