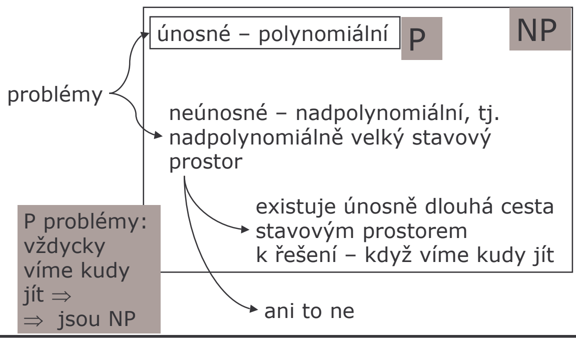
Třídy P a NP

- Model výpočtu: Turingův stroj
- Rozhodovací problémy: třídy P a NP
- Optimalizační problémy: třídy PO a NPO

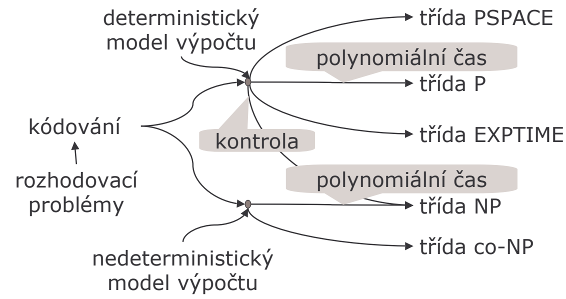
Třídy P a NP

- Model výpočtu: Turingův stroj
- Rozhodovací problémy: třídy P a NP
- Optimalizační problémy: třídy PO a NPO

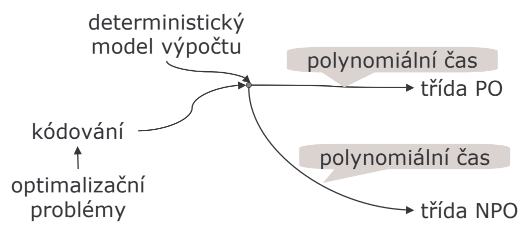
Složitost problémů



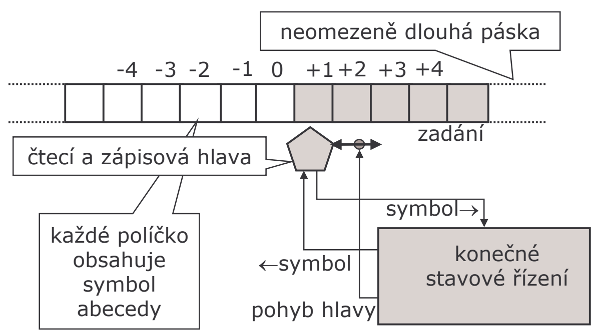
Třídy rozhodovacích problémů



Třídy optimalizačních problémů



Turingův stroj



Emulace RAM stroje

- výpočet instance velikosti n na stroji s přímým adresováním v čase $T(n)$
- na Turingově stroji?
- v $T(n)$ krocích lze popsat nejvýše $T(n)$ buněk paměti a tedy $T(n)$ políček pásky
- políčka lze uspořádat souvisle
- v nejhorším případě je nutno v každém kroku přestavit pásku o $T(n)$ políček
- výpočet na Turingově stroji v čase $T^2(n)$
- polynomiální poměr

Turingův stroj

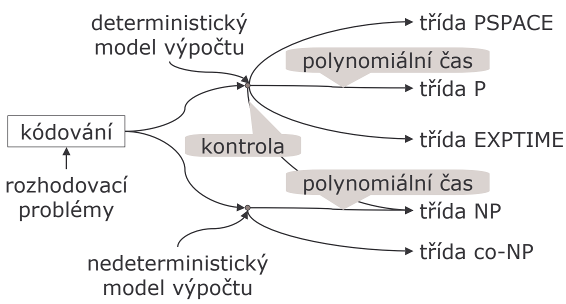
- Program:
  - množina Γ symbolů pásky, symbol $b \in \Gamma$
  - $\Sigma \subset \Gamma$ množina vstupních symbolů, $b \notin \Sigma$
  - množina stavů Q, počáteční stav $q_0 \in Q$, koncové stavy $q_{ANO}, q_{NE} \in Q$
  - přechodová funkce δ: $(Q - \{q_{ANO}, q_{NE}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{-1, +1\}$
- Inicializace: stav q_0, políčko 1 pásky
- Konec: q_{ANO}, q_{NE}
- Výpočet: $(q \in Q, s \in \Gamma) \rightarrow (q', s', \Delta)$

stav
čtený symbol
nový stav
zapsaný symbol
pohyb pásky

Řešení problému deterministickým Turingovým strojem

- Definice: řešení problému Turingovým strojem
- Program *M* pro deterministický Turingův stroj <u>řeší rozhodovací problém Π</u>**,** jestliže se výpočet zastaví po konečném počtu kroků pro každou instanci problému Π.
- Program *M* pro deterministický Turingův stroj <u>řeší rozhodovací problém Π v čase t</u>**,** jestliže se výpočet zastaví po t krocích pro každou instanci problému Π.

Třídy rozhodovacích problémů



... měření složitosti

- Jak měřit velikost instance?
  - hrubá míra: počet prvků instance (uzlů, čísel, prvků množiny)
  - jemná míra: počet bitů, nutných k zakódování instance
- Jak měřit čas výpočtu?
  - počet „typických operací“
  - počet kroků jednotného výpočetního modelu

Kódování instance

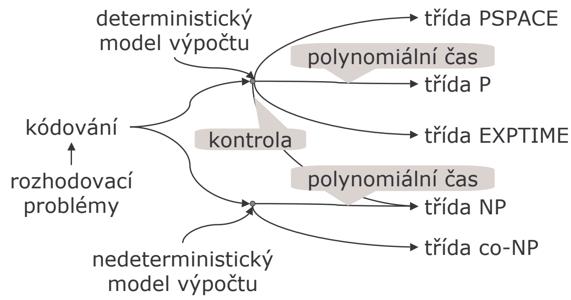
- Vstup Turingova stroje: řetěz q^* symbolů $q \in \Gamma$
- Výstup: „ano“, jestliže q^* kóduje instanci problému, která má řešení
- … lze mluvit o P-jazycích, NP-jazycích …
- abeceda Γ nezávisí na instanci
- způsob kódování instance neovlivní čas výpočtu více než polynomiálně

Kódování grafu *G*=(*V*, *E*)

|*X*| … počet prvků množiny *X*

- matice sousednosti:
  $|V|^2$ bitů
- incidenční matice:
  $|V||E| = O(|V|^3)$ bitů
- seznam hran jako dvojic indexů uzlů:
  $O(|E|\log|V|) = O(|V|^2\log|V|)$ bitů
- schválnost – patologický případ
  matice sousednosti a za ní 2^n nul

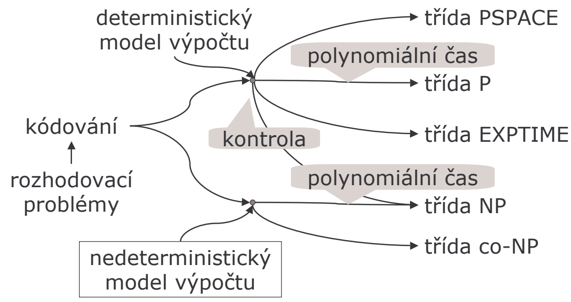
Třídy rozhodovacích problémů



Třída P

- Definice: třída P
  Rozhodovací problém patří do třídy P, jestliže pro něj existuje program pro deterministický Turingův stroj, který jej řeší v čase $O(n^k)$, kde *n* je velikost instance a *k* konečné číslo.
- PSPACE:
  polynomiální množství paměti (pásky)
- EXPTIME:
  v čase $O(2^{P(n)})$, kde $P(n)$ je polynom ve velikosti instance *n*.

Třídy rozhodovacích problémů



NP

Non-deterministically Polynomial

Nedeterministicky Polynomiální

Nedeterministický Turingův stroj

- Program: ...přechodová relace
  $\delta \subset (Q-\{q_{ANO}, q_{NE}\}) \times \Gamma \times Q \times \Gamma \times \{-1, +1\}$
- Výpočet: $(q \in Q, s \in \Gamma) \rightarrow \{(q', s', \Delta)\}$
- Představa: v každém kroku se stroj naklonuje a každá kopie vykoná jeden konkrétní možný krok
- Představa: jedna z kopií vykoná onu únosně dlouhou cestu stavovým prostorem, pokud tato cesta existuje, tj. pokud instance má řešení

Řešení problému **ne**deterministickým Turingovým strojem

- Definice: řešení problému nedeterministickým Turingovým strojem

  Nechť Π_{ANO} je množina instancí problému Π, které mají výstup ANO. Program M pro nedeterministický Turingův stroj <u>řeší rozhodovací problém Π v čase t,</u> jestliže se výpočet zastaví po t krocích pro každou instanci $I \in \Pi_{\text{ANO}}$ problému Π.

Vlastnosti

- Nic se neříká o instancích Π_{NE}. Pokud hledáme únosně dlouhou cestu k řešení, na Π_{NE} nemá význam.
- Věta (výpočetní mohutnost nedeterminismu): Jestliže nedeterministický Turingův stroj řeší problém Π v čase $T(n)$, pak deterministický Turingův stroj řeší Π v čase $2^{O(T(n))}$.

Třída NP

- Definice: třída NP
  Rozhodovací problém Π <u>patří do třídy NP</u>, jestliže pro něj existuje program pro nedeterministický Turingův stroj, který každou instanci $I \in \Pi_{\text{ANO}}$ problému Π řeší v čase $O(n^k)$, kde n je délka vstupních dat a k konečné číslo.
- Definice: třída NP
  Rozhodovací problém Π <u>patří do třídy NP</u>, jestliže pro každou instanci $I \in \Pi_{\text{ANO}}$ problému existuje konfigurace Y taková, že kontrola, zda Y je řešením, patří do P. V této souvislosti nazýváme Y <u>certifikátem</u>.

omezující podmínky lze vyhodnotit v polynomiálním čase

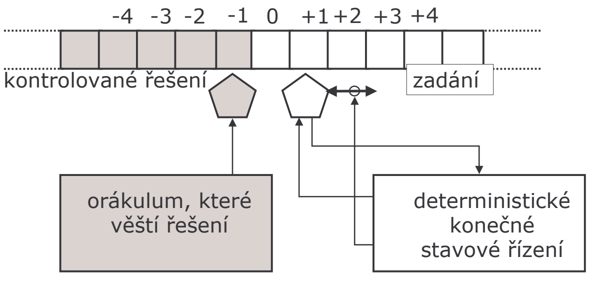
Příklad: Hamiltonova kružnice v grafu (HC), nedeterministický algoritmus

1. Nechť podgraf $G' = (V', E')$ je tvořen libovolným uzlem v původního grafu $G=(V,E)$.
2. V každém kroku, nechť každá „kopie“ algoritmu přidá jednu (různou) hranu $e \in E\text{-}E'$, $e = (u,v)$ takovou, že $u \in V'$, $v \notin V'$, stupeň u=1 a dále uzel v.
3. Není-li to možné, příslušná „kopie“ končí.
4. Jestliže přidaná hrana utvoří z G'
   1. kružnici kratší než $|V|$, příslušná „kopie“ končí.
   2. kružnici délky $|V|$, celý algoritmus vydá výstup „ano“.

Nedeterministický algoritmus, poznámky

- Jestliže graf obsahuje Hamiltonovu kružnici, po $|V|$ krocích je nalezena.
- Existuje nedeterministický algoritmus řešící problém HC $\Rightarrow$ HC patří do NP
- V tomto případě máme štěstí – pokud Hamiltonovu kružnici nenajdeme po $|V|$ krocích, pak neexistuje. Obecně tomu tak není

Nedeterministický Turingův stroj – představa, která vychází z kontroly řešení



Příklad: Hamiltonova kružnice v grafu – polynomiální kontrola

- Konfigurace: podgraf $G' = (V', E')$ původního grafu $G=(V,E)$.
- Certifikát: podgraf $G' = (V', E')$, o kterém se tvrdí, že je to Hamiltonova kružnice
- Kontrola:
  - $|V|$ uzlů … $O(|V|)$
  - $E' \subset E$ … $O(|V|)$
  - žádný uzel dvakrát (řazení) … $O(|V| \log |V|)$
- Existuje deterministický algoritmus kontrolující certifikát HC $\Rightarrow$ HC patří do NP

Vztah tříd P a NP

funkce je zvláštním případem relace

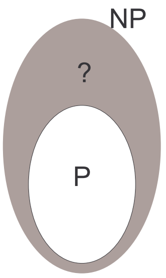
deterministický automat je zvláštním případem nedeterministického automatu

deterministický Turingův stroj je zvláštním případem nedeterministického Turingova stroje

$P \subseteq NP$

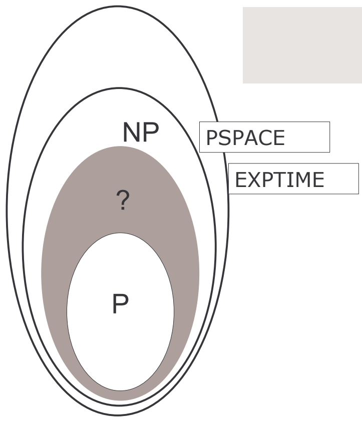
instance $I \in \Pi_{ANO}$ jsou podmnožinou všech instancí

Vztah tříd P a NP



- možná, že P = NP: na každý NP-problém existuje polynomiální algoritmus, ale my o něm nevíme
- ale jsou příznaky, že $P \subset NP$
- jeden z hlavních příznaků: viz příště

… a dál



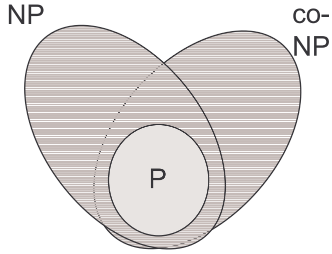
Problémy mimo NP

- Je dán graf $G=(V,E)$. Je tento graf <u>prost</u> Hamitonových kružnic?
- Je dána Booleovská formule $F(X)$ n proměnných $X = (x_1, x_2, \dots x_n)$. Je tato formule <u>nesplnitelná</u>?

Potíž:
- při odpovědi „ano" nemáme ceritifikát
- nemůžeme očekávat „únosně dlouhou cestu stavovým prostorem"

„protějšek" třídy NP:
třída co-NP

Třída P, NP, co-NP



Problémy doopravdy mimo NP

- SAT: Booleovská formule $F(X)$ n proměnných $X = (x_1, x_2, \dots x_n)$.
- $\exists\, Y, F(Y) = 1$?

- Booleovská formule $F(X_1,X_2)$ $2n$ proměnných $X_1 = (x_1, x_2, \dots x_n)$, $X_2 = (x_{n+1}, x_{n+2}, \dots x_{2n})$
- $\exists Y_1, \forall Y_2, F(Y_1, Y_2) = 1$?

algoritmus pro SAT musíme použít 2^n krát

Kontrola řešení

- Booleovská formule $F(X_1,X_2)$ $2n$ proměnných $X_1 = (x_1, x_2, \dots x_n)$, $X_2 = (x_{n+1}, x_{n+2}, \dots x_{2n})$
- $\exists\, Y_1, \forall\, Y_2, F(Y_1, Y_2) = 1$?

Dáno řešení Y_1, kontrola:

- platí, že $\forall Y_2, F(Y_1, Y_2) = 1$
- neplatí, že $\exists Y_2, F(Y_1, Y_2) = 0$
- neplatí, že $\exists Y_2, \neg F(Y_1, Y_2) = 1$
- $\neg F(Y_1, X_2)$ je nesplnitelná

problém kontroly je v **co-NP**

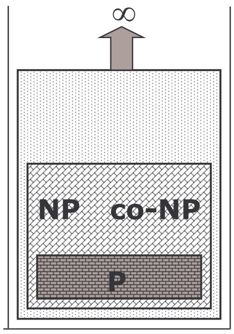
Polynomiální hierarchie

takto to vypadá právě tehdy, pokud P ≠ NP

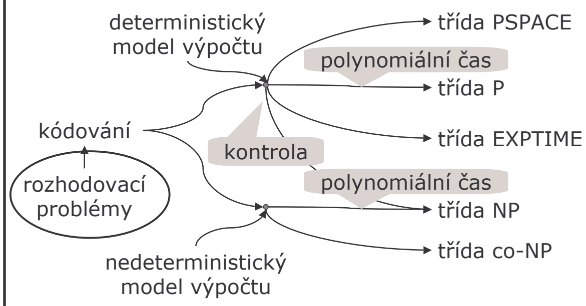
třída problémů, jejichž kontrola leží v předchozí třídě a jejich co- protějšků

třída problémů, jejichž kontrola leží v P a jejich co- protějšků

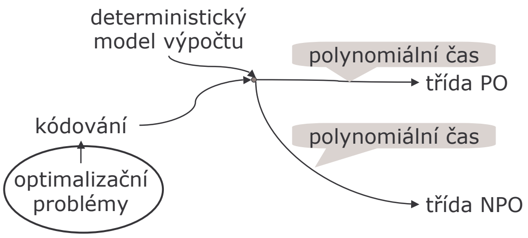
třída P



Třídy rozhodovacích problémů



Třídy optimalizačních problémů



Řešení optimalizačního problému Turingovým strojem

- Definice: řešení optimalizačního problému v čase t
- Program M pro deterministický Turingův stroj <u>řeší optimalizační problém Π v čase t,</u> jestliže se výpočet zastaví po t krocích pro každou instanci problému Π, která má řešení, a na pásce je zapsán výstup instance.
- Program M pro deterministický Turingův stroj <u>počítá optimalizační kritérium problému Π v čase t,</u> jestliže pro každé řešení každé instance problému Π, zapsané na pásce jako vstupní data, se výpočet zastaví po t krocích a na pásce je zapsána hodnota optimalizačního kritéria.

Třída NPO

- Definice: optimalizační problém Π patří do třídy NPO, jestliže splňuje následující podmínky:
  - velikost výstupu instance je omezena polynomem ve velikosti instance

    výstup lze zapsat v polynomiálním čase
  - problém, zda daná konfigurace je řešením, patří do P

    omezující podmínky lze vyhodnotit v polynomiálním čase
  - existuje program pro Turingův stroj, který vypočítá hodnotu optimalizačního kritéria pro všechna řešení všech instancí v polynomiálním čase

    optimalizační kritérium lze vyhodnotit v polynomiálním čase

Třída PO

- Definice (třída PO): optimalizační problém Π patří do třídy PO, jestliže splňuje následující podmínky:
  - patří do NPO
  - existuje program pro Turingův stroj, který každou instanci vyřeší v polynomiálním čase.
- Příklad: problém nejkratší cesty v grafu $G=(V,E)$ patří do PO
  - velikost výstupu O(|E|)
  - ověření omezení O(|E| log |E|)
  - optimalizační kritérium O(|E|)
  - existuje polynomiální algoritmus (Dijkstra)

Příklad NPO problému: optimalizační TSP

Dána množina n měst $C=\{c_1,c_2,\dots,c_n\}$. Pro každá dvě města c_i,c_j je dána vzdálenost $d(c_i,c_j)$. Nalezněte uzavřenou túru, která prochází každým městem právě jednou a má nejmenší délku.

- Velikost výstupu instance: $O(|C|)$
- Kontrola túry: $O(|C| \log |C|)$ - viz Hamiltonova kružnice
- Výpočet optimalizačního kritéria: $O(|C|)$