

Vyčíslitelnost a složitost výpočtů

Turingův stroj

- Jde o snadno pochopitelný model skutečného počítače s páskovou pamětí
- Lehce programovatelný
- Jednoduše lze studovat jeho vlastnosti a omezení

Turingův stroj

Základní principy Turingova stroje:

- **Páska:** Nekonečně dlouhá páska rozdělená na jednotlivé buňky, kde každá buňka může obsahovat symbol (např. 0 nebo 1) nebo může být prázdná. Páska slouží jako paměť stroje.
- **Hlava:** Pohyblivá čtecí a zapisovací hlava, která se může pohybovat doleva nebo doprava po pásce. Hlava může číst symboly z pásky, přepisovat je a provádět určité akce na základě stavu stroje.
- **Stavy:** Turingův stroj má konečný počet stavů, ve kterých se může nacházet. Každý stav představuje aktuální konfiguraci stroje.
- **Přechodová funkce:** Určuje, jak se Turingův stroj chová v závislosti na aktuálním stavu a symbolu, který právě čte. Na základě této funkce může stroj provést několik akcí: změnit symbol na pásce, posunout hlavu doleva nebo doprava a přejít do jiného stavu.

Turingův stroj

Jak Turingův stroj pracuje:

- Stroj začíná v počátečním stavu a na pásce je zapsán vstup.
- Hlava čte symbol z pásky a podle aktuálního stavu a symbolu se provede akce (např. přepis symbolu, pohyb hlavy, přechod do nového stavu).
- Tento proces pokračuje, dokud stroj nedosáhne koncového stavu nebo nekonečně neprovádí výpočty.

Rozhodnutelné a nerozhodnutelné problémy

Rozhodovací problém

- problém, kde na každou instanci odpovídáme **ANO / NE**.
- Instance = konkrétní vstup (číslo, graf, slovo, program...).
- Odpověď je vždy jednoznačná.
- Každý rozhodovací problém je reprezentován nějakým **jazykem L**.

Formálně:

- $L = \{ \langle x \rangle \mid x \text{ je instance problému a odpověď na ni je ANO} \}$
- Zbytek $(\Sigma^* \setminus L)$ jsou instance s odpovědí NE.

Rozhodovací problém

- **Řešit rozhodovací problém** znamená: umět rozhodnout, zda slovo patří do jazyka LL.
- Úloha pro **Turingův stroj / algoritmus**.
- Pokud existuje stroj, který pro každý vstup vždy správně zastaví a odpoví ANO/NE, pak je problém **rozhodnutelný**.

Rozhodovací problém

- Je dané přirozené číslo sudé?

Rozhodovací problém

- **Instance:** každé přirozené číslo $n \in \mathbb{N}$.
- **Odpověď:**
 - ANO, pokud n je sudé,
 - NE, pokud n je liché.

Rozhodovací problém

- Existuje algoritmus, který **vždy skončí**:
 - **Vstup**: číslo n .
 - Vypočti zbytek po dělení 2 .
 - **Pokud zbytek** = 0 $\rightarrow$ odpověď ANO.
 - Jinak $\rightarrow$ odpověď NE.

Rozhodovací problém

- $TS = (Q, \Sigma, \Gamma, \delta, q_0, F)$
- $Q = \{q_0, q_1, q_{\text{accept}}, q_{\text{reject}}\}$
- $\Sigma = \{1\}$
- $\Gamma = \{1, X, _ \}$
- Počáteční stav: q_0
- Konečný stav: $q_{\text{accept}}, q_{\text{reject}}$

Přechodové stavy:

- $\delta(q_0, 1) = (X, R, q_1)$
- $\delta(q_0, _) = (_, S, q_{\text{accept}})$
- $\delta(q_1, 1) = (X, R, q_0)$
- $\delta(q_1, _) = (_, S, q_{\text{reject}})$

Rozhodovací problém

- $L = \{ w0 \mid w \in \{0,1\}^* \}$ = množina všech binárních slov, která končí na 0
- $L = \{ w \in \{0,1\}^* \mid w \text{ končí na } 0 \}$

Rozhodovací problém

- Je dané slovo nad abecedou $\{a,b\}$ palindrom?

Rozhodovací problém

- Vstup: slovo w nad $\{a,b\}$.
- Nastav $i \leftarrow 1, j \leftarrow |w|$.
- Dokud $i < j$:
 - Pokud $w[i] \neq w[j] \rightarrow$ odpověď NE.
 - Jinak nastav $i \leftarrow i + 1$ a $j \leftarrow j - 1$.
- Po skončení cyklu $\rightarrow$ odpověď ANO.

Rozhodovací problém

- $TS = (Q, \Sigma, \Gamma, \delta, q_0, F)$
- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_{\text{accept}}, q_{\text{reject}}\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{a, b, X, _ \}$
- Počáteční stav: q_0
- Konečný stav: $q_{\text{accept}}, q_{\text{reject}}$
- $q_0 \rightarrow$ výběr levého nespárovaného znaku
- $q_1 \rightarrow$ hledání páru pro a (doprava a zpět vlevo)
- $q_2 \rightarrow$ hledání páru pro b (doprava a zpět vlevo)
- $q_3 \rightarrow$ návrat vlevo po spárování a
- $q_4 \rightarrow$ návrat vlevo po spárování b
- $q_{\text{accept}}, q_{\text{reject}} \rightarrow$ koncové stavy

Rozhodovací problém

- $\delta(q_0, X) = (X, R, q_0)$
- $\delta(q_0, a) = (X, R, q_1)$
- $\delta(q_0, b) = (X, R, q_2)$
- $\delta(q_0, _) = (_, S, q_{\text{accept}})$

- $\delta(q_1, a) = (a, R, q_1)$
- $\delta(q_1, b) = (b, R, q_1)$
- $\delta(q_1, X) = (X, R, q_1)$
- $\delta(q_1, _) = (_, L, q_3)$

- $\delta(q_2, a) = (a, R, q_2)$
- $\delta(q_2, b) = (b, R, q_2)$
- $\delta(q_2, X) = (X, R, q_2)$
- $\delta(q_2, _) = (_, L, q_4)$

- $\delta(q_3, X) = (X, L, q_3)$
- $\delta(q_3, a) = (X, L, q_5)$
- $\delta(q_3, b) = (b, S, q_{\text{reject}})$
- $\delta(q_3, _) = (_, R, q_0)$
- $\delta(q_4, X) = (X, L, q_4)$
- $\delta(q_4, b) = (X, L, q_5)$
- $\delta(q_4, a) = (a, S, q_{\text{reject}})$
- $\delta(q_4, _) = (_, R, q_0)$
- $\delta(q_5, X) = (X, L, q_5)$
- $\delta(q_5, a) = (a, L, q_5)$
- $\delta(q_5, b) = (b, L, q_5)$
- $\delta(q_5, _) = (_, R, q_0)$

Rozhodovací problém

- $q_0 : _ a b b a _$
– čtu $a \rightarrow$ přepíšu na X , jdu do **q_1**
- $q_1 : _ X b b a _$
– jedu doprava, na konci $_ \rightarrow$ přechod do **q_3**
- $q_3 : _ X b b a _$ (hlava na a)
– čtu a , přepíšu na X , jdu do **q_0**
- $q_0 : _ X b b X _$
– přeskočím X , čtu $b \rightarrow$ přepíšu na X , jdu do **q_2**
- $q_2 : _ X X b X _$
– jedu doprava, na konci $_ \rightarrow$ přechod do **q_4**
- $q_4 : _ X X b X _$ (hlava na b)
– čtu b , přepíšu na X , jdu do **q_0**
- $q_0 : _ X X X X _$
– všude jen X , pak $_ \rightarrow$ **q_{accept}**

Rozhodovací problém

- $L = \{ w \in \{a,b\}^* \mid w = w^R \}$

Nerozhodnutelné problémy

- Nerozhodnutelný problém je takový **rozhodovací problém**, pro který **neexistuje algoritmus**, jenž by pro všechny vstupy během konečného času určil správnou odpověď ANO / NE.

Nerozhodnutelné problémy

- **Halting problem (problém zastavení)**
„Zastaví se daný Turingův stroj M na vstupu w ?“
– Dokázáno jako nerozhodnutelné pomocí redukce.
- **Problém příslušnosti**
„Přijímá daný Turingův stroj M slovo w ?“
- **Problém prázdnosti**
„Je jazyk přijímaný Turingovým strojem M prázdný?“

Metoda redukce

- Redukce je způsob, jak **přenést obtížnost jednoho problému (A) na jiný problém (B)**.
- Pokud by se dal snadno vyřešit problém B, pak by šlo vyřešit i problém A.
- Proto – jestliže **A je nerozhodnutelný**, pak i **B musí být nerozhodnutelný**.

Metoda redukce

- **Cíl:** chceme ukázat, že problém **B** je nerozhodnutelný.
- **Známý problém A:** vybereme problém, u kterého už víme, že je nerozhodnutelný (např. problém příslušnosti nebo halting problem).
- **Redukce f:** navrhne algoritmus (tzv. redukční funkci), který:
 - vezme libovolný vstup x problému A,
 - převede ho na vstup $f(x)$ problému B,
 - tak, že odpovědi odpovídají:
 - $x \in A \iff f(x) \in B$

Metoda redukce

- **Spor:**
 - Předpokládejme, že existuje algoritmus pro problém **B**.
 - Pak bychom mohli vzít každý vstup x problému **A**, přepočítat na $f(x)$, spustit algoritmus pro **B** a dostat tím i odpověď pro **A**.
 - Tím bychom rozhodli **A**. Ale **A je nerozhodnutelný** → vzniká **spor**.
- **Závěr:** problém **B** je také nerozhodnutelný.

Metoda redukce

- **A = problém příslušnosti** („Přijímá Turingův stroj M slovo w ?“)
– víme, že je nerozhodnutelný.
- **B = halting problem** („Zastaví se M na vstupu w ?“).
- Ukážeme, že kdybychom měli algoritmus pro halting problem, mohli bychom s ním řešit i příslušnost.
- Proto je halting problem také **nerozhodnutelný**.