Elfde college complexiteit

23 april 2019

NP-volledigheid III

Als voorbeeld bekijken we het Travelling Salesman/person Problem, ofwel het Handelsreizigersprobleem **TSP.** Hiervoor geldt:

$$\text{TSP} \in \mathcal{NP}$$

Gegeven een volledige*, ongerichte graaf $\mathcal{G} = (V, E)$ met gewichten op de takken, en een geheel getal $k \geq 0$. Bestaat er in $\mathcal{G}$ een Hamiltonkring met totaalgewicht $\leq k$?
De invoer van het probleem is dus $x = \langle \mathcal{G}, k \rangle$.

*tussen elk tweetal knopen van $\mathcal{G}$ zit een tak

Terzijde: het maakt niet uit voor de polynomialiteit van het algoritme welke representatie gekozen is voor de invoer.

- Neem aan dat $V = \{1, 2, \ldots, n\}$, dus het aantal knopen $= n$. We nemen nu voor $\mathcal{G}$ de adjacency-list representatie, en voor k de binaire of decimale representatie (bijvoorbeeld). Dan is zoeken van een tak met bijbehorend gewicht $O(|V|) \subseteq O(|x|)$.

- Hetzelfde als hierboven, maar nu nemen we voor $\mathcal{G}$ de adjacency-matrix representatie. Dan is zoeken van een tak met bijbehorend gewicht $O(1)$.

- (denkend aan een Turingmachine) representeer $\langle \mathcal{G}, k \rangle$ als een rij van knopen, gevolgd door de takken met gewichten, gevolgd door k, alles binair of decimaal. Dan is zoeken van een tak met bijbehorend gewicht $O(|x|)$.

We gaan hier uit van de eerste optie.

Een polynomiaal begrensd *niet-deterministisch algoritme* voor TSP:

1. Fase 1 (gokfase)
   Er wordt een string s gegenereerd, hierna te interpreteren als een rij gehele getallen.
2. Fase 2 (verificatiefase)
   Er wordt gecontroleerd of s een Hamiltonkring voorstelt met gewicht $\leq k$:
   (1) controleer of er precies $|V|$ integers staan: $O(|s|)$
   (2) controleer of elke integer tussen 1 en $|V|$ zit: $O(|s|)$
   (3) controleer of alle knopen uit s verschillen: $O(|s|^2)$
   (4) controleer of het totale gewicht van de Hamiltonkring (dat stelt s voor als aan (1)–(3) voldaan is) $\leq k$ is: $O(|s| \cdot |x|)$ (want . . .)

Als de vier tests positief zijn wordt True geretourneerd, zodra een test negatief uitvalt wordt False teruggegeven (of er wordt in een oneindige loop gegaan, of . . .).

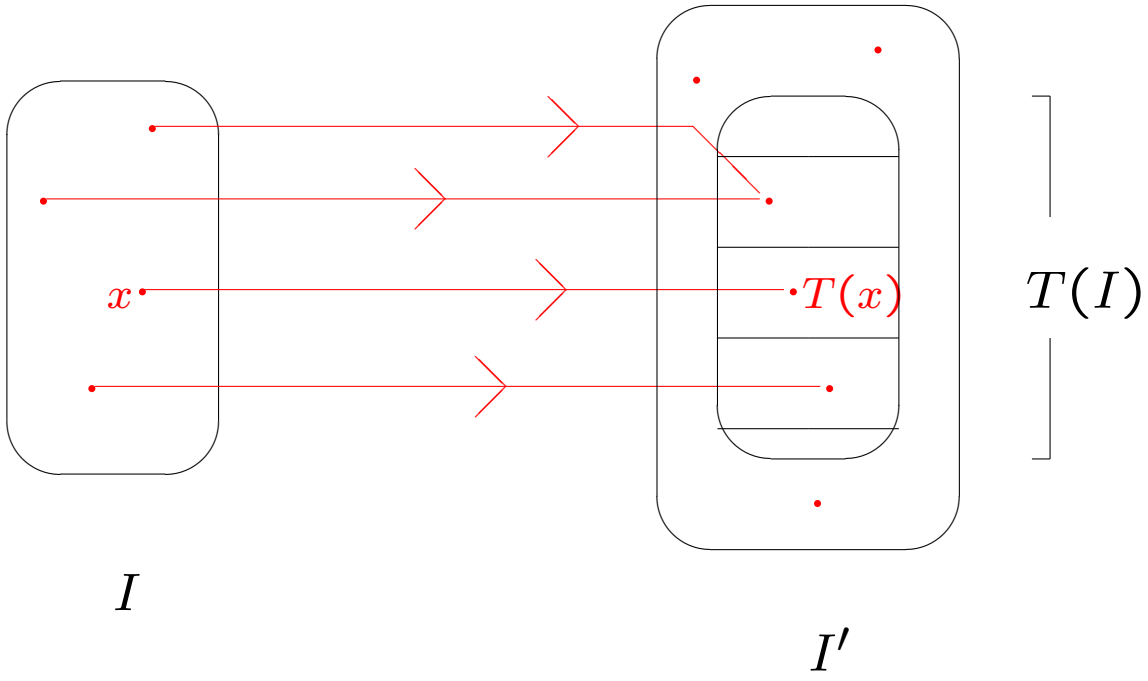
3. Fase 3 (uitvoerfase)
   Als fase 2 True oplevert wordt "ja" uitgevoerd, anders geen uitvoer.

Voor het hierboven beschreven algoritme geldt:

- Het antwoord van A op invoer $x = \langle \mathcal{G}, k \rangle$ is "ja" $\iff$ er bestaat een string s waarvoor Fase 2 True geeft $\iff$ er bestaat een string s die een Hamiltonkring voorstelt met gewicht $\leq k \iff \mathcal{G}$ heeft een Hamiltonkring met gewicht $\leq k \iff x$ is een ja-instantie voor TSP

- Het algoritme is polynomiaal, want voor een ja-executie stelt s een (goede) Hamiltonkring voor, dus dan is $|s| \in O(|V|) \subseteq O(|x|)$ en derhalve is Fase 2 dan $O(|x|^2)$

Zij T een functie van de invoerverzameling I van een beslissingsprobleem P naar de invoerverzameling I' van een beslissingsprobleem Q.
T beeldt dus elke $x \in I$ af op een $T(x) \in I'$.



Definitie

T heet een polynomiale reductie (of *polynomiale transformatie*) van P naar Q als geldt:

1. T kan berekend worden in polynomiaal begrensde tijd (als functie van $|x|$). D.w.z.: de constructie van $T(x)$ uit x kan in $O(|x|^k)$ stappen in de worst case ($k \geq 0$).

2. Voor elke x uit I geldt: als x een ja-instantie is voor P dan is $T(x)$ een ja-instantie voor Q.

3. Voor elke x uit I geldt: als x een nee-instantie is voor P dan is $T(x)$ een nee-instantie voor Q.

3′. Voor elke x uit I geldt: als $T(x)$ een ja-instantie is voor Q dan is x een ja-instantie voor P.
(Dit is equivalent met 3.)

Definitie

Een probleem P is polynomiaal reduceerbaar (of polynomiaal transformeerbaar) naar Q als er een polynomiale reductie bestaat van P naar Q.

Notatie: $P \leq_{\mathsf{P}} Q$.

Stelling

Als $P \leq_{\mathsf{P}} Q$ en Q zit in $\mathcal{P}$, dan zit P ook in $\mathcal{P}$.

De notatie $P \leq_{\mathsf{P}} Q$ betekent dat er een polynomiale reductie T van P naar Q bestaat:

1. T beeldt elke invoer* x van beslissingsprobleem P af op een invoer $T(x)$ van beslissingsprobleem Q.

2. De constructie van $T(x)$ uit x is polynomiaal: $O(|x|^k)$.

3. Reductie-eigenschap: voor elke x uit I (= invoerverzameling van P) geldt: x is een ja-instantie voor $P \iff T(x)$ is een ja-instantie voor Q.

*= instantie

HC1: **gegeven** een *gerichte* graaf $\mathcal{G} = (V, E)$.
Vraag: heeft $\mathcal{G}$ een Hamiltonkring?

HC2: **gegeven** een *ongerichte* graaf $\mathcal{G} = (V, E)$.
Vraag: heeft $\mathcal{G}$ een Hamiltonkring?

Bewering: HC1 $\leq_P$ HC2: zie volgende sheet.

Opmerking: er geldt ook: HC2 $\leq_P$ HC1. Bedenk zelf een eenvoudige reductie.

Een reductie van HC1 naar HC2

Transformatie T die een gerichte graaf $\mathcal{G} = (V, E)$ op een ongerichte graaf $T(\mathcal{G}) = \mathcal{G}' = (V', E')$ afbeeldt:

- $V' = \{v_1, v_2, v_3 : v \in V\}$: elke knoop $v \in V$ wordt afgebeeld op een drietal knopen v_1, v_2, v_3.
- $E' = \{(v_1, v_2), (v_2, v_3) : v \in V\} \cup \{(v_3, w_1) : (v, w) \in E\}$: binnen elk drietal knopen corresponderend met v loopt een tak tussen v_1 en v_2 en tussen v_2 en v_3, en voor elke tak (= pijl) van v naar w in $\mathcal{G}$ komt een tak in $\mathcal{G}'$ tussen v_3 en w_1.

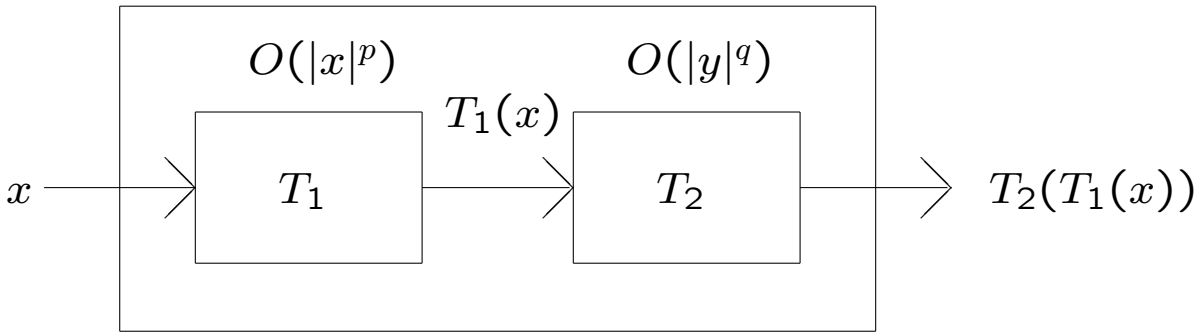
Dan geldt

1. T kan in polynomiaal begrensde tijd berekend worden (constructie van $T(\mathcal{G})$ uit $\mathcal{G}$ kan zeker in $O(|\mathcal{G}|^q)$ (met $q = 2$ bijvoorbeeld))
2. $\mathcal{G}$ is een ja-instantie voor HC1 $\iff$ $T(\mathcal{G})$ is een ja-instantie voor HC2, ofwel: $\mathcal{G}$ heeft een gerichte Hamiltonkring $\iff$ $\mathcal{G}'$ heeft een ongerichte Hamiltonkring

Lemma

$\leq_{\mathsf{P}}$ is transitief, dat wil zeggen: als $P_1 \leq_{\mathsf{P}} P_2$ en $P_2 \leq_{\mathsf{P}} P_3$ dan is $P_1 \leq_{\mathsf{P}} P_3$.

Het **bewijs**:



- De samenstelling van T_1 en T_2 is polynomiaal begrensd omdat T_1 en T_2 dat zijn: $O(|x|^{pq})$ (let op: NIET $O(|x|^{p+q})$!)
- $T_2 \circ T_1(x) = T_2(T_1(x))$ is een ja-instantie van $P_3 \iff T_1(x)$ is een ja-instantie van $P_2 \iff x$ is een ja-instantie van P_1
  (volgt uit de reductie-eigenschap van T_2 en T_1)

Definitie
Een probleem Q is NP-hard (ook wel: NP-moeilijk) als elk probleem P in $\mathcal{NP}$ polynomiaal reduceerbaar is tot Q, dat wil dus zeggen dat $P \leq_{\mathsf{P}} Q$ voor alle $P \in \mathcal{NP}$.

Definitie
Een probleem Q is NP-volledig als

1. $Q \in \mathcal{NP}$

2. Q is NP-hard

Notatie
De klasse van NP-volledige problemen geven we aan met $\mathcal{NPC}$ (NP-complete).

Stelling*

Stel Q is een probleem waarvoor geldt dat $P \leq_{\mathsf{P}} Q$ voor een of andere $P \in \mathcal{NPC}$. Dan is Q NP-hard.
Als bovendien $Q \in \mathcal{NP}$, dan geldt dat $Q \in \mathcal{NPC}$.

Dus door een bekend NP-volledig probleem te reduceren tot Q reduceren we impliciet alle problemen uit $\mathcal{NP}$ tot Q. Dit geeft ons derhalve een methode om aan te tonen dat een probleem Q NP-volledig is.

*bewijs op college

1. Bewijs dat $Q \in \mathcal{NP}$
2. Kies een bekend NP-volledig probleem P
3. Toon aan dat $P \leq_{\mathsf{P}} Q$

Stap 3 valt uiteen in:

3a. Geef een functie T van I (de invoerverzameling van P) naar I' (de invoerverzameling van Q) die elke $x \in I$ afbeeldt op een element $T(x)$ van I'

3b. Laat zien dat $T(x)$ uit x geconstrueerd kan worden in polynomiaal begrensde tijd ($O(|x|^k)$ voor zekere $k \geq 0$)

3c. Toon aan dat T voldoet aan: $x \in I$ is een ja-instantie voor $P \iff T(x) \in I'$ is een ja-instantie voor Q

In 1971 bewees Stephen Cook op een directe manier (dus door een reductie te geven van alle problemen uit $\mathcal{NP}$ naar SAT) dat SAT NP-volledig is. (Zie volgend college.)

Stelling
Gegeven een *willekeurig* probleem $P \in \mathcal{NP}$. Dan is P reduceerbaar tot SAT: $P \leq_{\mathsf{P}}$ SAT.

Sindsdien is met behulp van de reductiemethode van zeer veel bekende problemen aangetoond dat ze NP-volledig zijn. Bijvoorbeeld voor enige voorbeeldproblemen:

$$\text{SAT} \leq_{\mathsf{P}} \text{3SAT} \leq_{\mathsf{P}} \text{Kliek} \leq_{\mathsf{P}} \text{VC}$$
$$\text{3SAT} \leq_{\mathsf{P}} \text{HC2} \leq_{\mathsf{P}} \text{TSP}$$
$$\text{SAT} \leq_{\mathsf{P}} \text{3Kleur} \leq_{\mathsf{P}} \text{4Kleur}$$

3SAT

Gegeven een logische formule ϕ in 3-CNF. Bestaat er een waardering die ϕ True maakt?

Definitie

Een logische formule ϕ staat in 3-CNF als ϕ een conjunctie is van clausules, waarbij elke clausule een disjunctie is van drie verschillende (*) literals.

(*) deze eis wordt ook wel weggelaten bij de definitie van 3SAT

variabele: x_5, x_7, ...; literal: x_3, $\neg x_6$, ...; clausule: $x_5 \vee \neg x_6 \vee x_8$, ...

Kliek

Gegeven een ongerichte graaf $\mathcal{G} = (V, E)$ en een geheel getal k $(0 \leq k \leq |V|)$. Is er in $\mathcal{G}$ een kliek = clique ter grootte k?

Definitie

Een kliek in een ongerichte graaf $\mathcal{G} = (V, E)$ is een deelverzameling $V' \subseteq V$ zodanig dat voor elk tweetal knopen $u, v \in V'$ $(u \neq v)$ geldt dat $(u, v) \in E$. (Oftewel: tussen elk tweetal knopen uit V' zit een tak.)

Er geldt: 3SAT $\leq_P$ Kliek. Om dit aan te tonen moeten we een logische formule in 3-CNF afbeelden op een invoer voor Kliek, dus op een ongerichte graaf en een geheel getal.

Zij ϕ een logische expressie (formule) in 3-CNF, met zeg m clausules: $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$. Hierin is $C_r = \ell_1^r \vee \ell_2^r \vee \ell_3^r$ $(r = 1, \ldots, m)$ en ℓ_1^r, ℓ_2^r en ℓ_3^r steeds verschillend bij vaste r.

Construeer nu een ongerichte graaf $\mathcal{G}_\phi = (V, E)$ als volgt.

Voor elke clausule C_r uit ϕ doen we 3 knopen v_1^r, v_2^r en v_3^r in V (deze corresponderen met ℓ_1^r, ℓ_2^r en ℓ_3^r). $\mathcal{G}_\phi$ heeft dus $3m$ knopen.

Er komt een tak tussen twee knopen v_i^r en v_j^s als:

- v_i^r en v_j^s in verschillende drietallen zitten (dus $r \neq s$), en
- de bijbehorende ℓ_i^r en ℓ_j^s zó zijn dat $\ell_i^r \neq \neg\ell_j^s$, met andere woorden: ℓ_i^r en ℓ_j^s zijn niet elkaars negatie

Definieer nu de transformatie T als: $T(\phi) = \langle \mathcal{G}_\phi, m \rangle$.
Dan geldt:

- De constructie van $T(\phi) = \langle \mathcal{G}_\phi, m \rangle$ uit ϕ kan in $O(|\phi|^q)$ stappen (polynomiaal dus).

- Er is een waardering die ϕ waarmaakt $\iff$ $\mathcal{G}_\phi$ heeft een kliek ter grootte m.

Laat $\phi = C_1 \wedge C_2 \wedge C_3$, met $C_1 = x_1 \vee \neg x_2 \vee \neg x_3$, $C_2 = \neg x_1 \vee x_2 \vee x_3$ en $C_3 = x_1 \vee x_2 \vee x_3$. Hier is dus $m = 3$.

Dan $v_1^1 \leftrightarrow x_1, v_2^1 \leftrightarrow \neg x_2, v_3^1 \leftrightarrow \neg x_3$; alle uit clausule C_1, etcetera

- een waardering w die ϕ waarmaakt is bijvoorbeeld: $w(x_1) = w(x_2) =$ False en $w(x_3) =$ True. Een bijbehorende kliek in $\mathcal{G}_\phi$ ter grootte 3 is dan $\{v_2^1, v_3^2, v_3^3\}$ (*).

- een kliek ter grootte 3 in $\mathcal{G}_\phi$ is bijvoorbeeld $\{v_1^1, v_2^2, v_2^3\}$. Een bijbehorende waardering is $w(x_1) = w(x_2) = w(x_3) =$ True. Deze maakt ϕ waar.

(*) De bovenindex geeft aan met welke clausule een knoop correspondeert. Uit elk drietal (clausule) één knoop (literal).

SAT

Gegeven een logische formule ϕ in CNF. Bestaat er een waardering van de in ϕ voorkomende logische variabelen die ϕ True maakt?

Definitie

Een logische formule ϕ staat in Conjunctive Normal Form als ϕ een conjunctie is van clausules, waarin een clausule een disjunctie is van literals.

3SAT

Gegeven een logische formule ϕ in 3-CNF. Bestaat er een waardering die ϕ True maakt?

Definitie

Een logische formule ϕ staat in 3-CNF als ϕ een conjunctie is van clausules, waarbij elke clausule een disjunctie is van drie verschillende (*) literals.

(*) deze eis wordt ook wel weggelaten bij de definitie van 3SAT (het maakt niet uit)

Er geldt: SAT $\leq_P$ 3SAT. Om dit aan te tonen moeten we een logische formule ϕ in CNF afbeelden op een logische formule ϕ' in 3-CNF. We gaan er voor het gemak van uit dat de $\ell_1, \ell_2, \ldots, \ell_k$ per clausule al verschillend zijn (kan in $O(|\phi|^2)$ worden bewerkstelligd). Op clausuleniveau werkt de transformatie T als volgt:

$$\ell_1 \longrightarrow (\ell_1 \vee \widetilde{\ell_2} \vee \widetilde{\ell_3}) \wedge (\ell_1 \vee \widetilde{\ell_2} \vee \neg\widetilde{\ell_3}) \wedge (\ell_1 \vee \neg\widetilde{\ell_2} \vee \widetilde{\ell_3}) \wedge (\ell_1 \vee \neg\widetilde{\ell_2} \vee \neg\widetilde{\ell_3})$$

$$\ell_1 \vee \ell_2 \longrightarrow (\ell_1 \vee \ell_2 \vee \widetilde{\ell_3}) \wedge (\ell_1 \vee \ell_2 \vee \neg\widetilde{\ell_3})$$

$$\ell_1 \vee \ell_2 \vee \ell_3 \longrightarrow \ell_1 \vee \ell_2 \vee \ell_3$$

$$\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4 \longrightarrow (\ell_1 \vee \ell_2 \vee \widetilde{\ell_5}) \wedge (\ell_3 \vee \ell_4 \vee \neg\widetilde{\ell_5})$$

$$\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4 \vee \ell_5 \longrightarrow (\ell_1 \vee \ell_2 \vee \widetilde{\ell_6}) \wedge (\ell_3 \vee \neg\widetilde{\ell_6} \vee \widetilde{\ell_7}) \wedge (\ell_4 \vee \ell_5 \vee \neg\widetilde{\ell_7})$$

En in het algemeen voor $k \geq 4$:

$$\ell_1 \vee \ell_2 \vee \ldots \vee \ell_{k-1} \vee \ell_k \longrightarrow (\ell_1 \vee \ell_2 \vee \widetilde{\ell_{k+1}}) \wedge (\ell_3 \vee \neg\widetilde{\ell_{k+1}} \vee \widetilde{\ell_{k+2}}) \wedge (\ell_4 \vee \neg\widetilde{\ell_{k+2}} \vee \widetilde{\ell_{k+3}}) \wedge \ldots \wedge (\ell_{k-2} \vee \neg\widetilde{\ell_{2k-4}} \vee \widetilde{\ell_{2k-3}}) \wedge (\ell_{k-1} \vee \ell_k \vee \neg\widetilde{\ell_{2k-3}})$$

Hierin zijn $\widetilde{\ell_{k+1}}, \widetilde{\ell_{k+2}}, \ldots, \widetilde{\ell_{2k-3}}$ steeds nieuwe, frisse logische variabelen.
Een clausule met k (verschillende) literals wordt zo getransformeerd in een conjunctie van $k-2$ clausules met elk 3 verschillende literals.
Het beeld van een conjunctie van clausules definiëren we als een conjunctie van de beelden van de samenstellende clausules:

$$\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m \longrightarrow T(C_1) \wedge T(C_2) \wedge \ldots \wedge T(C_m) = T(\phi)$$

Voor deze transformatie T geldt:

- De constructie van $T(\phi)$ uit ϕ kan met een polynomiaal begrensd ($= O(|\phi|^q)$) algoritme.

- ϕ is een ja-instantie van SAT $\Longleftrightarrow$ $T(\phi)$ is een ja-instantie van 3SAT.

- Ofwel: er is een waardering die ϕ waarmaakt $\Longleftrightarrow$ er is een waardering die $T(\phi)$ waarmaakt.

- Conclusie uit de vorige punten: SAT $\leq_P$ 3SAT.

We hebben nu: SAT $\leq_{\mathsf{P}}$ 3SAT (*)
Verder is 1SAT: Gegeven een logische formule ϕ in 1-CNF. Bestaat er een waardering die ϕ True maakt? Een logische formule ϕ in 1-CNF heeft de volgende vorm: $\phi = \ell_1 \wedge \ell_2 \wedge \ldots \wedge \ell_n$.

1. Stel dat we weten dat 3SAT $\in \mathcal{NPC}$ en SAT $\in \mathcal{NP}$. Volgt dan uit (*) dat SAT $\in \mathcal{NPC}$?

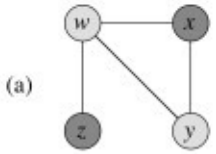
2. Stel dat we weten dat SAT $\in \mathcal{NPC}$ en 3SAT $\in \mathcal{NP}$. Volgt dan uit (*) dat 3SAT $\in \mathcal{NPC}$?

3. Stel dat we weten dat 3SAT $\in \mathcal{NPC}$. Is 1SAT $\leq_{\mathsf{P}}$ 3SAT?

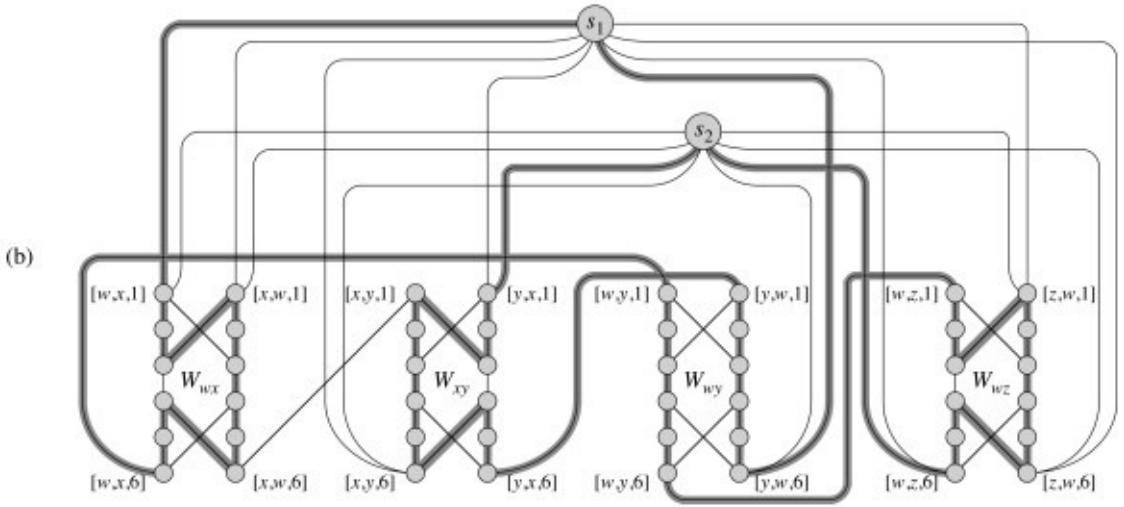
4. Stel dat we weten dat 3SAT $\in \mathcal{NPC}$. Is 3SAT $\leq_{\mathsf{P}}$ 1SAT?

Lastig: VC $\leq_P$ HC2

Het probleem HC2 (heeft een gegeven ongerichte graaf een Hamiltonkring) zit in $\mathcal{NP}$; een lastige reductie van VC (heeft een gegeven ongerichte graaf een vertex cover ter grootte k: een stel knopen dat elke tak raakt) naar HC2:



uit Cormen/Leiserson/Rivest/Stein,
Introduction to Algorithms



- Volgende college:
  dinsdag 7 mei, 11.00 – 12.45, zaal 174
  Let op: geen college op dinsdag 30 april, wel werkcollege

- Eerstvolgende werkcolleges:
  dinsdag 23 april, 13.30 – 15.15, zaal 174
  Opgaven 46, 47, 48, 56, 57
  dinsdag 30 april, 13.30 – 15.15, zaal 174
  Opgaven 53, 54, 59

- Vierde en laatste huiswerkopgave:

  * deadline: dinsdag 14 mei; LaTeX; print → college

  * www.liacs.leidenuniv.nl/~graafjmde/COMP/