- Trieda P: problémy riešiteľné v deterministickom poly čase
  Trieda NP: problémy riešiteľné v nedet. poly čase
- O časti problémov v NP si všetci myslia, že sa nedajú riešiť v deterministickom poly čase
  (napríklad TSP-D)
- NP-úplné problémy: Ak vyriešime v deterministickom polynomiálnom čase, všetky problémy v NP budeme vedieť riešiť v deterministickom polynomiálnom čase.
- Cookova veta: SAT je NP-úplný
- Ďalšie problémy pomocou redukcií

Cookova veta

SAT splniteľnosť: Uvažujme logickú formulu f.
Problém: Existuje priradenie hodnôt premenných také, aby f bola splnená?

Veta: (Cook) SAT je NP-úplný

Na minulej prednáške: SAT$\in$NP
–alebo–
Existuje nedeterministický polynomiálny algoritmus, ktorý rieši SAT.

Dnes ukážeme: SAT je NP-ťažký
–alebo–
pre ľubovoľný problém $Q \in NP$, ak by sme vedeli riešiť problém SAT v deterministickom polynomiálnom čase, tak by sme vedeli vyriešiť aj Q v deterministickom polynomiálnom čase

(Q možno **polynomiálne zredukovať** na SAT)

SAT je NP-ťažký

Uvažujme $Q \in \mathrm{NP}$

$\Longrightarrow$existuje polynomiálny nedeterministický algoritmus, ktorý rieši Q

Ako taký algoritmus zapíšeme?

- Každý register má v sebe uložené číslo konštantnej veľkosti (registre označíme $R_1, R_2, \ldots$)
- Program je nemeniaca sa postupnosť príkazov s konštantným počtom očíslovaných riadkov
- Na začiatku je vstup uložený v prvých n registroch (n je veľkosť vstupu)
- Program beží nanajvýš $p(n)$ krokov
  a pristupuje najviac ku $q(n)$ prvým registrom
  ($p(n)$ a $q(n)$ sú polynómy závisiace od n)

- Sada inštrukcií:
  - ACCEPT
  - REJECT
  - GOTO m
  - IF $R_\ell = 0$ THEN GOTO m
  - CHOOSE R_l BETWEEN 0 AND 1
  - základné aritmetické operácie (napr. $R_\ell := R_u + R_v$, $R_\ell := R_u * R_v$)
  - nejaký mechanizmus na adresáciu prvých $q(n)$ registrov (detaily sú mierne komplikované, ale dá sa)

SAT je NP-ťažký: Q možno polynomiálne zredukovať na SAT

Chceme:

- Daný je program A, ktorý rieši problém Q v polynomiálnom čase a inštancia $x = x_1, x_2, \ldots, x_n$.
- Vyrobíme veľkú logickú formulu f, ktorá "simuluje" program A na vstupe x;
- A dosiahne ACCEPT $\iff$ f je splniteľná

Premenné formuly f:

- $Q[i, k]$ – v čase i program vykonáva riadok k
- $S[i, j, k]$ – v čase i má register R_j hodnotu k

Formula f bude konjunkcia ("AND") niekoľkých menších formúl t.j. všetky tieto menšie formuly musia byť splnené, aby formula f bola splnená

$\boxed{1}$ "V každom čase i program vykonáva práve jeden riadok."

$\boxed{\neg(Q[i,k] \wedge Q[i,\ell])}$ pre všetky i a $k \neq \ell$

$\boxed{2}$ "V každom čase i každý register obsahuje práve jednu hodnotu."

$\boxed{\neg(S[i,j,k] \wedge S[i,j,l])}$ pre všetky i, j a $k \neq \ell$

$\boxed{3}$ V čase 0:

- Program vykonáva riadok 1: $\boxed{Q[0,1]}$
- Prvých n registrov má hodnoty $x_1, \ldots, x_n$:
  $\boxed{S[0,1,x_1] \wedge S[0,2,x_2] \wedge \ldots \wedge S[0,n,x_n]}$
- Ostatné registre majú hodnotu 0:
  $\boxed{S[0,n+1,0] \wedge S[0,n+2,0] \wedge \ldots \wedge S[0,q(n),0]}$

$\boxed{4}$ "Po $p(n)$ krokoch program dosiahne riadok s inštrukciou ACCEPT"

$\boxed{Q[p(n),k]}$ k je riadok s inštrukciou "ACCEPT"

$\boxed{5}$ "Stav počítača sa mení v čase v súlade s programom."

k-**ty riadok**	**Formula**
ACCEPT alebo REJECT	$Q[i,k] \Rightarrow Q[i+1,k]$
GOTO ℓ	$Q[i,k] \Rightarrow Q[i+1,\ell]$
IF $R_\ell = 0$ THEN GOTO m	$Q[i,k] \wedge S[i,\ell,0] \Rightarrow Q[i+1,m]$ $Q[i,k] \wedge \neg S[i,\ell,0] \Rightarrow Q[i+1,k+1]$
CHOOSE R_ℓ	$Q[i,k] \Rightarrow Q[i+1,k+1] \wedge$ $(S[i+1,\ell,0] \vee S[i+1,\ell,1])$
atď. pre ďalšie inštrukcie	

SAT je NP-ťažký: zhrnutie

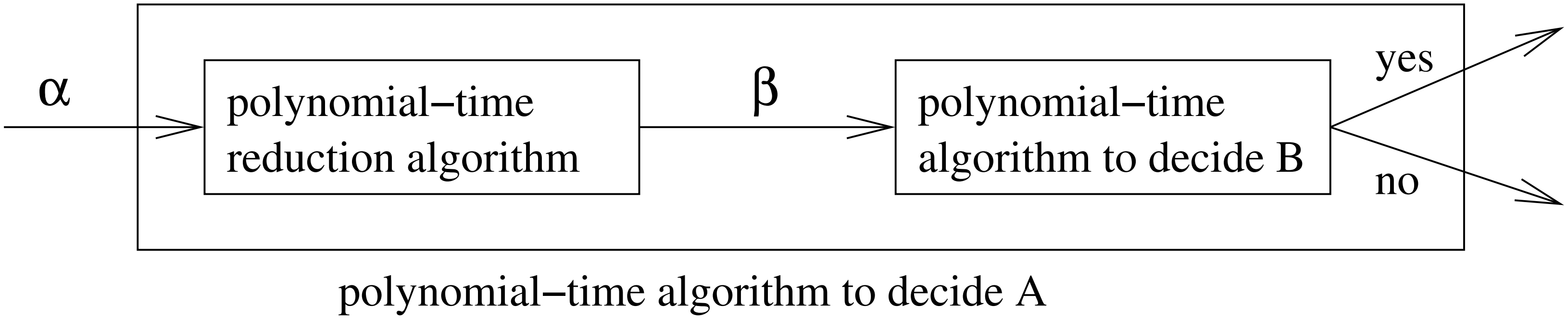
Polynomiálny algoritmus pre riešenie Q:

1. Skonštruuj formulu $f_{A,x}$ pre daný algoritmus A a vstup x
   - Det. polynomiálny čas v závislosti od n.
   - Výsledná formula má polynomiálnu veľkosť v závislosti od n.
   - $f_{A,x}$ je splniteľná $\Longleftrightarrow$ A akceptuje x
2. Zavolaj polynomiálne riešenie problému SAT na formulu f
3. Výsledok je súčasne riešením problému Q pre vstup x

$\Longrightarrow$**Ukázali sme: Ľubovoľný problém $Q \in NP$ možno polynomiálne zredukovať na SAT**

$\Longrightarrow$**SAT je NP-ťažký**

Polynomiálne redukcie



Hovoríme, že **problém A možno polynomiálne redukovať na problém B ($A \leq_p B$)**

- Ak by sme vedeli B riešiť v det. polynom. čase, vedeli by sme aj A riešiť v det. polynom. čase
- Ak neexistuje det. polynomiálne riešenie pre problém A, neexistuje ani det. polynomiálne riešenie pre problém B

Príklad: HAM $\leq_p$ TSP-D

Ako dokázať, že problém Q je NP-ťažký?

1. Vyberme si problém N o ktorom už vieme, že je NP-úplný
2. Ukážeme $N \leq_P Q$:
   - Navrhneme polynomiálny algoritmus, ktorý prerobí vstup x pre problém N na vstup $f(x)$ pre problém Q.
   - Dokážeme: Ak je x pozitívny vstup pre N, potom
     $f(x)$ je pozitívny vstup pre Q
   - Dokážeme: Ak je x negatívny vstup pre N, potom
     $f(x)$ je negatívny vstup pre Q
     —ALEBO—
     Ak $f(x)$ je pozitívny vstup pre Q, potom
     x je pozitívny vstup pre N
3. Keďže N je NP-úplný, Q musí byť NP-ťažký.

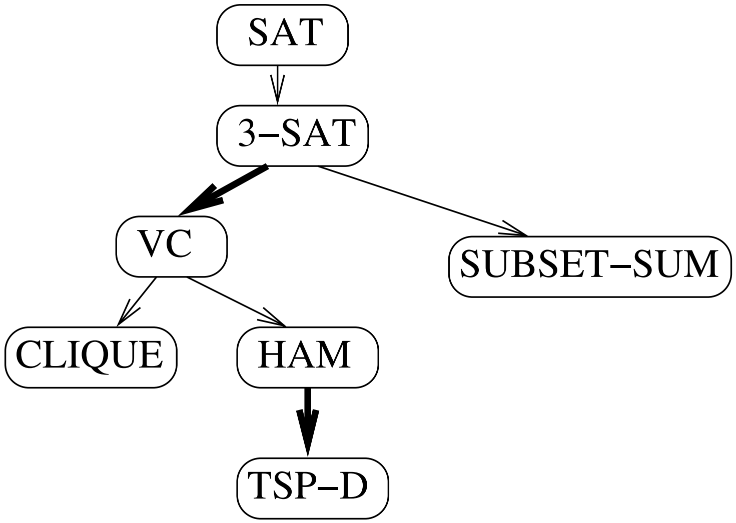
Dokončenie dôkazu NP-úplnosti: $Q \in NP$

4a Vytvoríme polynomiálny nedeterministický algoritmus riešiaci Q.

—ALEBO—

4b Pre každý vstup zadefinujeme certifikát polynomiálnej veľkosti.

5b Vytvoríme polynomiálny algoritmus, ktorý pre daný vstup x a certifikát y overí tento certifikát v polynomiálnom čase.



$A \longrightarrow B$ means reduction A to B

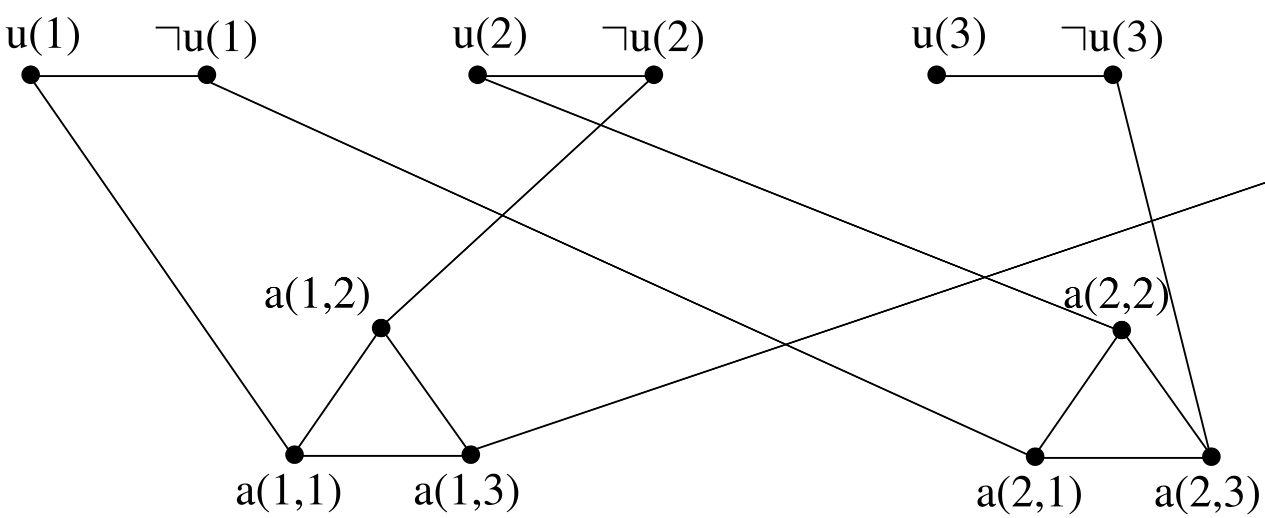
→ means reduction shown already

3-SAT$\leq_p$ VC-D

VC-D: existuje vrcholové pokrytie C veľkosti $|C| \leq c$?
(pre každú hranu aspoň jeden vrchol C)

- riešime 3-SAT pomocou VC-D
- pre formulu f musíme vytvoriť graf G a číslo c

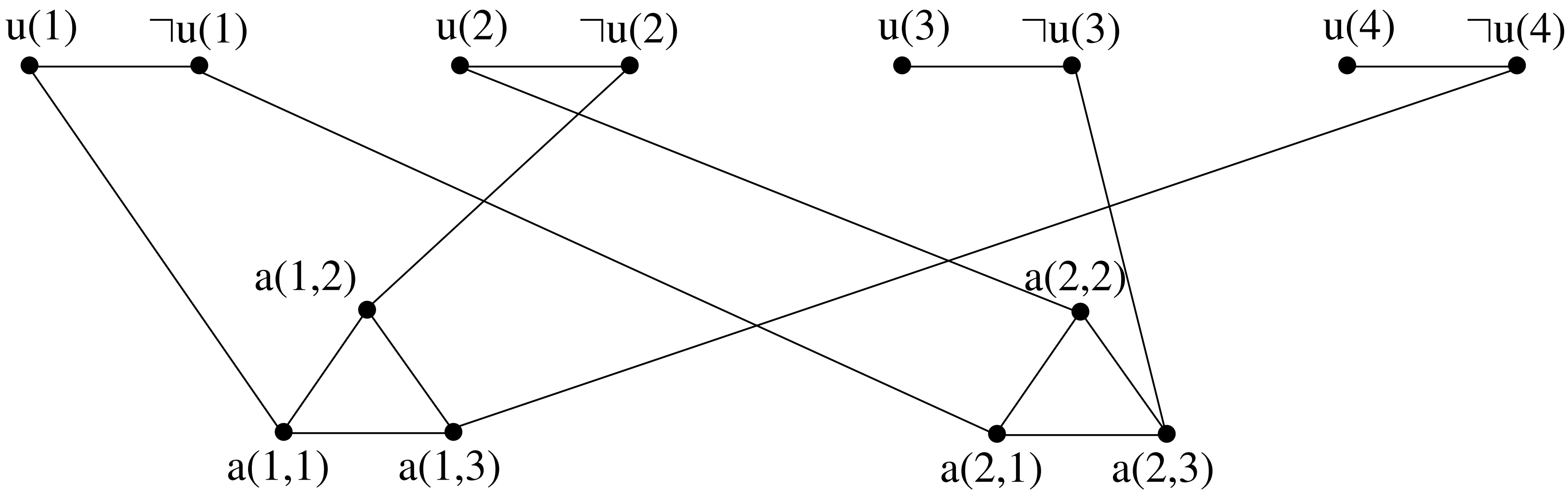
$(u_1 \vee \neg u_2 \vee \neg u_4) \wedge (\neg u_1 \vee u_2 \vee \neg u_3)$



$c = 2m + n$

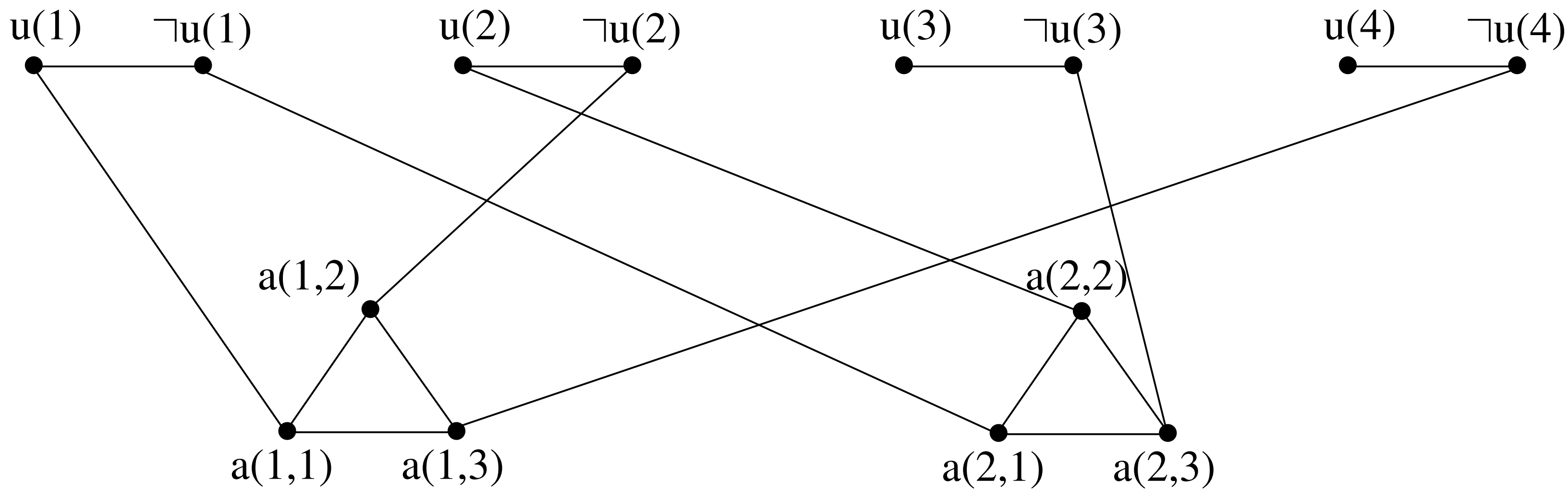
Lema: 3-SAT formula je splniteľná práve vtedy keď graf zostavený podľa horeuvedeného návodu má pokrytie o veľkosti $2m + n$

($\Rightarrow$) Ak je formula splniteľná, potom graf má pokrytie veľkosti $2m + n$



Lema: 3-SAT formula je splniteľná práve vtedy keď graf zostavený podľa horeuvedeného návodu má pokrytie o veľkosti $m + 2n$

($\Leftarrow$) Ak graf má pokrytie veľkosti $2m + n$, potom formula je splniteľná



Zhrnutie

- NP-ťažké / NP-úplné problémy: Problémy o ktorých sa domnievame, že sa nedajú riešiť v deterministrickom polynomiálnom čase.

  (Ak by sme ktorýkoľvek vyriešili, všetky problémy z NP by sa dali riešiť v polynomiálnom čase.)

- Cookova veta: SAT je NP-úplný problém.

- Dokazovanie NP-ťažkosti nového problému N:
  - Vyberieme si známy NP-úplný problém Z
  - Ukážeme $Z \leq_p N$
  - Keďže pre všetky problémy $Q \in \mathrm{NP}$ tiež $Q \leq_p Z$ $\Rightarrow Q \leq_p N$

- Pozor na smer dokazovania!