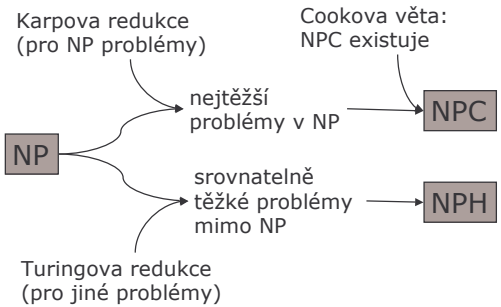
NP-úplné (NPC) a NP-těžké (NPH) problémy

- Karpova redukce
- NP-úplné problémy (NPC)
- Cookova věta
- Turingova redukce
- NP-těžké problémy (NPH)
- problémy mezi P a NPC

Pojmy X-těžký a X-úplný
(X-complete a X-hard)

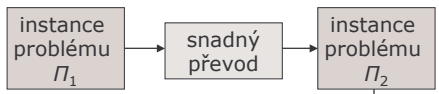
- Problém Π je X-těžký, jestliže se efektivní řešení všech problémů z třídy X dá zredukovat na efektivní řešení problému Π.
- Problém Π je X-úplný, jestliže je X-těžký a sám patří do třídy X.
- efektivní řešení: v polynomiálním čase (jindy např. s omezenou chybou)
- zredukovat: vyřešit pomocí
- za X dosadit: NP, NPO, APX…

NPH a NPC

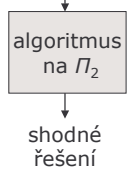


Co jsou nejtěžší problémy v NP?

Co je „lehčí" a „těžší" problém?

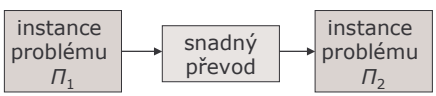


- získali jsme algoritmus na Π_1
- který není horší než algoritmus na Π_2



- Π_1 je nejvýše tak těžký jako Π_2
- Π_2 je nejméně tak těžký jako Π_1

Který problém je nejtěžší?



- Π_1 je nejvýše tak těžký jako Π_2
- Π_2 je nejméně tak těžký jako Π_1

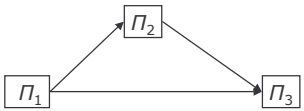
Ten, na který jdou převést všechny ostatní.

Karpova redukce
(polynomiální transformace)

- Definice Karpovy redukce
  Rozhodovací problém Π_1 je Karp-redukovatelný na Π_2 ($\Pi_1 \propto \Pi_2$), jestliže existuje polynomiální program pro (deterministický) Turingův stroj, který převede každou instanci I_1 problému Π_1 na instanci I_2 problému Π_2 tak, že výstup obou instancí je shodný.
- Jiné značení: $\lhd$

Vlastnosti

- Tranzitivita
  $\Pi_1 \propto \Pi_2 \wedge \Pi_2 \propto \Pi_3 \Rightarrow \Pi_1 \propto \Pi_3$



- Třídy polynomiální ekvivalence
  $\Pi_1 \propto \Pi_2 \wedge \Pi_2 \propto \Pi_1 \Rightarrow \Pi_1$ a Π_2 jsou polynomiálně ekvivalentní.

Příklad: HC $\propto$ TSP

Dán graf G=(V,E). Obsahuje tento graf Hamiltonovu kružnici?

převést

Dána množina n měst $C=\{c_1,c_2,\ldots,c_n\}$. Pro každá dvě města c_i,c_j je dána vzdálenost $d(c_i,c_j)>0$. Existuje uzavřená túra, která prochází každým městem právě jednou a má délku nejvýše B?

Karpova redukce HC $\propto$ TSP

$V, E \rightarrow C, d(c_i, c_j), B$

- Nechť každému uzlu v_i odpovídá jiné město c_i.
- Je-li $(v_i, v_j) \in E$, nechť $d(c_i, c_j)=1$ jinak $d(c_i, c_j)=2$
- Nechť $B=|V|$.

Důkaz:

1. HC $\propto$ TSP má polynomiální složitost
2. výstup je stejný
   - 2.1 $\exists$ kružnice v $G \Rightarrow \exists$ túra v C
   - 2.2 $\exists$ túra v $C \Rightarrow \exists$ kružnice v G

Důkaz HC $\propto$ TSP

1. HC $\propto$ TSP má polynomiální složitost ($n=|V|$)
   - Konstrukce měst: $O(n)$; vzdáleností: $O(n^2)$; B: $O(1)$
   - $\Rightarrow$ složitost $O(n^2)$;

2.1 $\exists$ kružnice v $G \Rightarrow \exists$ túra v C

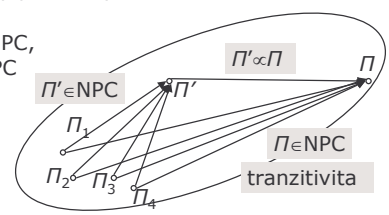
- $(v_1,v_2, \ldots v_n,v_1)$ Hamiltonova kružnice v G.
- $\Rightarrow (c_1,c_2, \ldots c_n,c_1)$ je túra v C o délce n.1 (každý úsek túry odpovídá hraně)
- $n \le B$.

2.2 $\exists$ túra v $C \Rightarrow \exists$ kružnice v G

- $(c_1,c_2, \ldots c_n,c_1)$ je túra délky nejvýše B.
- n úseků, délka $B=n$ každý úsek túry má délku 1
- $\Rightarrow$ každý úsek odpovídá hraně
- $(v_1,v_2, \ldots v_n,v_1)$ Hamiltonova kružnice v G. *Q.e.d.*

Třída NP-úplný
(NP-Complete, NPC)

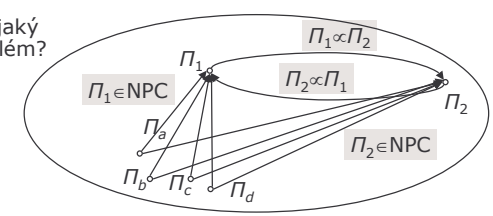
- Definice (třída NP-úplný):
- Problém Π je NP-úplný, jestliže
  - $\Pi \in$ NP
  - pro všechny problémy $\Pi' \in$ NP, $\Pi' \propto \Pi$
- Důsledek:
  $\Pi \in$NP, $\exists \Pi' \in$ NPC,
  $\Pi' \propto \Pi \Rightarrow \Pi \in$NPC



NP-úplný jako třída ekvivalence

- Všechny NPC problémy tvoří třídu ekvivalence
- $\Pi_1, \Pi_2 \in$NPC ($\Rightarrow \Pi_1 \in$NP, $\Pi_2 \in$NP)
- $\Pi_1 \propto \Pi_2$ (protože $\Pi_1 \in$NP, $\Pi_2 \in$NPC)
- $\Pi_2 \propto \Pi_1$ (protože $\Pi_2 \in$NP, $\Pi_1 \in$NPC) *Q.e.d.*

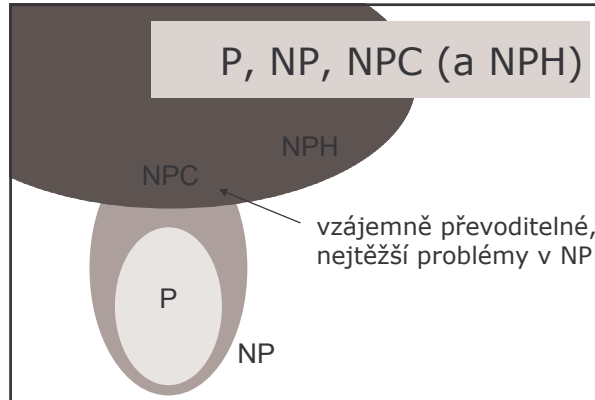
existuje vůbec nějaký NPC problém?



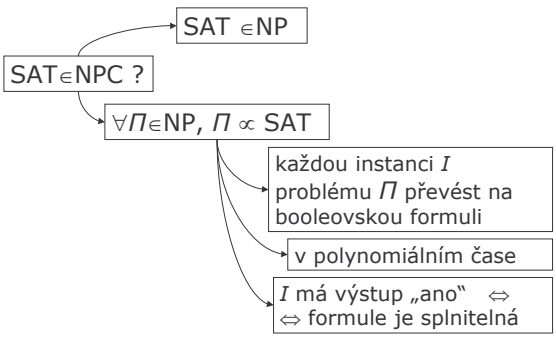
Cookova věta a důsledky

- SAT je NP-úplný
- říká, že NPC není prázdná
- otevírá cestu k důkazům NP-úplnosti převodem
- jsou známy tisíce NPC problémů
- které tvoří třídu ekvivalence
- polynomiální program na jeden ⇒ ⇒ polynomiální program na všechny
- nevypadá to, že by P=NP…

P, NP, NPC (a NPH)



Osnova důkazu



Důsledky $\Pi \in NP$, $I \in \Pi_{ANO}$

- Existuje program M pro Turingův stroj, který kontroluje certifikát Y instance I v čase $p(n)$, kde p je polynom a n velikost instance I a skončí ve stavu q_{ANO}.
- Velikost certifikátu je nejvýše $p(n)$.
- Rozsah políček pásky je $-p(n)\ldots p(n)+1$.

Konstruovaná formule

- Jestliže $I \in \Pi_{ANO}$, má ohodnocení proměnných, při němž vyjadřuje výrok proběhl výpočet stroje M, který se zastavil ve stavu q_{ANO}.
- „Náhrada naprogramovaného počítače kombinačním obvodem"
- Musí obsahovat
  - vlastnosti Turingova stroje
  - program M
  - výsledek „ano"

Celkový stav Turingova stroje

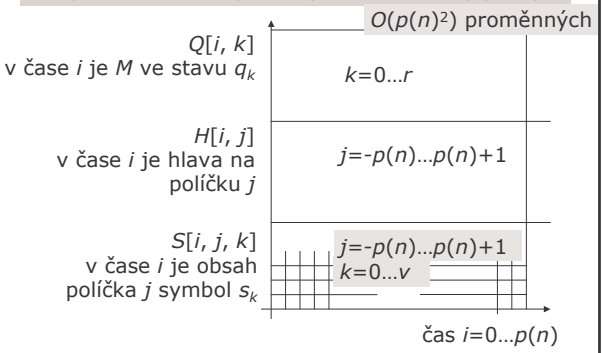
- Stav řídícího automatu
- Obsah všech políček pásky
- Pozice hlavy na pásce

Výpočet Turingova stroje

- Posloupnost celkových stavů v čase $0\ldots t$, kde t je celkový čas výpočtu
- → proměnné formule

Proměnné formule

r…počet stavů; *v*…počet symbolů abecedy pásky



Klauzule formule

musí být splněny současně (součin)

počítá to jako Turingův stroj

- v každém čase *i*, řízení je v právě jednom stavu
- v každém čase *i*, hlava je na právě jednom políčku
- v každém čase *i*, každé políčko obsahuje právě jeden symbol
- v čase 0, celkový stav je inicializován

výstup je „ano"

- v čase $p(n)$, řízení je ve stavu q_{ANO}

program

- v každém čase *i*, celkový stav je výsledkem aplikace přechodové funkce δ na předchozí celkový stav

Ukázky konstrukce některých skupin

v každém čase *i*, řízení je v právě jednom stavu

$\neg(\ Q[i,0]\ .Q[i,1]\ ) = (\neg Q[i,0] + \neg Q[i,1]\ )$

… v nejvýše jednom stavu

$(\neg Q[i,j] + \neg Q[i,j']\ )$

$i=0\ldots p(n) \quad j=-p(n)\ldots p(n)+1 \quad j'=j+1\ldots p(n)+1$

… v alespoň jednom stavu

$(\ Q[i,0] + Q[i,1] + \ldots + Q[i,r]\ )$

$i=0\ldots p(n) \quad k=0\ldots r$

Ukázky konstrukce některých skupin

v každém čase *i*, celkový stav je výsledkem aplikace přechodové funkce δ na předchozí celkový stav

když hlava není na políčku *j*, obsah se nezmění

$(\ \neg S[i,j,l] + H[i,j] + S[i+1,j,l]\ )$

$i=0\ldots p(n) \quad j=-p(n)\ldots p(n)+1 \quad l=0\ldots v$

$a \Rightarrow b = \neg a + b$

$(\ \neg H[i,j] + \neg Q[i,k] + \neg S[i,j,l] + H[i+1,j+\Delta]\ )$

$(\ \neg H[i,j] + \neg Q[i,k] + \neg S[i,j,l] + Q[i+1,k']\ )$

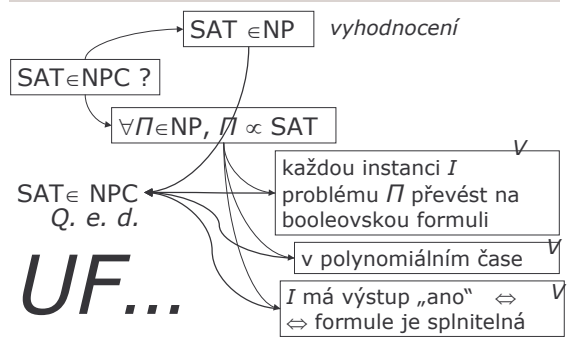
$(\ \neg H[i,j] + \neg Q[i,k] + \neg S[i,j,l] + S[i+1,j,l']\ )$

$i=0\ldots p(n) \quad j=-p(n)\ldots p(n)+1 \quad l=0\ldots v \quad k=0\ldots r$

Polynomiální složitost

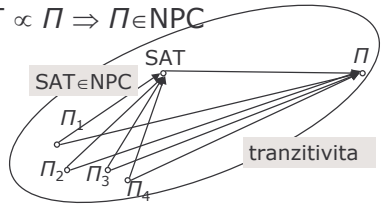
- Ukázat, že velikost výsledné formule *F* je polynomiální s *n* – velikostí původní instance
- velikost formule s množinou *C* klauzulí nad množinou *X* proměnných: $|X|.|C|$
- *r*, *v* … konstantní pro daný problém *Π*
- $|X| = O(p(n)^2) \quad |C| = O(p(n)^2)$
- $|F| = O(p(n)^4)$

Osnova důkazu

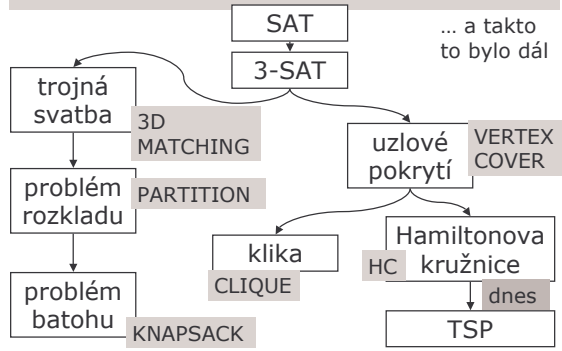


Dokazování NP-úplnosti *Π*

- Z definice (lehce nepraktické, že)
- $\Pi' \in$ NP je speciálním případem Π
- $\Pi \in$NP, $\exists \Pi' \in$ NP, $\Pi' \propto \Pi \Rightarrow \Pi \in$NP
  $\Pi \in$NP, SAT $\propto \Pi \Rightarrow \Pi \in$NP



Na počátku je SAT…



Bestiarium

- 3-SAT: každá klauzule má právě 3 literály
- Trojná svatba:
  - dány disjunktní množiny W, X, Y, $|W|=|X|=|Y|=q$, množina $M \subseteq W \times X \times Y$;
  - existuje $M' \subseteq M$ taková, že $|M'|=q$ a žádné dva prvky M' se neshodují ani v jedné souřadnici?
- Uzlové pokrytí:
  - dán graf $G=(V,E)$, celé číslo $K \leq |V|$;
  - existuje $V' \subseteq V$ taková, že $|V'| \leq K$ a $\forall (u,v) \in E$, $u \in V'$ nebo $v \in V'$?

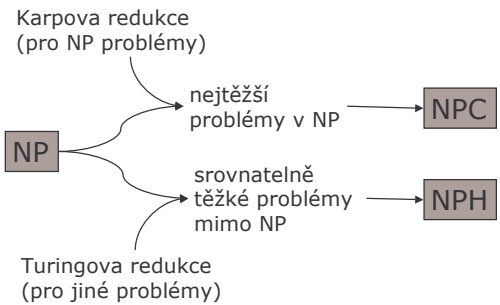
Bestiarium

- Klika: („politická klika“)
  - dán graf $G=(V,E)$, celé číslo $K \leq |V|$;
  - existuje úplný podgraf $G'=(V',E')$ grafu G takový, že $|V'| \geq K$?
- Problém rozkladu:
  - dána množina $A=\{a_1, \ldots, a_n\}$ a funkce $s: A \rightarrow \mathbf{Z}^+$;
  - existuje podmnožina $A' \subseteq A$ taková, že

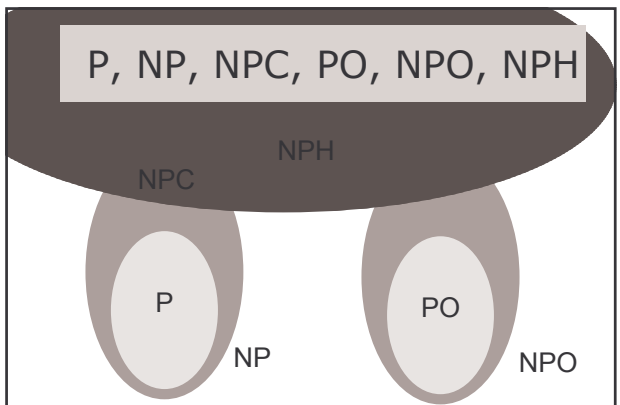
$$\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a)$$

(rozklad na podmnožiny se stejnou cenou)

NPH a NPC



P, NP, NPC, PO, NPO, NPH



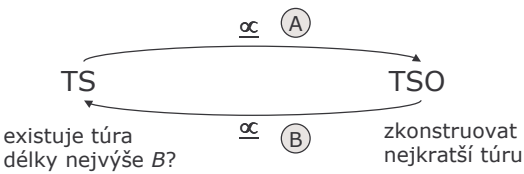
Turingova redukce
(Turingova transformace)

- Definice Turingovy redukce
  Rozhodovací problém Π_1 je Turing-redukovatelný na Π_2 ($\Pi_1 \propto \Pi_2$), jestliže existuje polynomiální program pro (deterministický) Turingův stroj, který řeší každou instanci I_1 problému Π_1 tak, že používá program M_2 pro problém Π_2 jako podprogram (jehož trvání považujeme za jeden krok).
- Jiné značení: $\lhd$ $\overset{T}{\propto}$

Třída NP-těžký
(NP-Hard, NPH)

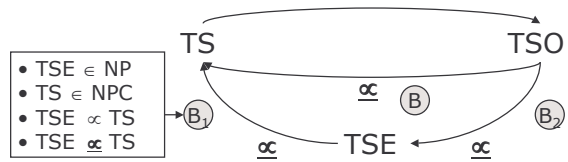
- Definice (třída NP-těžký):
  Problém Π je NP-těžký, jestliže pro všechny problémy $\Pi' \in$ NP, $\Pi' \propto \Pi$
- Karpova redukce je speciálním případem Turingovy redukce (volání podprogramu jednou, přímé použití výsledku)
- NPC $\subset$ NPH

Rozhodovací (TS) a optimalizační (TSO) verze TSP



A
- spočítat nejkratší túru pomocí TSO
- porovnat

Turingova redukce



TSE: Dána množina n měst $C=\{c_1,c_2,\ldots,c_n\}$. Pro každá dvě města c_i,c_j je dána vzdálenost $d(c_i,c_j)$. Dále dána mez B a cesta Θ procházející K městy. Dá se Θ prodloužit na túru délky nejvýše B?

TSO ∝ TSE

TSE: Dána množina n měst $C=\{c_1,c_2,\ldots,c_n\}$, vzdálenost $d(c_i,c_j)$. Dále mez B a cesta Θ procházející K městy. Dá se Θ prodloužit na túru délky $\leq B$?

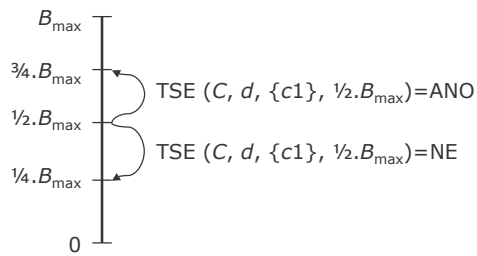
B_2

- Víme, že $B_{min}=n$, $B_{max}=n.\max\{d(ci,cj)\}$
- Velikost instance měřme $N = n+\log_2 B_{max}$
- Nechť existuje program TSE (C, d, Θ, B). Jak pomocí něj vyřeším TSO?
- 1. Určím B^* pomocí $\log_2 B_{max}$ volání TSE (C, d, $\{c_1\}$, B).
- 2. Určím další město k C_1 pomocí TSE (C, d, $\{c_1, c_i\}$, B^*).
- 3. Opakuji, až určím celou kružnici

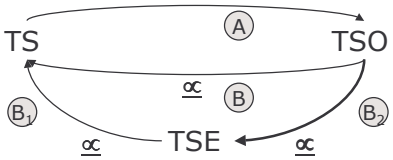
$$O(\log_2 B_{max}) + O(n^2) = O(N^2)$$

K předchozímu důkazu

1. Určím B^* pomocí $\log_2 B_{max}$ volání TSE (C, d, $\{c_1\}$, B).

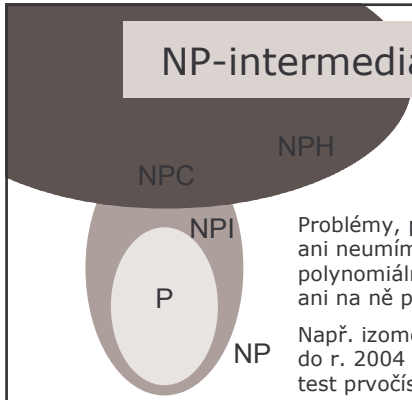


TS a TSO



TS a TSO jsou Turing-ekvivalentní a tedy stejně těžké.

NP-intermediate (NPI)



Problémy, pro které ani neumíme nalézt polynomiální algoritmus, ani na ně převést SAT.

Např. izomorfismus grafů, do r. 2004 také test prvočíselnosti