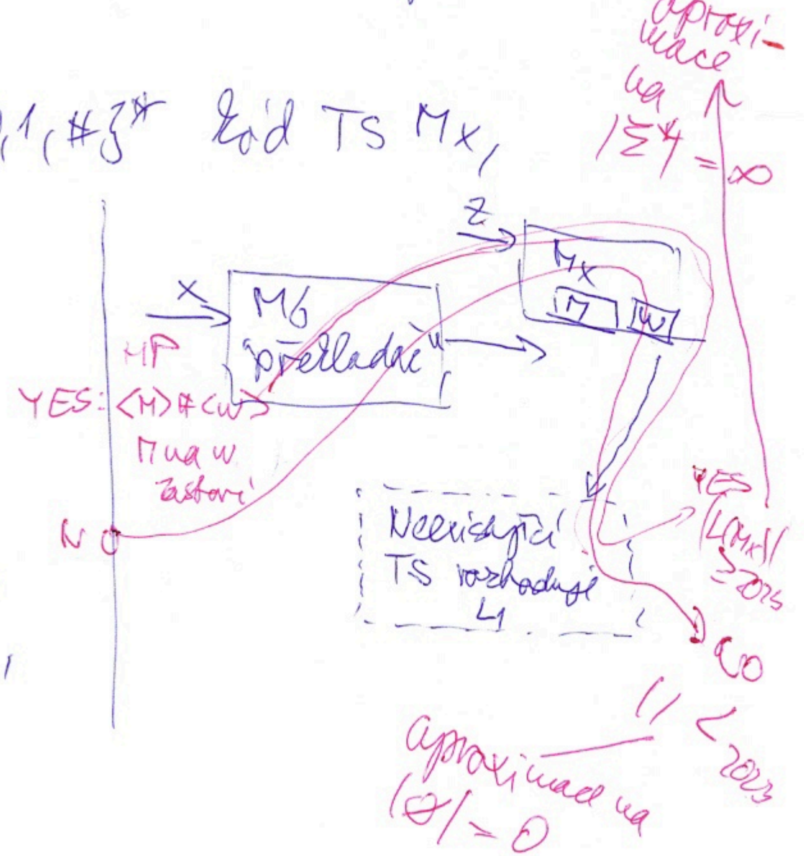
TIN, Cvičení 5 – 16/11/2023 – Nerozhodnutelnost

1. Uveďte klíčové myšlenky důkazu redukcí, že $L_1 = \{ \langle M_1 \rangle \mid M_1$ je TS takový, že $|L(M_1)| \geq 2023\} \notin REC$.

Základní myšlenky důkazu:

- Sestrojíme redukci $\sigma : \{0,1,\#\}^* \to \{0,1\}^*$ redukující HP na L_1.
- Redukce σ přiřadí každému $x \in \{0,1,\#\}^*$ kód TS M_x, který pracuje tak, že:
  1. M_x smaže svůj vstup z.
     [Poznámka: jazyk M_x je tedy vždy buď Σ^* nebo $\emptyset$.]
  2. M_x zapíše na vstup řetězec x
  3. M_x ověří, zda $x = \langle M \rangle \# \langle w \rangle$ pro TS M a vstup w. Pokud ne, zamítne (a tedy $L(M_x) = \emptyset$)
  4. Jinak M_x spustí simulaci M na w.



5. Pokud simulace doběhne, M_x přijme (a tudíž $L(M_x) = \Sigma^*$), jinak cyklí ($L(M_x) = \emptyset$).

- σ lze evidentně implementovat úplným TS
- σ zachovává členství v jazycích MP a L_1:

$\forall x \in \{0,1,\#\}^* : \sigma(x) \in L_1 \Leftrightarrow \sigma(x) = \langle M_x \rangle$, kde $L(M_x) = \Sigma^* \Leftrightarrow x = \langle M \rangle \# \langle w \rangle$, kde M je TS a w jeho vstup takový, že M na w zastaví. $\Leftrightarrow x \in MP$.

2. Uveďte základní myšlenku redukce, která dokazuje, že $L_2 = \{ \langle M_1 \rangle \# \langle M_2 \rangle \mid M_1$ a M_2 jsou TS takové, že $L(M_1) \cap L(M_2) \neq \emptyset \} \notin REC$.

- Redukce $\sigma : \{0,1,\#\}^* \to \{0,1,\#\}^*$ redukující MP na L_2 musí přiřadit každému $x \in \{0,1,\#\}^*$ řetězec $\langle M_x^1 \rangle \# \langle M_x^2 \rangle$, kde M_x^1 a M_x^2 jsou TS, které pracují následovně:
  - M_x^1 je TS, který vyprodukuje redukce z příkladu 1.
  - M_x^2 je TS, který vždy přijme cokoliv ($\to \circledcirc$) a jeho jazyk je tedy Σ^*

- Tato redukce je zřejmě implementovatelná ÚTS.
- σ zachovává členství:

  $\forall x \in \{0, 1, \#\}^* : \sigma(x) \in L_2 \iff \sigma(x) = \langle M_x^1 \rangle \# \langle M_x^2 \rangle$

  a $L(M_x^1) = \Sigma^* \iff x = \langle M \rangle \# \langle w \rangle$, kde M je TS a w jeho vstup takové, že M na w zastaví $\iff x \in HP$.

- DOPLNĚK: Je či není $L_2 \in RE$ (částečně rozh.) a proč?

ANO, $L_2 \in RE$ — k částečnému rozhodování L_2 lze užít NTS M', který
  - uhodne řetězec w,
  - odsimuluje M_1 na w, pokud zastaví, odsimuluje na w také M_2, pokud i ten zastaví, přijme; jinak cyklí.

3. Uveďte základní myšlenku redukce ukazující, že $L_3 = \{ \langle M \rangle \mid M \text{ je TS takový, že } |L(M) \cap \{a, b\}| = 1 \} \notin REC$

m

pro $\Sigma = \{a, b\}$.

$$(a \in L(M) \wedge b \notin L(M)) \\ \vee (a \notin L(M) \wedge b \in L(M))$$

- Idea redukce $\sigma : \{0,1,\#\}^* \to \{0,1\}^*$, která každému $x \in \{0,1,\#\}^*$ přiřadí TS M_x, který pracuje následovně:
  - M_x ověří, zda jeho vstup $z =$ "a" a výsledek si uloží ve stavovém řízení.
  - M_x smaže vstup a zapíše na něj x.
  - M_x posoudí, zda $x = \langle M \rangle \# \langle w \rangle$ pro TS M a vstup w. Pokud ne, odmítne ($L(M_x) = \emptyset$).
  - Jinak spustí simulaci M na w.
  - Pokud ta doběhne, pak přijme, pokud vstup z byl "a" i jinak odmítne ($L(M_x) = \{a\}$).
  - Jinak cyklí — a tudíž $L(M_x) = \emptyset$.
- σ je implementovatelná ÚTS a zachová členství v jazyce:

$\forall x \in \{0,1,\#\}^* : \sigma(x) \in L_3 \iff \sigma(x) = \langle M_x \rangle$, kde $L(M_x) = \{a\}$
$\iff x = \langle M \rangle \# \langle w \rangle$, kde TS M zastaví na $w \iff x \in HP$.

DOPLNĚK: Je či není $L_3 \in RE$? Proč?

NE: – lze ukázat redukcí např. z problému co-HP.

– redukce $\sigma: \{0,1,\#\}^* \to \{0,1\}^*$ přiřadí každému $x \in \{0,1,\#\}^*$ TS M_x, který:

– M_x ověří, zda jeho vstup $z =$ "a". Pokud ano, přijme. (Pak $L(M_x) \supseteq \{a\}$.)

– M_x smaže vstup, zapíše na něj x a ověří, zda $x = \langle M \rangle \# \langle w \rangle$ pro TS M a vstup w. Pokud ne, přijme. ($L(M_x) = \Sigma^*$)

– Jinak spustí simulaci M na w.

– Pokud simulace doběhne, pak přijme ($L(M_x) = \Sigma^*$). Jinak cyklí ($L(M_x) = \{a\}$).

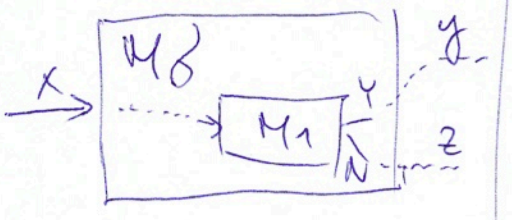
– Zachování členství – DOMA.

4. Dokažte nebo vyvraťte následující tvrzení:

$\forall \Sigma \; \forall L_1, L_2 \subseteq \Sigma^* : L_1, L_2 \in REC \Rightarrow (L_1 \leq L_2)$

implikace ↑ … ↑ redukce.

Hypotéza: Platí — zkusme dokázat.



- $L_1, L_2 \in REC$ a tedy $\exists$ úplné TS M_1 a M_2 rozhodující L_1 a L_2.
- Redukci σ z L_1 na L_2 lze pak sestrojit tak, že stroj M_σ, implementující redukci σ, pro vstup x pomocí TS M_1 zpracuje, zda $x \in L_1$. Pokud ano, pak M_σ jednoduše vypíše jeden, konstantně zvolený řetězec $y \in L_2$. Jinak M_σ vypíše konstantně zvolený řetězec $z \notin L_2$.

ALE POZOR! Co když $L_2 = \emptyset$ nebo $L_2 = \Sigma^*$?
V tom případě nelze najít buď y nebo z!

- Tvrzení lze upravit do platné podoby tak, že řekneme, že $\emptyset \subset L_1, L_2 \subset \Sigma^*$ nebo $L_1 = L_2$.
- Pokud $\emptyset \subset L_1, L_2 \subset \Sigma^*$, pak výše uvedený důkaz lze použít!

$\emptyset$ se redukuje na $\emptyset$
Σ^* — " — Σ^*

5. <u>S využitím diagonalizace dokažte, že $\mathcal{L}_1 \subset REC$.</u>

LOA úplné TS

- Uvažme jazyk LOA $= \{ w \in \{0,1\}^* \mid w$ je kód LOA nad $\Sigma = \{0,1,\#\}$ používajících "#" jako ohraničovač použitelné části pásky $\}$

- Dále uvažme jazyk $L = \{ w \in LOA \mid w\# \notin L(M_w) \}$, kde M_w je LOA s kódem w.

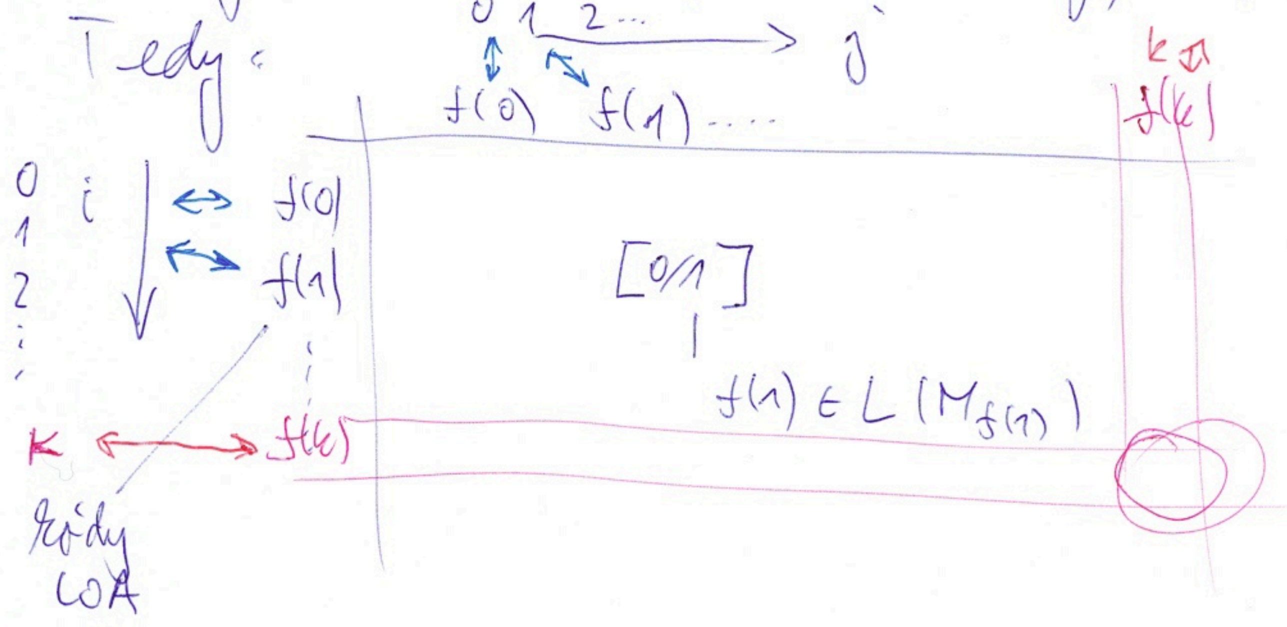
- Ukážeme, že $L \in REC$ a $L \notin \mathcal{L}_1$. Druhou část ukážeme pomocí diagonalizace.

- Množina LOA je spočetná – konečně je nekonečná a současně $LOA \subseteq \{0,1\}^*$, kde $\{0,1\}^*$ je spočetná – její řetězce lze uspořádat lexikograficky.

- Tedy existuje bijekce $f : \mathbb{N} \leftrightarrow LOA$

- zavedeme notaci $M_{f(n)}$ pro $n \in \mathbb{N}$, jež bude reprezentovat LOA s kódem $f(n)$ – tedy $M_{f(n)}$ je "n"-tý LOA.

- Sestavíme matici $A: N \times N \to \{0,1\}$, kde
  $\forall i,j \in N: A[i,j] = 1 \Leftrightarrow f(j) \in L(M_{f(i)})$
  Tedy:



- Ukážeme sporem, že $L \notin \mathcal{L}_1$.
- Předpokládejme, že $L \in \mathcal{L}_1$. Pak musí existovat LOA, který L přijímá. Tento LOA se musí nacházet na nějakém řádku/sloupci k.
- Zkoumejme $A[k,k]$:
  - Pokud $A[k,k] = 1$, pak z definice $A[\,]$ plyne, že $f(k) \in L(M_{f(k)})$. Pak ale $f(k) \notin L$.
    Ale současně jsme věděli, že $M_{f(k)}$ L přijímá, tedy $L(M_{f(k)}) = L$. Z definice L ale

plyne, že $\boxed{f(u) \notin L(M_{f(u)})}$. SPOR.

- Analogicky pro $A[u,u] = 0$. Pak z definice $A[\,]$ plyne, že $\boxed{f(u) \notin L(M_{f(u)})}$. Pak ale $f(u) \in L$. Současně víme, že $L = L(M_{f(u)})$. A tedy $\boxed{f(u) \in L(M_{f(u)})}$. SPOR.

- Žádný třetí případ není a tedy neplatí předpoklad a opravdu platí $L \notin \mathcal{L}_1$.

- Zbývá ukázat, že $L \in REC$. Ukážeme tak, že L lze přijmout úplným TSM, který pracuje následovně:
  1. M posoudí, zda na vstupu má správně zformovaný kód w nějakého LOA.
  2. M si spočte, kolik kroků může LOA s kódem w učinit na pásce délky $|w|$ aniž by cyklil — viz přednášky.
  3. M odsimuluje max $|w|$ kroků LOA s kódem

w na vstupu w. Pokud by tento LOA přijal i přijme i jinak odmítne.

⌈POZNÁMKA: Konstrukce v bodech 2 a 3 výše je podobná jako zúplnění LOA na přednášce – tam se ale používá speciální číselná soustava tak, aby se počet kroků dal umístit v mezích původního vstupu. Tato abeceda bude jiná pro každý LOA (resp. může být jiná). To je možné při individuálním zúplnění, ale nelze to realizovat v rámci jednoho LOA, který by měl přijímat jazyk L převzatý v důkazu $L \notin \mathcal{L}_1$. Jeden LOA má jednu abecedu. ⌋