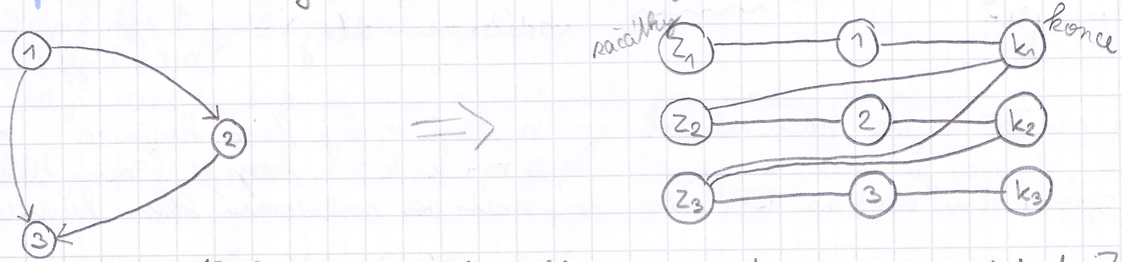
Tvrzení: Hamiltonovský cyklus $\triangleleft_p$ Hamiltonovská kružnice

Nástin převodu: Každý vrchol nahradíme trojicí vrcholů



Vidíme, že v grafu cyklus není, protože se nejde dostat do Z_1 a z K_3.

Tvrzení: Hamiltonovská kružnice $\triangleleft_p$ problém obchodního cestujícího (TSP)

Nástin převodu: V grafu TSP zavedeme ceny hran podle původního grafu $G=(V,E)$ následovně:

$d(i,j) = 1$ iff $\{v_i, v_j\} \in E$

$d(i,j) = 2$ iff $\{v_i, v_j\} \notin E$

Obchodní cestující najde trasu. Pokud je cena trasy rovna počtu vrcholů, pak v G existuje Ham. kružnice. Pokud je trasa dražší, kružnice neexistuje.

Tvrzení: Hamiltonovský cyklus $\triangleleft_p$ problém otevřené Ham. cesty ← bude na writku (ale já chyběla)

Tvrzení: Otevřená orientovaná Ham. cesta $\triangleleft_p$ hledání nejdelší cesty

Nástin převodu: $G=(V,E)$ $\rightsquigarrow$ $G'=(V',E')$, $c: E' \to \mathbb{Z}$, k

existuje otevřená Ham. cesta? existuje cesta délky aspoň k?

Vezmeme $G' = G$, dáme $c(e)=1$, $k = |V|-1$. Cesta být delší nemůže, a najdeme ji, pokud existuje otevřená Hamiltonovská cesta.

Tvrzení: Nejdelší cesta $\triangleleft_p$ problém nejkratší cesty

Nástin převodu: $c'(e) = -c(e)$, $k' = -k$

Problém 3dimenzionálního párování: Máme 3 množiny X, Y, Z, všechny mají velikost n. Máme množinu S trojic prvků z $X \times Y \times Z$, otázka zní:

$\exists A \subseteq S$, $|A| = n$, iff $(x,y,z) \neq (x',y',z')$ tak $x \neq x'$, $y \neq y'$, $z \neq z'$

Když je problém NP úplný, nehledáme už optimální řešení, ale snažíme se najít řešení aspoň dostatečně přesné.

Přednáška 23.4.2019

Heuristika pro vrcholové pokrytí 1: do pokrytí přidáváme vrcholy s nejvyšším stupněm. Pokud je v pokrytí vrchol, který už má pokryté všechny hrany, tak ho odstraníme.
- pokud je v grafu pokrytí o k vrcholech, tento alg najde pokrytí o $\Theta(k \cdot \lg(k))$ vrcholech.

Heuristika pro vrcholové pokrytí 2: vybíráme hrany a do pokrytí přidáme oba krajní vrcholy. Z E odebereme všechny hrany incidentní s u, v.

Tvrzení: Označme C_{min} minimální vrcholové pokrytí, a C pokrytí nalezené heuristikou 2. Platí: $|C| \leq 2 \cdot |C_{min}|$

Důkaz: F je množina vybraných hran, a žádné dvě hrany z F nemají společný vrchol.
$|C| = 2 \cdot |F|$, $|C_{min}| \geq F$ z toho: $2 \cdot |C_{min}| \geq 2 \cdot |F| = |C|$.

Definice: Je dána optimalizační úloha U s účelovou fcí c. Polynomiální alg. A je <u>R-aproximační algoritmus</u> pro U, jestliže $\exists R \in \mathbb{R}$, že pro každou instanci I najde algoritmus A přípustné řešení $A(I)$ s hodnotou účelové fce ne horší než R-krát hodnota účelové fce optimálního řešení.

Tvrzení: Kdybychom měli r-aproximační algoritmus na TSP, tak $P = NP$.

Dk: $G = (V, E)$, existuje Ham. kružnice? $\rightsquigarrow$ TSP, města $\{1, \dots, n\}$

$V = \{1, \dots, n\}$ vzdálenosti $d(i,j) = \begin{cases} 1 & \text{iff } \{i,j\} \in E \\ n \cdot r + 1 & \text{iff } \{i,j\} \notin E \end{cases}$

$E = \{ \dots \}$

Optimální alg na TSP najde trasu velikosti $\begin{cases} n, & \text{když v } G \text{ je Ham. kružnice} \\ n \cdot r + k,\ k \geq 1, & \text{když v } G \text{ není H. k.} \end{cases}$

Máme-li r-aproximační alg na TSP, můžu ho použít na rozhodování Ham. kružnice, tedy $P = NP$.

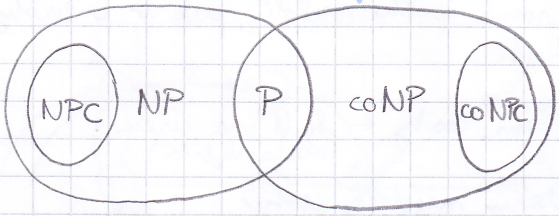
$d(i,j) \leq d(i,k) + d(k,j)$

Věta: Pokud se omezíme na grafy splňující Δ nerovnost, pak existuje 2-aproximační alg.

- najdeme min kostru a vypíšeme průchod kostrou.
- z průchodu odstraníme duplicitní vrcholy

} detailněji to je v KO

Definice: Jazyk L patří do třídy co-NP, jestliže jeho doplněk $\overline{L}$ patří do NP.
Rozhodovací úloha U patří do třídy co-NP, jestliže L_U patří do co-NP.



Zjevně $P = coP$, prostě v TM vyměníme koncové a nekoncové stavy, čas se nezmění

není známo, zda $NP = coNP$

Lemma: Máme dva jazyky L_1, L_2, pro které platí $L_1 \triangleleft_p L_2$. Pak taky $\overline{L_1} \triangleleft_p \overline{L_2}$.

Dk: Označme $L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Pi^*$. Když $L_1 \triangleleft_p L_2$, znamená to, že máme alg / TM / program, který pro každé $u \in \Sigma^*$ zkonstruuje $v \in \Pi^*$ tak, že $u \in L_1$ iff $v \in L_2$.

$$u \in L_1 \text{ iff } v \in L_2 \Rightarrow u \notin L_1 \text{ iff } v \notin L_2 \Rightarrow u \in \overline{L_1} \text{ iff } v \in \overline{L_2}$$

Tvrzení: Platí $coNP = NP$ právě tehdy, když existuje NP úplný jazyk L, pro který $\overline{L} \in NP$.

Dk: ($\Rightarrow$) Když $co\text{-}NP = NP$, tak každý jazyk má vlastnost, že $\overline{L} \in NP$

($\Leftarrow$) Když existuje $L_u \in NPC$ tak, že $\overline{L_u} \in NPC$, tak $NP = coNP$?

1) Platí $NP \subseteq coNP$?
$L \in NP$ libovolné $\Rightarrow L \triangleleft_p L_u$, tj $\overline{L} \triangleleft_p \overline{L_u} \in NP$, proto $\overline{L} \in NP$, $L \in coNP$

2) Platí $coNP \subseteq NP$?
Vezmeme libovolné $L_1 \in coNP$, z definice $\overline{L_1} \in NP$. Jelikož $\overline{L_1} \triangleleft_p L_u$,
$\overline{L_1} \triangleleft_p L_u \Rightarrow L_1 \triangleleft_p \overline{L_u} \Rightarrow L_1 \in NP \Rightarrow coNP \subseteq NP$

Obě implikace platí.

Definice: Jazyk L patří do Pspace, jestliže existuje deterministický TM, který přijímá jazyk L a pracuje s polynomiální paměťovou složitostí.

Definice: Jazyk L patří do NPspace, jestliže existuje nedeterministický TM, který přijímá L a pracuje s polynomiální paměťovou složitostí.

Tvrzení: $P \subseteq Pspace$, $NP \subseteq NPspace$

Věta: Je dán TM M, který přijímá L s paměťovou složitostí $p(n)$, kde p je polynom. Pak existuje konstanta c taková, že M přijme slovo w délky n po nejvýše $c^{p(n)+1}$ krocích.

důkaz by fungoval pro libovolnou fci $p(n)$, nejen polynom

Dk: M přijímá L s $p(n)$, proto pro $w \in L$: $q_0 w \vdash^* \alpha q_F \beta$
- pokud jde o TM, tak se výpočet nikdy v jedné konfiguraci neopakuje
- pokud jde o NTM, tak se výpočet může do nějaké konfigurace vrátit, ale v tom případě můžeme kus výpočtu zahodit.

$\Rightarrow$ když $w \in L$, tak existuje výpočet, kde jsou všechny ID různé.

Kolik je různých ID? $\langle$ stav $\times$ poloha hlavy $\times$ slovo na pásce $\rangle$, $s = |Q|$, $t = |\Gamma|$

$$\underbrace{\text{stav}}_{s} \cdot \underbrace{\text{poloha hlavy}}_{p(n)} \cdot \underbrace{\text{slovo na pásce}}_{t^{p(n)}}$$

Horní odhad je $s \cdot p(n) \cdot t^{p(n)}$, zkusíme zvolit $c = s+t$

$$(s+t)^{p(n)+1} = t^{p(n)+1} + \binom{p(n)+1}{1} \cdot t^{p(n)} \cdot s + \sum_{i=1}^{p(n)+1} \binom{p(n)+1}{i} t^{p(n)+1-i} \cdot s^i \geq$$

$$\geq p(n)+1 \; t^{p(n)} \cdot s > p(n) \cdot s \; t^{p(n)}$$ $\Rightarrow$ Ano, zvolíme $c = s+t$

Věta: Jestliže jazyk L patří do Pspace (NPspace), pak L je rozhodován deterministickým (nedeterministickým) TM M s polynomiální paměťovou složitostí, který se zastaví na každém slově délky n po maximálně $c^{q(n)}$ krocích, kde $q(n)$ je polynom a c je vhodná konstanta.

Dk: Sestavím TM se dvěma páskami: páska 1 – stejná jako páska M
páska 2 – počítá kroky na první pásce

Když udělá první páska víc než $c^{p(n)+1}$ kroků, tak:
- v NTM ukončím výpočet a zkusím jinou větev
- v TM ukončím výpočet, M slovo nepřijímá

Stroj převedu na jednopáskový (paměť se nemění, jen bude víc stop). Čas bude:
$O((c^{p(n)+1})^2)$, což je $O(c^{2p(n)} \cdot c^2) = O(c^{2p(n)})$
Čas je shora omezen $T(n) \leq d \cdot c^{2p(n)} = c^{\log_c d} \cdot c^{2p(n)} = c^{2p(n)+\log_c d} = c^{q(n)}$
kde $q(n) = 2p(n) + \log_c d$, což je polynom, když $p(n)$ je polynom.

Savitchova věta: Pspace = NPspace

Dk: To, že Pspace $\subseteq$ NPspace, je jasné. Chceme ukázat, že NPspace $\subseteq$ Pspace.
Když $L \in$ NPspace, pak existuje NTM rozhodující L s polynomiální paměťovou složitostí s počtem kroků $c^{q(n)}$, kde $|w| = n$, $w \in L$, $m := c$

Procedura REACH: vrací True, když $I \vdash^* J$ v nejvýš m krocích, False v opačném případě

```
REACH (I, J, m)
  if m = 1:
    if I = J or I ⊢ J:
      return True
    else:
      return False
  else:
    for K in all_possible_IDs:
      if REACH (I, K, ⌊m/2⌋) and REACH (K, J, ⌈m/2⌉):
        return True
    return False
```

Začneme v $I = ID_0 = q_0 w$, chceme se dostat do $J = \alpha q_F \beta$.

Zavoláme proceduru REACH$(I, J, c^{q(n)})$, pokud vrátí True, máme vyhráno, jinak zkusíme jiné α, β (a těch je polynom.).

Čas výpočtu je nám jedno, počítáme paměť. Jedna konfigurace (ID) zabere délku $\Theta(p(n))$.

ID se v paměti ukládá ze 3 délek: $\alpha \,|\, q \,|\, \beta$, na pásce jsou uloženy trojice ID, ID, m:
$I_1, J_1, m/2$; $I_2, J_2, m/4$; … . Těchto trojic je nejvýš polynomiálně, a každá z nich je polynomiální, takže zůstaneme v polynomiální paměti.