

Tentamen IN3105

Complexiteitstheorie

30 maart 2009, 9.00 – 12.00 uur

- Dit tentamen bestaat uit 10 meerkeuzevragen, 5 korte open vragen en 2 open vragen.
- Per meerkeuzevraag kunnen 0 tot 4 alternatieven juist zijn.
- Voor de meerkeuzevragen kunt u maximaal 50 punten behalen, voor de korte open vragen maximaal 30 punten en voor de open vragen maximaal 20 punten.
- Het totaal aantal punten mp voor de meerkeuzevragen wordt als volgt bepaald:
 - o per meerkeuzevraag i wordt het aantal foutief beantwoorde alternatieven f_i bepaald (0-4);
 - o de som f van dit aantal voor alle 10 meerkeuzevragen wordt bepaald ($0 \leq f \leq 40$)
 - o het aantal punten mp is dan gelijk aan $mp = 50 \times \max\{0, (40 - 1.5 \times f) / 40\}$
- Het eindcijfer c wordt bepaald door de som te nemen van mp , het totaal aantal punten behaald voor de 5 korte vragen en de twee open vragen, en vervolgens het resultaat te delen door 10 en af te ronden naar een geheeltallig cijfer.
- Het gebruik van dictaat, aantekeningen, boek of andere bronnen is niet toegestaan.
- Het gebruik van grafische en niet-grafische rekenmachines is eveneens niet toegestaan.
- Beantwoord de meerkeuzevragen op het bijgeleverde antwoordformulier.
- Beantwoord de open vragen zo bondig mogelijk; *geef geen irrelevante informatie, dit kan leiden tot puntenaftrek!*
- Controleer of u op ieder blaadje uw naam en studienummer heeft vermeld en geef het totaal aantal ingeleverde bladen op (tenminste) de eerste pagina.

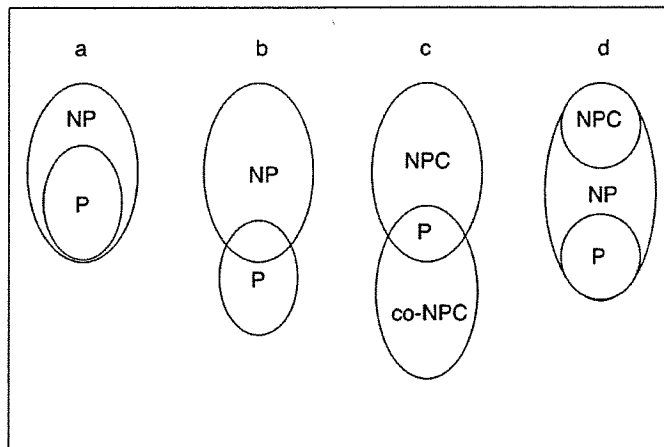
Notationele conventies:

- $A \leq B$ betekent dat er een polynomiale-tijd reductie van probleem A naar probleem B bestaat.
- A^c is het complement van het probleem A .
- Er wordt steeds vanuit gegaan, dat $P \neq NP$ en $NP \neq co-NP$, tenzij uitdrukkelijk het tegendeel wordt vermeld.

MEERKEUZEVRAGEN

Vraag 1. In onderstaande situaties (weergegeven door a, b, c en d) zijn een aantal relaties tussen complexiteitsklassen weergegeven. Ga na welke van onderstaande situaties zich *nooit* kan voordoen, onafhankelijk van het feit of nu $P = NP$, dan wel $P \neq NP$.

- a. situatie a.
- ☒ b. situatie b.
- ☒ c. situatie c.
- d. situatie d.



Vraag 2. Gegeven zijn de beslissingsproblemen A , B en C met $A \leq C$ en $B \leq C$. Het is bekend dat B een NP-hard probleem is. Onder welke condities is het zeker dat B een NPC-probleem is?

- a. als C een NP-probleem is.
- b. als A een NPC-probleem is.
- c. als $A \leq B$ en A is een NPC-probleem.
- d. als A een NP-probleem is.

Vraag 3. Als A een probleem in $NP \cup \text{co-NP}$ is, dan geldt dat

- a. A geen polynomiaal algoritme heeft.
- b. yes-instanties of no-instanties van A polynomiaal verifieerbaar zijn, maar niet beide.
- c. A geen P-probleem is, tenzij $NP = \text{co-NP}$.
- d. voor elk (co)-NP probleem B geldt $B \leq A$.

Vraag 4. Ga na welke van de volgende uitspraken juist zijn:

- a. Als A een P-probleem is dan is A^c een NP-probleem.
- b. $NP^P \subseteq NP$.
- c. Als $A \in \text{co-NP}$ dan $A \in PSPACE$.
- d. $NP \cup \text{co-NP} \subseteq P^{NP}$.

Vraag 5. Het ROUTEKEUZEPROBLEEM MET TUSSENSTOPS is het volgende probleem:
Gegeven een graaf $G = (V, E)$, een subset $V' \subseteq V$, twee knopen s en t in V , en een pos. integer K , bestaat er een s - t pad in G dat alle knopen in V' aandoet en een lengte heeft hoogstens gelijk aan K ? Dit probleem is:

- a. een NPC-hard probleem, want HAMPAD is polynomiaal reduceerbaar naar dit probleem.
- b. een P-probleem, want het is immers een restrictie van het KORTSTE PAD probleem.
- c. een co-NP-probleem, want nee-instanties zijn polynomiaal verifieerbaar.
- d. een NP-probleem, want yes-instanties zijn polynomiaal verifieerbaar.

Vraag 6. We noemen een optimaliseringsprobleem A *benaderbaar* als A een approximatie-algoritme heeft met approximatieverhouding $1 \leq 1 + \varepsilon < \infty$. Dan geldt:

- MINCOVER is een benaderbaar probleem.
- TSP-OPT is niet benaderbaar, tenzij $P=NP$.
- Een probleem is benaderbaar als het een element is van de klasse **APX**.
- Ieder probleem in **NPO** is benaderbaar.

Vraag 7. Neem het volgende probleem

$SPECIALCLIQUE = \{ \langle G, v, k \rangle \mid G \text{ is een ongerichte graaf met een clique ter grootte van } k \text{ die de knoop } v \text{ bevat} \}$

Dit probleem is een **NP**-hard probleem, omdat de volgende transformatie f een polynomiale tijdreductie is van het **NP**-complete **CLIQUE** probleem naar het **SPECIALCLIQUE** probleem: voor een instantie $I = (G = (V, E), k)$ van het ~~VERTEX COVER~~ **CLIQUE** probleem, laat

- $f(I) = (G = (V \cup \{v\}, E^c), v, k+1)$.
- $f(I) = (G = (V \cup \{v\}, E), v, k)$.
- $f(I) = (G = (V \cup \{v\}, E), v, k+1)$.
- $f(I) = (G = (V \cup \{v\}, E \cup \{\{w, v\} : w \in V\}), v, k+1)$.

Nb: In alle gevallen geldt dat $v \notin V$.

Vraag 8. Het **BINPACKING** probleem bestaat uit een verzameling van N items $1, 2, \dots, N$, waarbij ieder item j een gewicht $0 < s_j \leq 1$ heeft. Verder is een verzameling vaten (bins) beschikbaar, ieder met capaciteit 1 . Gevraagd wordt zowenig mogelijk vaten te gebruiken om alle items op te bergen zonder dat de capaciteit van de vaten overschreden wordt, dwz. de som van de gewichten s_j van de items j die opgeborgen wordt in enig vat v mag niet meer dan 1 bedragen.

Op het college is hiervoor het volgende (**NEXTFIT**) benaderingsalgoritme besproken:

Plaats de items in een willekeurige volgorde. Neem een vat.

Voor item $j = 1$ tot N doe het volgende:

Als item j in het meest recente vat past, stop het in het vat, anders pak een nieuw vat (dit wordt het meest recente vat) en stop het item j in dit vat.

Dit algoritme laat zien dat er voor het **BINPACKING** probleem een benaderingsalgoritme bestaat met een approximatieverhouding

- $1 + \varepsilon = 2$
- $1 + \varepsilon = 1.5$
- $1 + \varepsilon = O(\log N)$
- $1 + \varepsilon = 1$

Vraag 9. Neem het probleem om, gegeven een ongerichte graaf $G = (V, E)$ en twee positieve integers K en L , te beslissen of er een minstens één simpel pad ter lengte van K in G bestaat en geen enkel simpel pad in G te vinden is met lengte $\geq L$. Er geldt:

- als dit probleem een **P**-probleem zou zijn, dan zou gelden dat $P=NP$.
- dit probleem is een **NP**-probleem.
- dit probleem is een **co-NP** probleem.
- HAMPAD** is polynomiaal reduceerbaar naar dit probleem.

Vraag 10. Het door Sipser besproken *amplification lemma* komt er op neer dat $BPP = BPP_\delta$ voor elke δ met $0 < \delta < 0.5$. Hierbij is BPP_δ de klasse van alle talen L waarvoor er een polynomiale probabilistische Turing machine M bestaat met de eigenschap dat

- als $w \in L$ dan geldt $\Pr(M \text{ accepteert } w) \geq 0.5 + \delta$
- als $w \notin L$ dan geldt $\Pr(M \text{ verwierpt } w) \geq 0.5 + \delta$.

Er geldt nu het volgende:

- Voor $\delta = 0.5$ is BPP_δ gelijk aan **P**.
- Voor $\delta = 0.1$ is BPP_δ gelijk aan **PP**.
- $P \subseteq BPP$.
- $BPP \subset P$.

OPEN VRAGEN I

Onderstaande vragen dient u kort te beantwoorden.
Een antwoord dient in hoogstens 5-7 regels gegeven te worden.

Vraag 11. Gegeven is een taal $L \in \mathbf{NP}$. Laat zien dat, als voor iedere taal $L' \in \mathbf{co-NP}$ geldt $L' \leq L$, er tevens geldt dat $\mathbf{NP} = \mathbf{co-NP}$.

Vraag 12. Iemand beweert dat $3\text{SAT} \leq 2\text{SAT}$. Hij geeft hiervoor het volgende bewijs:

Gegeven een 3-SAT formule ϕ (een conjunctie van clauses), vind een waarheidstoekenning τ die ϕ vervult. Als zo'n waarheidstoekenning niet bestaat, retourneer dan een onvervulbare 2-SAT instantie γ , anders doe voor iedere clause C_i in ϕ het volgende:

- als alle literals in C_i door τ worden waargemaakt, verwijder dan een arbitraire literal uit C_i .
- als minstens een literal uit C_i door τ onwaar wordt gemaakt, kies dan een zo'n literal en verwijder deze uit C_i .

Laat γ de conjunctie zijn van de aldus verkregen clauses.

Allereerst geldt natuurlijk dat γ een conjunctie is van 2 literal clauses. Dus $\gamma \in 2\text{SAT}$.

De reductie kost polynomiale tijd want iedere clause C_i uit ϕ wordt maar eenmaal bekeken en vergt per clause $O(1)$ -tijd. De reductie is correct, omdat

- (1) als $\phi \in 3\text{SAT}$, dan zal dezelfde waarheidstoekenning die de 3SAT instantie waarmaakt ook de resulterende 2-SAT formule γ waarmaken.
- (2) als $\phi \notin 3\text{SAT}$ dan geldt γ is onvervulbaar.

Geef exact aan welke fout(en) in dit bewijs gemaakt word(t)(en).

Vraag 13. Toon aan dat $\mathbf{NP}$ naar beneden gesloten is onder polynomiale tijd reducties.

Vraag 14. Waarom geldt $\mathbf{NP}^{\mathbf{co-NP}} = \mathbf{NP}^{\mathbf{NP}}$?

Vraag 15. Welke evidentie bestaat er voor het vermoeden dat quantum computing, ten opzichte van klassieke berekeningen voor de oplossing van $\mathbf{NP}$ -problemen, voor sommige problemen in $\mathbf{NP}$ een exponentiele speed-up zou kunnen geven?

OPEN VRAGEN II

Beantwoord onderstaande vragen zo kort mogelijk. Geef geen irrelevante informatie, dit kan tot puntenaftrek aanleiding geven.

Vraag 16. (10 punten)

Neem het volgende PIZZA-probleem:

Er moeten N pizza's $p_1, p_2, \dots, p_N$ gebakken worden. Er zijn $M \geq 1$ ovens beschikbaar. In iedere oven kan slechts één pizza tegelijk worden gebakken. De baktijd van een pizza p_i is mede-afhankelijk van de oven waarin de pizza gebakken wordt: als pizza p_i in oven j gebakken wordt, dan is de baktijd $t_{ij} \in \mathbb{Z}^+$. Bovendien is een beschikbare baktijd $B \in \mathbb{Z}^+$ gegeven.

Gevraagd wordt of er een bakschema bestaat zodat de benodigde baktijd voor het bakken van alle pizza's niet meer dan B bedraagt.

- Toon aan, dat het PIZZA probleem voor $M=1$ een polynomiaal probleem is (≤ 5 regels).
- Laat zien dat voor $M \geq 2$ dit probleem een NP-hard probleem is (≤ 10 regels).

Vraag 17. (10 punten)

Student O. Pletter beweert een polynomiaal algoritme gevonden te hebben voor het VERTEX COVER probleem. Hij beweert het volgende:

- Een instantie van het VERTEX COVER probleem bestaat uit een ongerichte graaf $G=(V,E)$ en een positieve integer K .
- Om de vraag te beantwoorden of G een vertex cover ter grootte van K bevat, kunnen we het volgende algoritme toepassen:
 - Selecteer alle deelverzamelingen V' van knopen uit G die precies K knopen bevatten;
 - Ga voor iedere dergelijke deelverzameling V' na of voor iedere kant $e = \{v, w\}$ uit E geldt: $v \in V'$ en/of $w \in V'$;
 - Als er zo'n deelverzameling V' bestaat die aan de bovengenoemde voorwaarde voldoet, dan is G een yes-instantie, anders is G een no-instantie.
- De looptijd van dit algoritme is polynomiaal: immers, er zijn hoogstens $O(|V|!/(K!(|V|-K)!)) = O(|V|^K)$ deelverzamelingen V' van V ter grootte van K ;
- Per deelverzameling V' kost het niet meer dan $O(K \times |E|)$ -tijd om na te gaan of V' een vertex cover is of niet;
- Het totale algoritme kost derhalve $O(|V|^K \times K \times |E|)$ -tijd en dit is voor vaste K een algoritme dat polynomiaal is in de grootte $|I| = |V| + |E| + \log K$ van de probleem instantie. Derhalve bestaat er een polynomiaal algoritme voor dit probleem.

Geef precies aan welke van de genummerde beweringen *incorrect* zijn en motiveer uw antwoord kort en bondig (niet meer dan 10 regels).

EINDE TENTAMEN