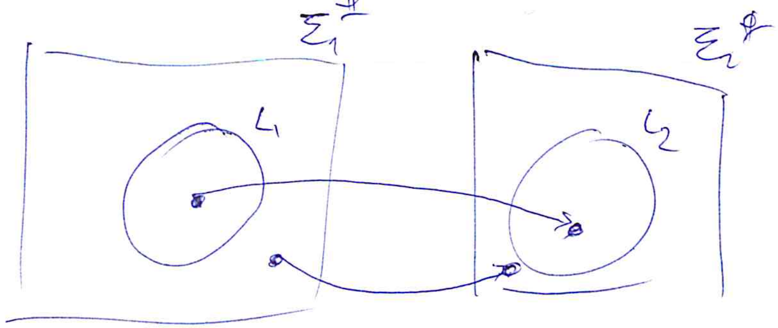
Nerozhodnutelnost, redukce



- Mějme $L_1 \subseteq \Sigma_1^*$ a $L_2 \subseteq \Sigma_2^*$,
  redukce $\sigma : \Sigma_1^* \to \Sigma_2^*$ takovou, že:
  1. zachovává členství v jazycích, neboli $\forall w \in \Sigma_1^* : w \in L_1 \Leftrightarrow \sigma(w) \in L_2$,
  2. σ je totální f-ce vyčíslitelná úplným TS.
- Rozhodnutelnost / částečná rozh.

REC   RE $= \mathcal{L}_0$

P_1: problém / jazyk, jehož status REC / RE ?

$P_1 \xrightarrow{\sigma} P_2$ ✓

P_2 známý problém / známý jazyk $\in$ REC / RE

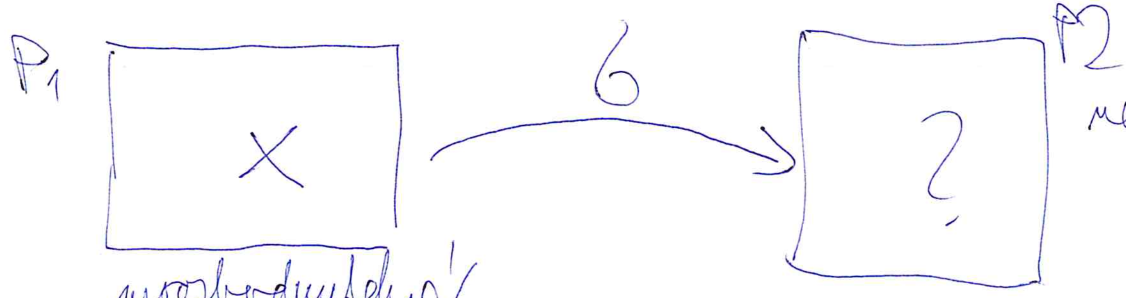
přesněji: problém, jehož status rozhodnutelnosti / část. rozh. je známý!

T_σ ; T – rozhoduje P_2 (část. rozh.)

└ TS, který vyčísluje σ

TS, který rozhoduje / část. rozh. P_1

- Nerozhodnutelnost / ani částečná rozh.



neznámý status rozhodu.

nerozhodnutelný / ani částečně rozh.

- ukazuji nerozh. / ani část. rozhodnutelnost.
- kdyby tomu tak nebylo, pak známe T_σ i T_{P2} a máme rozhodovací / část. rozhodovací proceduru pro P1 : SPOR

- Dokažte, že problém neprázdnosti jazyka TS není rozhodnutelný!

<u>Důkaz redukcí z problému zastavení:</u>

- Problém neprázdnosti lze charakterizovat jazykem
  $NEP = \{\langle M \rangle \mid M \text{ je TS takový, že } L(M) \neq \emptyset\} \subseteq \{0,1\}^*$
- Problém zastavení je charakterizován jazykem
  $HP = \{\langle M \rangle \# \langle w \rangle \mid M \text{ je TS, který zastaví na } w\} \subseteq \{0,1,\#\}^*$.
- Sestavíme redukci $\sigma: \{0,1,\#\}^* \rightarrow \{0,1\}^*$ z jazyka HP na NEP.

- TS implementující δ přiřadí každému vstupu $x \in \{0,1,\#\}^*$ řetězec $\langle M_x \rangle$, kde M_x je TS, který na vstupu $y \in \{0,1\}^*$ pracuje následovně:
  1. M_x smaže svůj vstup y.
  2. Zapíše na pásku řetězec x.
  3. M_x posoudí, zda $x = x1 \# x2$ pro $x1$, který je kódem TS, a x_2, které je kódem jeho vstupu. Pokud ne, odmítne.
  4. Jinak M_x simuluje činnost TS s kódem x_1 na řetězci s kódem x_2. Pokud simulace skončí, přijme (jinak cyklí).
- M_δ lze implementovat úplným TS. Konkrétně tento TS vypíše kód M_x, který sestává ze 4 komponent, které odpovídají výše uvedeným krokům. Tři z nich jsou přitom konstantní (nezávisí na x) – konkrétně (1) smazání pásky, (2) test na dobře zformovanou instanci MP (členství v reg. jazyce) a (3) simulace daného TS na daném vstupu (pomocí UTS). TS implementující tyto kroky, které evidentně existují, lze připravit předem a M_δ jen vypíše kód spolu s kódem pro předání řízení,

Zbývá vygenerovat kód TS, který zapíše na pásku dané x. To je ale triviální.

- Zkoumejme možné jazyky TS M_x:
  - $L(M_x) = \emptyset \iff$ (x není správně zformovaná instance MP) nebo $x = x_1 \# x_2$ a TS s kódem x_1 na řetězci s kódem x_2 neskončí).
  - $L(M_x) = \Sigma^* \iff$ x je správně zformovaná instance MP, kde $x = x_1 \# x_2$, a x_1 je kód TS, který na vstupu s kódem x_2 zastaví.
- Nyní již snadno ukážeme, že σ zachovává členství:

$\langle M_x \rangle \in NEP \iff L(M_x) = \Sigma^* \iff x = x_1 \# x_2$, kde x_1 kód TS, který zastaví na vstupu s kódem $x_2 \iff x \in HP$. $\square$

- Dokažte, že problém prázdnosti jazyka TS není ani částečně rozhodnutelný.

Idea důkazu:

- Problém prázdnosti je charakterizován jazykem EMP = $\{ \langle M \rangle \mid M$ je TS takový, že $L(M) = \emptyset \}$
- Užijeme redukci z CO-HP = $\{ \langle M \rangle \# \langle w \rangle \mid M$ je TS, který na w nezastaví $\}$

- Lze užít takřka identickou redukci, jako v předchozím případě, jen s jedinou úpravou: konkrétně, pokud x není správně zformovaná instance, pak M_x přijme (a jeho jazyk tedy bude Σ^*).

- Dokažte, že problém regularity jazyka TS <u>není ani částečně rozhodnutelný</u>!

<u>Důkaz redukcí z CO-HP</u>:

- Problém regularity lze charakterizovat jazykem
$REG = \{ \langle M \rangle \mid M \text{ je TS takový, že } L(M) \in \mathcal{L}_3 \}$
- Navrhneme redukci $\sigma : \{0, 1, \#\}^* \to \{0, 1\}^*$, která redukuje CO-HP na REG.
- TS M_σ implementující σ přiřadí vstupu $x \in \{0, 1, \#\}^*$ TS M_x, který na vstupu $y \in \{0, 1\}^*$ postupuje následovně:
  1. M_x ověří, zda $y \in \{0^n 1^n \mid n \geq 0\}$ a výsledek ano/ne si zapamatuje ve stavovém řízení.
  2. Smaže pásku a zapíše na ni x.

3. Ověří, zda x má požadovanou strukturu. Pokud ne, přijme, pokud $y \in \{0^n1^n \mid n \geq 0\}$, jinak odmítne.

4. Odsimuluje TS s kódem x_1 na vstupu s kódem x_2 ($x = x_1 \# x_2$). Pokud simulace skončí, přijme, pokud $y \in \{0^n1^n \mid n \geq 0\}$, jinak odmítne. Pokud simulace cyklí, cyklí.

- M_f lze snadno implementovat úplným TS. To lze ukázat velmi podobně jako u redukce z HP na NEP, jen je zapotřebí si uvědomit, že bude třeba dodat komponentu, která otestuje členství $y \in \{0^n1^n \mid n \geq 0\}$. Tato komponenta je konstantní (tedy nezávisí na x), uvedený test lze jednoduše implementovat TS a M_f jej vypíše kód (propojený s ostatními komponentami).

- Zkoumejme jazyk $L(M_x)$. Existují následující možnosti pro různá x:

  a) $L(M_x) = \{0^n1^n \mid n \geq 0\} \iff$ (x nemá požadovanou strukturu, nebo $x = x_1 \# x_2$ a TS s kódem x_1 na vstupu s kódem x_2 zastaví).

b) $L(M_x) = \emptyset \iff x = x_1\#x_2$ a TS s kódem x_1 na vstupu s kódem x_2 nezastaví.

- z toho již plyne zachování členství v jazyce:
$\langle M_x \rangle \in REG \iff L(M_x) = \emptyset \iff x = x_1\#x_2$ a TS s kódem x_1 na vstupu s kódem x_2 nezastaví $\iff x \in CO\text{-}HP$. □

- Dokažte, že problém neregularity jazyka TS také není ani částečně rozhodnutelný.

<u>Důkaz redukcí z CO-HP:</u>

- Jazyk vedeného problému je $NREG = \{ \langle M \rangle \mid M$ je TS takový, že $L(M) \notin \mathcal{L}_3 \}$
- sestavíme redukci $\sigma : \{0,1,\#\}^* \to \{0,1\}^*$, která redukuje CO-HP na NREG.
- TS M_σ implementující σ přiřadí vstupu $x \in \{0,1,\#\}^*$ TS M_x, který na vstupu $y \in \{0,1\}^*$ postupuje takto:
  1. M_x přetransformuje obsah pásky $\Delta y \Delta^\omega$ na $\Delta y \Delta y \Delta^\omega$.
  2. Na druhé části vstupu si ověří, zda $y \in \{0^n 1^n \mid n \geq 0\}$.
  Výsledek si zapamatuje v konečném stavovém řízení.

a druhou část vstupu smaže a nahradí za x
(což vede na počát. konfig $\triangleright y \Delta \times \Delta^{\omega}$).

3. Ověří, zda x má požadovanou strukturu; pokud ne, odmítne.
4. Odsimuluje TS s kódem x_1 na vstupu s kódem x_2 ($x = x_1 \# x_2$), ale pouze do max. $|y|$ kroků.
5. Pokud ~~simulace~~ TS s kódem x_1 v max. $|y|$ krocích skončí, odmítne.
6. Pokud TS s kódem x_1 na vstupu s kódem x_2 v max $|y|$ krocích neskončí, pak M_x přijme, pokud $y \in \{0^n 1^n \mid n \geq 0\}$, jinak odmítne.

- M_x lze evidentně implementovat úplným TS — lze argumentovat analogicky jako v předchozím případě.
- Zkoumejme $L(M_x)$ pro různá x:
  a) x nemá odpovídající strukturu $\Rightarrow$ $L(M_x) = \emptyset$
  b) $x = x_1 \# x_2$ a TS s kódem x_1 na x_2 zastaví $\Rightarrow$ Pak existuje nějaký počet kroků k, ve kterém ~~simulace~~ TS s kódem x_1 na vstupu s kódem

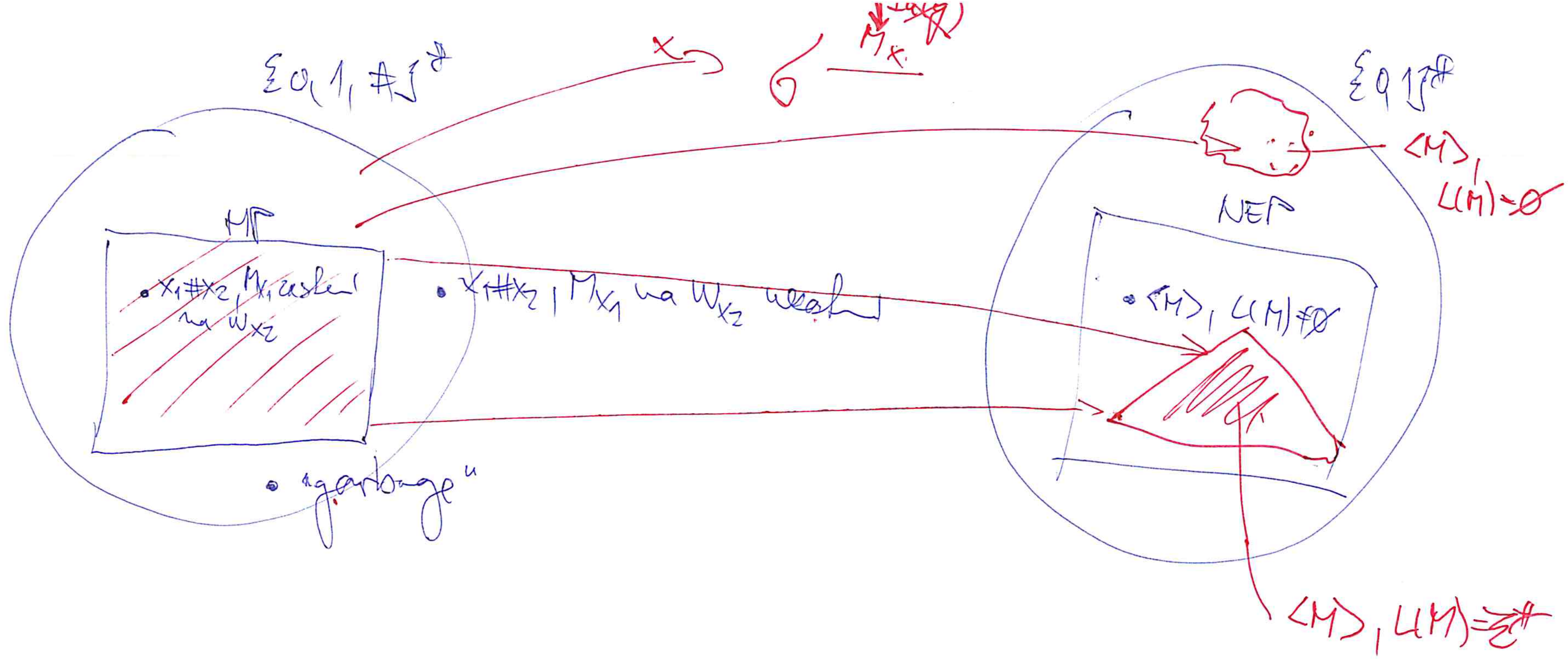
x_2 zastaví. Pak pro $|y| < k$, $y \in L(M_x)$ pokud $y \in \{0^n 1^n, n \geq 0\}$ jinak $y \notin L(M_x)$. V tomto případě tedy
$L(M_x) = \{0^n 1^n \mid 2n < k\} \in Fin$

~~Dohromady~~

c) $x = x_1 \# x_2$ a TS s kódem x_1 na x_2 nezastaví $\Rightarrow$
Pak pro y libovolné délky simulace TS s kódem x_1 na vstupu s kódem x_2 nevede ke konci běhu TS s kódem x_1. y je pak přijat, pokud $y \in \{0^n 1^n \mid n \geq 0\}$

- Můžeme tedy shrnout, že nastávají dva případy:
a) $L(M_x) \in Fin \iff x$ nemá požadovanou strukturu, nebo $x = x_1 \# x_2$ a TS s kódem x_1 na x_2 zastaví.
b) $L(M_x) = \{0^n 1^n \mid n \geq 0\} \iff x = x_1 \# x_2$ a TS s kódem x_1 na vstupu s kódem x_2 nezastaví.

- Redukce tedy zachovává členství v jazyce, neboť:
$\langle M_x \rangle \in NREG \iff L(M_x) = \{0^n 1^n \mid n \geq 0\} \iff x = x_1 \# x_2$ a TS s kódem x_1 na vstupu s kódem x_2 nezastaví
$\iff x \in CO\text{-}MP$. $\square$


$\{0,1,\#\}^*$
$x \mapsto \sigma$
M_x
$\{0,1\}^*$
$M\Gamma$
$x_1\#x_2$, M_{x_1} entscheidet W_{x_2}
$x_1\#x_2$, M_{x_1} ua W_{x_2} entscheidet
"garbage"
$\langle M\rangle$, $L(M)=\emptyset$
NEP
$\langle M\rangle$, $L(M)\neq\emptyset$
$\langle M\rangle$, $L(M)=\Sigma^*$