A

(A1) Riceova věta, důkaz pomocí m-převoditelnosti.

- Je-li $\mathcal{C}$ množina PD jazyků, pak jazyk

$$L_{\mathcal{C}} = \{\langle M \rangle \mid L(M) \in \mathcal{C}\}$$

  je rozhodnutelný pouze pokud $\mathcal{C} = \emptyset$ nebo $\mathcal{C} = \text{PD}$
- **Dk.**: $L_{\mathcal{U}} \leq_m L_{\mathcal{C}}/\overline{L_{\mathcal{C}}}$
  - pokud $\mathcal{C}$ neobsahuje prázdný jazyk, převedeme $\langle M, x \rangle$ na TS, který pro vstup y nejdřív zavolá $M(x)$ a pokud přijme, zavolá TS pro nějaký (neprázdný z předpokladu) $L \in \mathcal{C}$ na vstup y
    - pokud M přijímá x, jazykem vytvořeného TS bude $L \in \mathcal{C}$
    - pokud M nepřijímá x, jazykem vytvořeného TS bude $\emptyset \notin \mathcal{C}$
  - pokud $\mathcal{C}$ obsahuje prázdný jazyk, převedeme $\langle M, x \rangle$ na TS, který pro vstup y nejdřív zavolá $M(x)$ a pokud přijme, zavolá TS pro nějaký (neprázdný z předpokladu) $L' \notin \mathcal{C}$ na vstup y
    - pokud M přijímá x, jazykem vytvořeného TS bude $L' \notin \mathcal{C}$
    - pokud M nepřijímá x, jazykem vytvořeného TS bude $\emptyset \in \mathcal{C}$

(A2) Savičova věta.

- Pro každou funkci $f(n) \geq \log_2 n$ platí:

$$\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$$

- **Dk.**:
  - NTS M v $\mathcal{O}(f(n)) \longrightarrow$ DTS M' v $\mathcal{O}(f^2(n))$
  - technické předpoklady
    - M má jednostranně nekonečnou pracovní pásku (proč?)
    - M má jednoznačnou přijímací konfiguraci C_F
  - M' prohledá všechny konfigurace M, než najde cestu z C_0^x do C_F
    - celkem $2^{c_M f(n)}$ konfigurací
    - rekurzivní rozděl a panuj algoritmus
    - $\mathcal{O}(f(n))$ úrovní rekurze, každá potřebuje $\mathcal{O}(f(n))$ na uložení čísla konfigurace

- pokud není $f(n)$ vyčíslitelná (nebo potřebuje moc prostoru k vyčíslení)
  - iterace $f(n) = i = 1, 2, 3...$, dokud není nalezená cesta, nebo není dosažitelná žádná konfigurace, které nestačí prostor

(A3) Deterministická prostorová hierarchie.

- **Prostorově konstruovatelná** funkce $f : \mathbb{N} \to \mathbb{N}$, $f(n) \geq \log_2 n$, která zobrazuje 1^n na $f(n)$ vyčíslitelná v prostoru $\mathcal{O}(f(n))$
- Pro každou prostorově konstruovatelnou funkci f existuje jazyk A rozhodnutelný v prostoru $\mathcal{O}(f)$ ale ne v $o(f)$.
  - vytvoříme stroj D, který pracuje v prostoru $\mathcal{O}(f)$, odsimuluje libovolný stroj v pracující v $o(f)$[1] a odpoví opačně
    - kvůli asymptotice uvažujeme řetězce ve tvaru $\langle M \rangle 1^*0$ (můžeme na konec přidat $1^{n_0}0$, n_0 je z definice $o(f)$)
    - při zacyklení rejectneme po $2^{f(n)}$ krocích

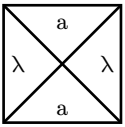
(A4) Deterministická časová hierarchie.

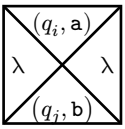
- **Časově konstruovatelná** funkce $f : \mathbb{N} \to \mathbb{N}$, $f(n) \in \Omega(n \log n)$, která zobrazuje 1^n na $f(n)$ vyčíslitelná v čase $\mathcal{O}(f(n))$
- Pro každou časově konstruovatelnou funkci f existuje jazyk A rozhodnutelný v čase $\mathcal{O}(f)$ ale ne v $o\left(\frac{f}{\log f}\right)$.
  - vytvoříme stroj D, který pracuje v čase $\mathcal{O}(f)$, odsimuluje libovolný stroj v pracující v $o\left(\frac{f}{\log f}\right)$ a odpoví opačně
    - stejný trik s $\langle M \rangle 1^*0$
    - počítání kroků trvá $\Theta(\log f)$
    - inicializace čítače na $\left\lceil \frac{f}{\log f} \right\rceil$ trvá $\mathcal{O}(f)$
    - stav simulovaného stroje a $\langle M \rangle$ se drží na druhé pásce neustále u hlavy
    - na další pásce počítadlo – posun trvá $\Theta(\log f)$ na každý krok

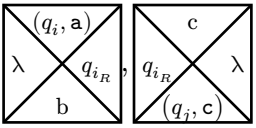
(A5) Cookova-Levinova věta (NP-úplnost SAT)

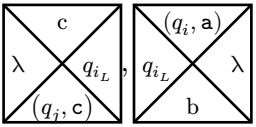
- SAT je NP-úplný
- ukážeme, že $\forall A \in \text{NP}\ A \leq_m^p \square \leq_m^p \text{SAT}$

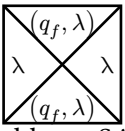
[1] Tedy specificky neodsimuluje sám sebe a tedy může existovat :)

- NP $\Longrightarrow$ NTS přijme v $p(n)$ čase
- předpoklady:
  - jednoznačný přijímací stav a konfigurace s prázdnou páskou
  - stroj neopouští vymezený prostor velikosti $p(n)$
- koupelna
  - vysoká a široká $p(n)$
  - horní strana $(q_0, x_1), x_2 ... x_n, \lambda ... \lambda$
  - dolní strana $\big(q_f, \lambda\big), \lambda ... \lambda$
  - levá i pravá $\lambda ... \lambda$
  - kachle:
    - 

$\forall \mathtt{a} \in \Sigma$
    - 

$\forall \big(q_j, \mathtt{b}, N\big) \in \delta(q_i, \mathtt{a})$
    - 

$\forall \big(q_j, \mathtt{b}, R\big) \in \delta(q_i, \mathtt{a})$
    - 

$\forall \big(q_j, \mathtt{b}, L\big) \in \delta(q_i, \mathtt{a})$
    - 

  - kachle $\longrightarrow$ SAT
    - formule pro nejvýše jednu kachli na každém políčku
    - formule pro alespoň jednu kachli na každém políčku
    - formule pro sousedění kachlí vedle sebe
    - formule pro sousedění kachlí nad sebou
    - formule pro levou, pravou, horní a dolní stěnu

B

(B1) Univerzální Turingův stroj a nerozhodnutelnost jazyka univerzálního Turingova stroje.

- kódování pomocí $\Gamma = \{\texttt{0}, \texttt{1}, \texttt{L}, \texttt{N}, \texttt{R}, |, \#, ;\}$
  - kóduje se celá přechodová funkce :
    - $\forall C \in \delta : (q, c \to q', c', Z)$
    - $(q)_B, (c)_B, (q')_B, (c')_B, Z$
    - $C_1 \# C_2 ... \# C_n$
- $\Gamma \to \{\texttt{0}, \texttt{1}\}^3$
- $\{\texttt{0}, \texttt{1}\}^* \to \text{index}(w)$
- 3-páskový TS
  1. $\langle M, x \rangle$
  2. pracovní páska M, zakódovaná do $\{\texttt{0}, \texttt{1}, |\}$
  3. $(q_i)_B$
- nerozhodnutelnost:
  - $$\boldsymbol{A}_{i,j} = \begin{cases} 1 \text{ pokud } w_j \in L(M_i) \\ 0 \text{ pokud } w_j \notin L(M_i) \end{cases}$$
  - DIAG = negace diagonály $\boldsymbol{A} \implies$ liší se od každého řádku

(B2) RAM a ekvivalence s Turingovým strojem.

- instrukce
  - set reg to constant
  - add
  - sub
  - copy to indirect
  - copy from indirectrandom access stored program
  - jump if not zero
  - read
  - write
- RASP – Random Access Stored Program (já bych to pojmenoval XRAM – eXecutable RAM ☒)
- PRAM – Parallel RAM
- TS ⟶ RAM

  - zleva omezená páska
  - triv
- RAM $\longrightarrow$ TS
  - pásky
    1. Vstup
    2. Výstup
    3. Paměť RAM
       - index | hodnota # index | hodnota ...
    4. Pomocná

(B3) Vlastnosti (turingovsky) rozhodnutelných a částečně rozhodnutelných jazyků (uzávěrové vlastnosti, Postova věta, enumerátory).

- PD i DEC jsou uzavřené na $\cap$, $\cup$, $\cdot$, $*$
- DEC jsou uzavřené na $\overline{L}$
- co-PD := $\overline{\text{PD}}$
- Postova věta: PD $\cap$ co-PD $=$ DEC (TS pro L a $\overline{L}$ paralelně rozhodnou)
- $L = \{x \in \Sigma^* \mid \exists y \in \Sigma^* \langle x, y \rangle \in B\}$
- PD $\iff$ existuje enumerátor
- DEC $\iff$ existuje enumerátor, enumeruje v shortlex

(B4) Definice základních tříd složitosti a důkaz $\text{NTIME}(f(n)) \subseteq \text{SPACE}(f(n))$.

- pro NTS v NTIME(f) lze zkonstruovat DTS, který pracuje ve SPACE(f)
  - iterujeme přes všechny posloupnosti adresy uzlů stromu výpočtu
  - posloupnost je dlouhá max f, stroj jako takový nezabere víc než f

(B5) Definice základních tříd složitosti a důkaz věty o vztahu prostoru a času $\forall L \in \text{NSPACE}(f(n)) \exists c : L \in \text{TIME}\left(2^{c \cdot f(n)}\right)$

- $2^{c \cdot f(n)}$ je počet možných konfigurací, po kterých se původní NTS nutně zacyklí

- všechny se dají prohledat v $\mathcal{O}\left(2^{c \cdot f(n)}\right)$
- velikost konfigurace nevíme dopředu, můžeme generovat inkrementálně
- c je různé pro každý stroj, primárně protože velikost abecedy a počet stavů

(B6) Dvě definice třídy NP a jejich ekvivalence.

- NTIME($p(n)$)
- polynomiální verifikátor
- $\Longrightarrow$ verifikátorem je posloupnost konfigurací
- $\Longleftarrow$ NTS zapisuje na pásku certifikát

(B7) Polynomiální převod 3-SAT na Vrcholové pokrytí.

- dvojičky pro proměnné a trojúhélníky pro klauzule

(B8) Definice třídy FPT a kernelu a jejich souvislost. Kernelizace Vrcholového pokrytí

- Problémy řešitelné v $\mathcal{O}(f(k) \cdot |I|^c)$, pro algoritmicky vyčíslitelnou f a konst. c
- Kernel = problém, jehož velikost závisí už jen na k
- kernelizace VP
  - iterujeme
    1. odstraníme izolované vrcholy
    2. „před-označíme“ všechny vrcholy s $\deg(v) > k$ (odstraníme je a snížíme k)
  - pokud už nelze, buď ($|V| \leq k^2 + k$ a $|E| \leq k^2$) nebo neexistuje vrcholové pokrytí nejvýš k

(B9) Definice třídy FPT a parametrizovaný algoritmus pro Vrcholové pokrytí založený na prohledávání s omezenou hloubkou (se složitostí menší než $\mathcal{O}^*\left(2^k\right)$).

- vybírám libovolnou hranu, označím jeden nebo druhý konec a rekurzím

(B10) Třída #P a #P-úplnost, důkaz těžkosti počítání cyklů v grafu.

- #P převádí přes konstantní orákulum polynom. počtem dotazů for some reason
- parsimonious převod („šetřivý“, „skrblík“) zachovává počet certů
- zaokrouhlení počtu vrcholů na 2^kNP ≠ co-NP
- každou hranu v G nahradíme $m = n \log n$ dlohou pomlázkou $\longrightarrow G'$
- OHK $\Longrightarrow$ počet cyklů alespoň $(2^m)^n = \left(2^{n \log n}\right)^n = n^{n^2}$
- !OHK
  - každý cyklus v G je max. $n - 1$ dlouhý
  - G obsahuje nejvýš n^{n-1} cyklů
  - počet cyklů nejvýš $(2^m)^{n-1} \cdot n^{n-1} < n^{n^2}$

(B11) Třída co-NP a co-NP-úplnost.

- $\overline{\text{SAT}}$ a TAUT
- převádí se normálně $\leq_m^p$
- P $\subset$ NP $\cap$ co-NP

(B12) Pseudonáhodné generátory, jednosměrné funkce a jejich souvislost s kryptografií (symetrické šifrování, bit-commitment).

- $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$
- $\forall n \in \mathbb{N} : \ell(n) > n$
- pro každý pravděpodobnostní poly alg $\mathcal{A}$ platí, že s delším řetězcem je rozdíl v pravděpodobnosti předpovědi dalšího bitu zanedbatelný
- závazek s 3n bity, $c = \begin{cases} G(y) \text{ pokud } b{=}1 \\ G(y) \oplus r \text{ pokud } b{=}0 \end{cases}$

(B13) Příklad zjemnělé redukce (redukce SETH na OV nebo OV na hledání regulárního výrazu v textu).

- ne ❤