Automaty a gramatiky (BI-AAG)

12. Jazyky kontextové, neomezené. Turingův stroj.

Jan Holub

Katedra teoretické informatiky

Fakulta informačních technologií

ČVUT v Praze

Turingův stroj

Definice

Deterministický Turingův stroj je sedmice $R = (Q, T, G, \delta, q_0, B, F)$, kde:

- Q je konečná množina vnitřních stavů,
- T je konečná vstupní abeceda,
- G je konečná pracovní abeceda ($T \subseteq G$),
- δ je zobrazení z $(Q \setminus F) \times G$ do $Q \times G \times \{-1, 0, 1\}$,
- $q_0 \in Q$ je počáteční stav,
- B je prázdný symbol (Blank, $B \in G \setminus T$),
- $F \subseteq Q$ je množina koncových stavů.

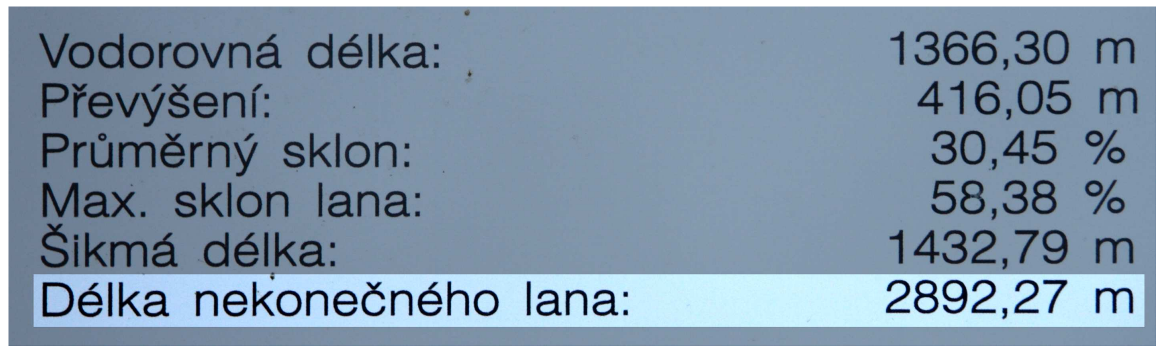
T.s. může zapisovat na vstupní pásku a libovolně se po ní pohybovat.

http://aturingmachine.com

Turingův stroj

Nekonečná páska?

Žádný problém.



Turingův stroj

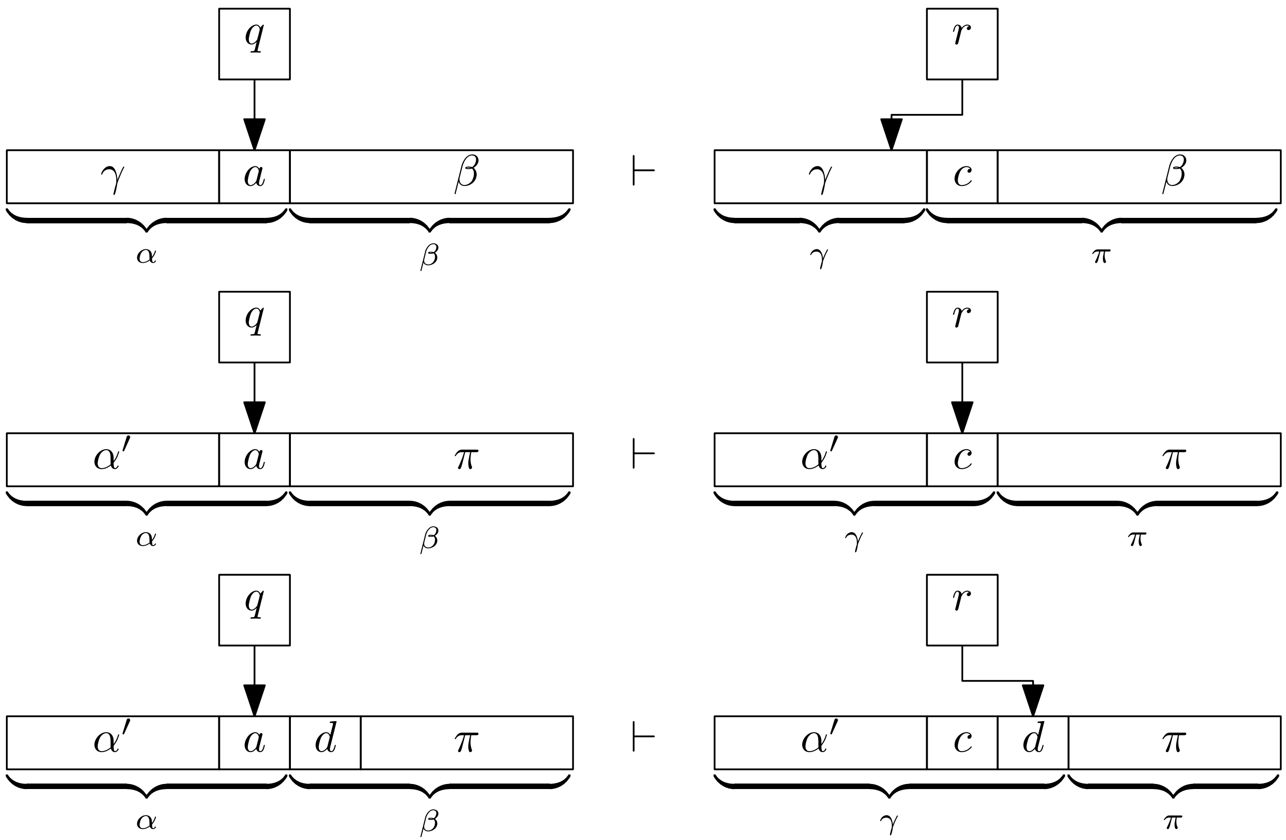
Konfigurace T. s. R: $(\alpha, q, \beta) \in G^* \times Q \times G^*$, kde

- q je okamžitý vnitřní stav,
- hlava R na vstupní pásce čte pozici $|\alpha|$,
- na i-tém políčku vstupní pásky je i-té písmeno řetězce $\alpha\beta$, když $i \leq |\alpha\beta|$, nebo B když $i > |\alpha\beta|$.

Turingův stroj

T. s. přejde $(\alpha, q, \beta) \vdash (\gamma, r, \pi)$, pokud:

- $\alpha = \gamma a, c\beta = \pi, \delta(q, a) = (r, c, -1)$, $a, c \in G$,
- $\alpha = \alpha' a, \gamma = \alpha' c, \beta = \pi, \delta(q, a) = (r, c, 0)$, $a, c \in G$,
- $\alpha = \alpha' a, \gamma = \alpha' cd, \beta = d\pi, \delta(q, a) = (r, c, 1)$, $a, c, d \in G$.



Turingův stroj

Definice

Turingův stroj $R = (Q, T, G, \delta, q_0, B, F)$ *přijímá slovo* $a\alpha \in T^+$, pokud platí $(a, q_0, \alpha) \vdash^* (B, q, \varepsilon)$, $q \in F$.
T. s. R *přijímá* ε, pokud $(B, q_0, \varepsilon) \vdash^* (B, q, \varepsilon)$, $q \in F$.

Definice

Jazyk L je *rekurzivně spočetný*, pokud je přijímán nějakým T. s. R ($L = L(R)$).

Věta

k-páskové Turingovy stroje přijímají právě rekurzivně spočetné jazyky, $k \geq 1$.

Nedeterministický Turingův stroj

Definice

Nedeterministický T. s. je sedmice $R = (Q, T, G, \delta, q_0, B, F)$, kde:

- Q je konečná množina vnitřních stavů,
- T je konečná vstupní abeceda,
- G je konečná pracovní abeceda ($T \subseteq G$),
- δ je zobrazení z $(Q \setminus F) \times G$ do $\mathcal{P}(Q \times G \times \{-1, 0, 1\})$,
- $q_0 \in Q$ je počáteční stav,
- B je prázdný symbol (Blank, $B \in G \setminus T$),
- $F \subseteq Q$ je množina koncových stavů.

Nedeterministický T. s. přijímá slovo $a\alpha$, jestliže existuje $(a, q_0, \alpha) \vdash^* (B, q, \varepsilon)$, $q \in F$.
N. T. s. přijímá slovo ε jestliže existuje $(B, q_0, \varepsilon) \vdash^* (B, q, \varepsilon)$, $q \in F$.

Nedeterministický Turingův stroj

Věta

Nedeterministické Turingovy stroje přijímají právě rekurzivně spočetné jazyky.

Lineárně omezený Turingův stroj

Lineárně omezený Turingův stroj $=$ Lineárně omezený automat

Definice
T. s. je *lineárně omezený*, pokud nemůže překročit délku k-násobku vstupního slova pro nějaké $k \geq 1$.

$\delta(q, B) = \emptyset$ nebo $\delta(q, B) = (q', B, -1)$

Věta
Pro každou nezkracující gramatiku G existuje ekvivalentní kontextová gramatika.

Věta
Pro každou gramatiku G existuje T. s. R takový, že $L(G) = L(R)$. Pro každou nezkracující gramatiku G existuje lineárně omezený T. s. R takový, že $L(G) = L(R)$

Lineárně omezený Turingův stroj

Důsledek

Gramatiky generují právě rekurzivně spočetné jazyky. Kontextové jazyky jsou přijímány právě lineárně omezenými T. s.

Věta

Rekurzivně spočetné jazyky jsou uzavřené na operacích sjednocení, součin a iterace.

Věta

Kontextové jazyky jsou uzavřené na operacích sjednocení, součin, iterace a doplněk.

Algoritmus

Definice

Turingův stroj R *rozhoduje* jazyk L nad abecedou T, když se výpočet pro každé slovo zastaví a $L(R) = L$.

Jazyk L je rekurzivní, pokud existuje T. s., který ho rozhoduje.

Věta

L je rekurzivní, právě když L a $\overline{L}$ jsou rekurzivně spočetné.

Věta

Každý kontextový jazyk je rekurzivní.

Church-Turingova teze

Každý jazyk, který lze nějakým způsobem popsat konečným výrazem, je rekurzivně spočetný.

Ke každému algoritmu existuje ekvivalentní Turingův stroj.

Univerzální Turingův stroj

Definice

Turingův stroj je *univerzální*, právě když přijímá všechny dvojice $(\textit{kód}(R); \alpha)$ takové, že T. s. R přijímá slovo α.

Nerozhodnutelné problémy:

- Problém zastavení T. s. (Halting Problem):

  $$\textit{Zastaví}(P; \alpha) = \begin{cases} \textit{ano}, & \text{když se } P \text{ pro vstup } \alpha \text{ zastaví,} \\ \textit{ne}, & \text{když se } P \text{ pro vstup } \alpha \text{ nezastaví.} \end{cases}$$

  P: L: if $\textit{Zastaví}(P; P)$ then goto L else halt

- Postův korespondenční problém
- ...

Třídy P a NP

Problémy

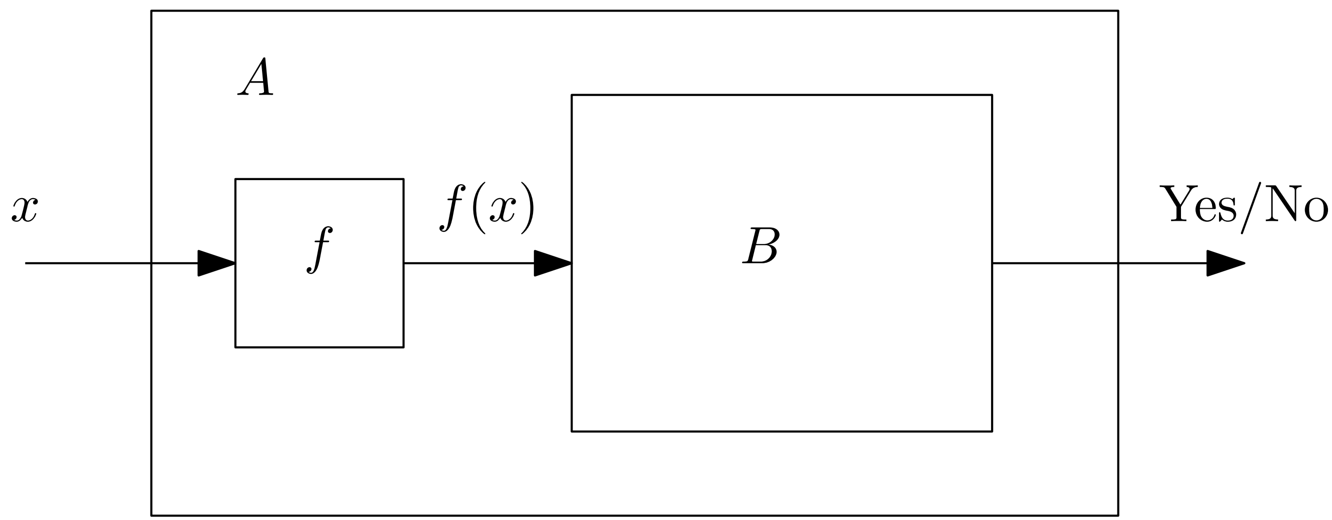
- rozhodovací (ano/ne)
- optimalizační (nejlepší řešení)

Definice
Třída NP (non-deterministic polynomial-time) je množina problémů, které lze řešit v polynomiálně omezeném čase na nedeterministickém Turingově stroji.

Definice
Třída P (polynomial-time) je množina problémů, které lze řešit v polynomiálně omezeném čase na deterministickém Turingově stroji.

Polynomiální redukce

Definice (Polynomiální redukce)
Říkáme, že jazyk $A \subseteq \{0,1\}^*$ je redukovatelný v polynomiálním čase (podle Karpa) na jazyk $B \subseteq \{0,1\}^*$ (značíme $A \leq_p B$) jestliže existuje funkce $f : \{0,1\}^* \rightarrow \{0,1\}^*$ spočetná v polynomiálním čase taková, že pro každé $x \in \{0,1\}^*$, $x \in A$ právě tehdy když $f(x) \in B$.



Polynomiální redukce

Příklad (CNF-SAT $\leq_p$ Hledání kliky)
Problém CNF-SAT:
Je daný výraz v konjunktivní normální formě (CNF). Existuje přiřazení, při kterém bude výraz ohodnocen jako pravdivý?

$$(x \vee y \vee z) \wedge (\neg x \vee z \vee w) \wedge (\neg x \vee \neg w) \wedge (\neg w \vee x)$$

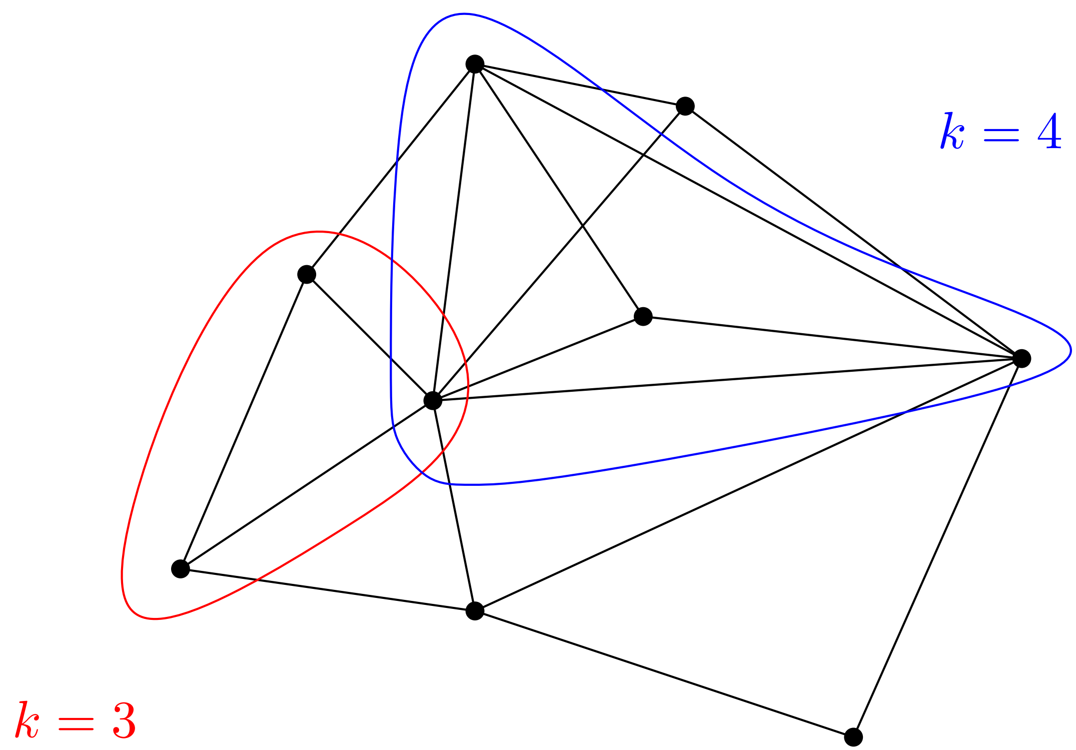
přiřazení: x, y libovolné hodnoty, $z =$ true, $w =$ false

Polynomiální redukce

Příklad (CNF-SAT $\leq_p$ Hledání kliky (pokračování))

Problém Hledání kliky:

Mějme graf $G = (V, E)$ a číslo k. Existuje v grafu klika o velikosti k, t.j. podmnožina vrcholů S o velikosti k taková, že pro každou dvojici $u, v \in S$, platí $(u, v) \in E$?



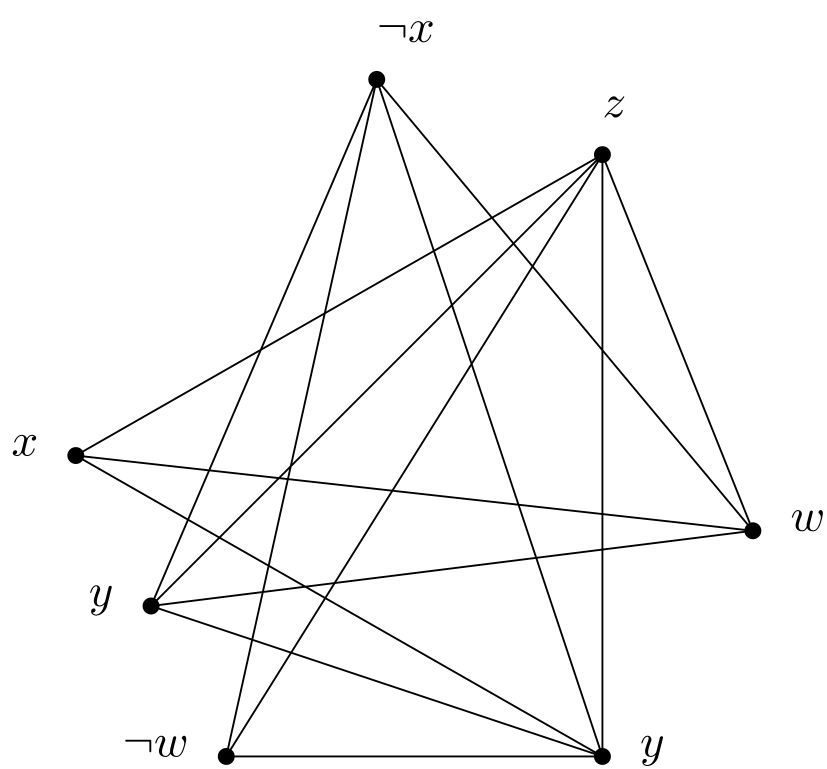
Polynomiální redukce

Příklad (CNF-SAT $\leq_p$ Hledání kliky (pokračování))

Redukce:

Spojíme hranami uzly různých klauzulí, které mohou současně nabýt hodnoty true.

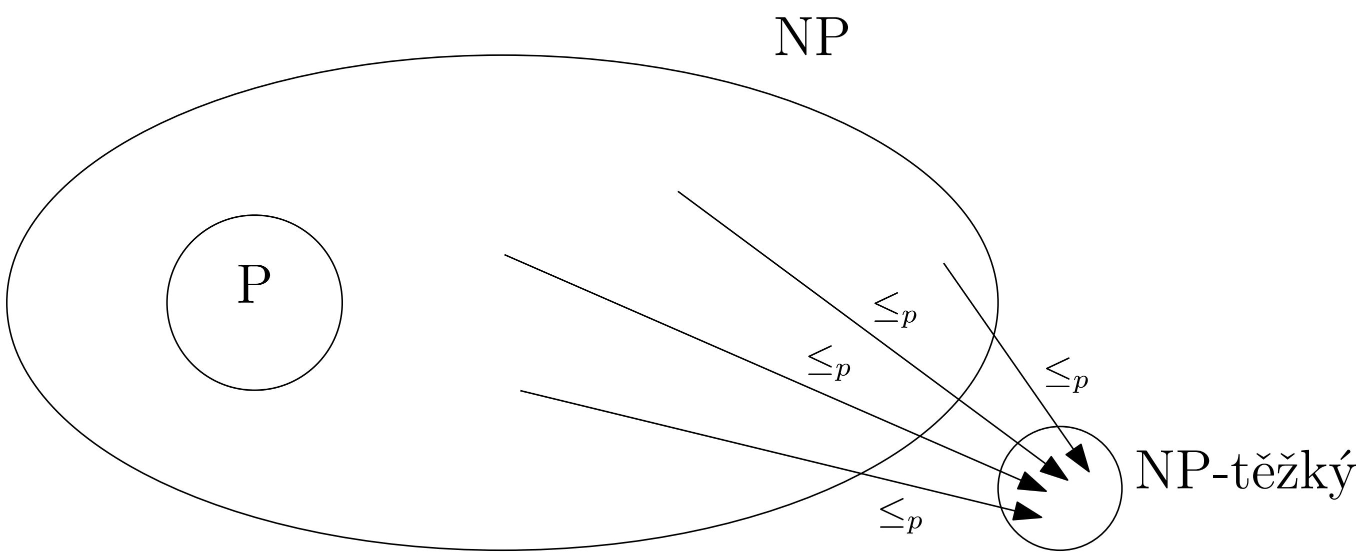
$(x \vee y \vee \neg w) \wedge (\neg x \vee z) \wedge (y \vee w)$



NP-těžký problém

Definice (NP-těžký)
Říkáme, že B je NP-těžký, jestliže $A \leq_p B$ pro každé $A \in$ NP.

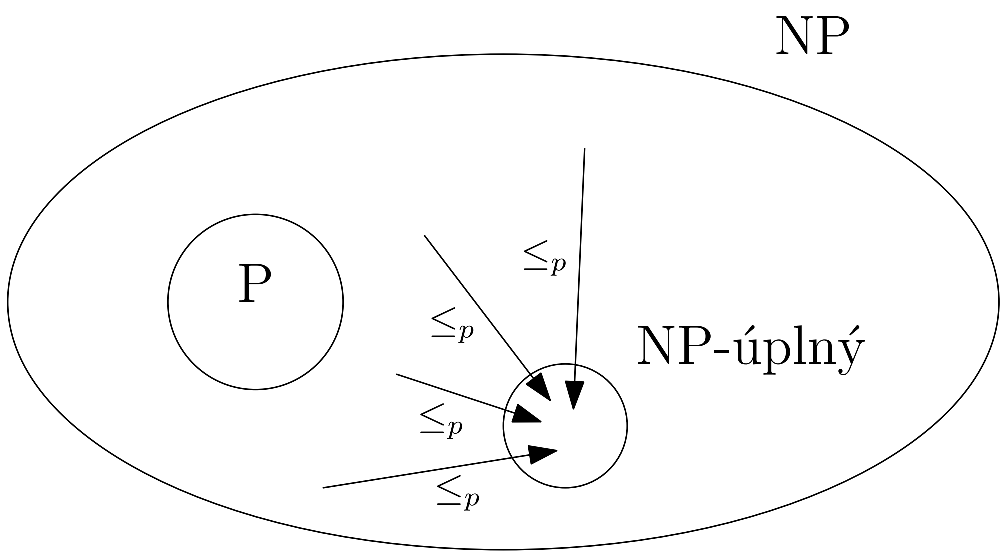
NP-těžké (NP-hard) problémy jsou takové problémy, na které jsou polynomiálně redukovatelné všechny ostatní problémy z NP. (Alespoň tak těžké, jako nejtěžší problémy v NP.)



NP-úplný problém

Definice (NP-úplný)
Říkáme, že B je NP-úplný, jestliže B je NP-těžký a $B \in$ NP.

NP-úplné (NP-complete, NPC) problémy jsou takové nedeterministicky polynomiální problémy, na které jsou polynomiálně redukovatelné všechny ostatní problémy z NP.



NP-úplné problémy

- **Cook' Theorem**
  Jazyk SAT (splnitelnost boolských formulí) je NP-úplný.
- Barvení grafu, hledání kliky, kachličkování, podmnožinový součet, problém batohu
- ...