

Tentamen IN3105

Complexiteitstheorie

21 juni 2010, 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 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 op halve punten.
- 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 A naar B bestaat.
- A^c is het complement van het probleem A .
- Er wordt steeds vanuit gegaan, dat $P \neq NP$ en $NP \neq coNP$, tenzij uitdrukkelijk het tegendeel wordt vermeld.

VEEL SUCCES!

MEERKEUZEVRAGEN (50 PUNTEN)

Vraag 1. Het is bekend dat het Vertex Cover (VC) probleem een **NP**-compleet probleem is. Dan geldt dat:

- a. VC polynomiaal reduceerbaar is naar ieder probleem in **NP**.
- b. geen enkele probleeminstantie van VC in polynomiale tijd oplosbaar is.
- c. alle probleeminstanties van VC in exponentiële tijd oplosbaar zijn.
- d. sommige probleeminstanties van VC niet in polynomiale tijd oplosbaar zijn, tenzij **P = NP**.

Antwoord

Alternatief a. is onjuist (het is net andersom), b. is onjuist, c. en d. zijn juist

Vraag 2. Ga na welke van de volgende uitspraken juist is/zijn:

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

Antwoord

a., b. en d. zijn juist.

Vraag 3. Welke van onderstaande beweringen zijn correct als **NP = P**?

- a. Ieder probleem $A \in \mathbf{P}$ ligt ook in **NPC**.
- b. Alle **NP**-harde problemen zijn tevens **NPC** problemen.
- c. Voor ieder probleem $A \in \mathbf{NP}$ bestaat er een probleem $B \in \mathbf{NP}$ met $A \leq B$.
- d. alle **P**-problemen zijn onderling polynomiaal reduceerbaar tot elkaar.

Antwoord

alleen c. is juist (neem voor B een **NP**-compleet probleem).

Vraag 4. Het probleem om, gegeven een ongerichte graaf $G = (V, E)$, te bepalen of

- a. G geen enkel hamiltons circuit bevat, is een **NP**-volledig probleem.
- b. G hoogstens één hamiltons circuit bevat is een **co-NP**-volledig probleem.
- c. G exact één hamiltons circuit bevat is een probleem in **NP** $\cap$ **co-NP**.
- d. tenminste twee hamiltonse circuits bevat is een **NP**-volledig probleem.

Antwoord

b. en d. zijn juist

Vraag 5. Voor twee problemen X en Y geldt dat er een **NPC**-probleem A bestaat zodanig dat $X \leq A$ en $A \leq Y$. Welke van de volgende uitspraken is (zijn) waar?

- a. Y is een **NP**-probleem.
- b. X is een **NP**-probleem.
- c. Als Y een **P**-probleem is, dan is **P = NP**.
- d. Y is een **NPC**-probleem.

Antwoord

b. en c. zijn juist

Vraag 6. Welke van de onderstaande inclusierelaties zijn geldig:

- a. **P** $\subseteq$ **NP** $\subseteq$ **PSPACE**.
- b. **P^{NP}** $\subseteq$ **NP**.
- c. **P^{PSPACE}** $\subseteq$ **PSPACE**.
- d. **P** $\subseteq$ **PP**.

Antwoord

a., c. en d. zijn juist

Vraag 7. Gegeven is het volgende probleem

Instantie: Graaf $G = (V, E)$, een positieve integer K .

Vraag: Heeft G een hamiltons pad *en* heeft G een vertex-cover ter grootte van K ?

- Dit probleem is een **NPC**-probleem.
- Dit probleem is een **coNPC**-probleem.
- Dit probleem is een **NP^{NP}**-probleem en geen **NP**-probleem.
- Dit probleem is een **P**-probleem.

Antwoord

Alleen a. is juist

Vraag 8 Het FINDKNAPSACK probleem is het probleem om, gegeven een verzameling items $A = \{a_1, a_2, \dots, a_n\}$, een gewicht functie $g: A \rightarrow \mathbb{Z}^+$ en een integer t , een deelverzameling A' van A te vinden (als zo'n deelverzameling bestaat), zodanig dat de som van de gewichten $g(a)$ van de items a in A' gelijk is aan t . Als zo'n deelverzameling niet bestaat moet **nil** als antwoord worden gegeven. De beslissingsvariant van dit probleem is het KNAPSACK probleem. Iemand presenteert het volgende algoritme voor FINDKNAPSACK. Dit algoritme gebruikt een algoritme voor het KNAPSACK probleem als orakel:

input KNAPSACK-instantie $(A = \{a_1, a_2, \dots, a_n\}, g, t)$

begin

$n := |A|$; $i := 1$; $Rest := A$; $A' := \emptyset$;

for $i = 1$ to n

$Rest := Rest - \{a_i\}$;

if KNAPSACK($Rest, g, t - g(a_i)$) = "yes"

then

$A' := A' \cup \{a_i\}$;

$t := t - g(a_i)$;

if $t = 0$

then

return A'

else

return nil

end

Over dit algoritme kan het volgende opgemerkt worden:

- de conditie if KNAPSACK($Rest, g, t - g(a_i)$) = "yes" is onjuist en moet vervangen worden door if KNAPSACK($Rest, g, t - g(a_i)$) = "no" .
- dit algoritme termineert niet voor no-instanties van KNAPSACK.
- dit algoritme levert een onjuiste uitkomst als $t > \sum_{a \in A} g(a)$.
- dit algoritme toont aan dat het FINDKNAPSACK probleem een **P^{NP}** probleem is.

Antwoord

alternatief d. is juist

Vraag 9. We noemen een optimaliseringsprobleem A *onbenaderbaar* als A geen approximatie-algoritme heeft met approximatieratio $c < \infty$. Dan geldt:

- MINCOVER is een onbenaderbaar probleem.
- TSP-OPT is onbenaderbaar.
- GEWOGEN KNAPSACK is onbenaderbaar.
- Alle onbenaderbare **NPO**-problemen liggen in de klasse **NPO-APX**.

Antwoord

b. en d. zijn juist

Vraag 10. Het *amplification lemma* in Sipser stelt dat $\mathbf{BPP} = \mathbf{BPP}_\delta$ voor elke $0 < \delta < 0.5$.
Hierbij is $\mathbf{BPP}_\delta$ de klasse van alle talen L waarvoor er een polynomiale probabilistische Turing machine M bestaat zodanig dat

- (i) als $w \in L$ dan geldt $\Pr(M \text{ accepteert } w) \geq 1/2 + \delta$
- (ii) als $w \notin L$ dan geldt $\Pr(M \text{ verwierpt } w) \geq 1/2 + \delta$.

Er geldt nu het volgende:

- a. Voor $\delta = 0.5$ is $\mathbf{BPP}_\delta$ gelijk aan $\mathbf{P}$.
- b. Voor $\delta = 0.1$ is $\mathbf{BPP}_\delta$ gelijk aan $\mathbf{BPP}$.
- c. $\mathbf{P} \subseteq \mathbf{BPP}$.
- d. $\mathbf{NP} \subseteq \mathbf{BPP} \subseteq \mathbf{PP}$.

Antwoord

a., b. en c. zijn juist

OPEN VRAGEN I (30 PUNTEN)

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

Vraag 11. Waarom geldt $P^{co-NP} = P^{NP}$?

Antwoord

$P^{co-NP} \subseteq P^{NP}$: neem een NP-orakel A , en laat A^c het complementaire co-NP orakel zijn. Beschouw een algoritme X dat A aanroept middels een call A_X . Door alle calls A_X in het algoritme X te vervangen door $\neg A_X^c$, is een algoritme in P^{co-NP} verkregen dat equivalent is aan X . Hierbij is A_X^c een call vanuit X naar A^c . Idem voor $P^{NP} \subseteq P^{co-NP}$. Resultaat is $P^{co-NP} = P^{NP}$.

Vraag 12. Leg uit waarom het probleem:

Gegeven een SAT-instantie (U, C) , bestaat er een waarheidstoekenning die C onwaar maakt?
een **P**-probleem is.

Antwoord

Als C een clause bevat die niet zowel een literal u als zijn complement $\neg u$ bevat, dan is er altijd een waarheidstoekenning die deze clause onwaar maakt en daarmee C onwaar maakt. Dit kan in lineaire tijd worden beslist door na te gaan of geen van de clauses een literal en zijn complement bevat.

Vraag 13. Beschouw de volgende uitspraak:

Als L en M twee gegeven talen zijn, dan volgt uit $L \subseteq M$ dat $L \leq M$.

Geef een voorbeeld waarin deze uitspraak opgaat en een tegenvoorbeeld voor deze uitspraak.

Antwoord

Neem de lege taal $\emptyset$ en de volledige taal Σ^* . Er geldt $\emptyset \subseteq \Sigma^*$, maar niet $\emptyset \leq \Sigma^*$, want nee-instanties kunnen niet worden afgebeeld. Neem echter een NP-complete taal A over Σ . Dan geldt $\emptyset \subseteq A$ en ook $\emptyset \leq A$ per definitie van NP-compleetheid.

Vraag 14. Waarom is het probleem:

Gegeven een graaf $G = (V, E)$ met $|V| \leq 1000$, heeft G een vertexcover ter grootte van $|V|/2$?
een **P**-probleem is.

Antwoord

Omdat $|V|$ begrensd is, zijn er hoogstens $(1000 \text{ over } 500) = \text{constant}$ aantal subsets van knopen die gecheckt moeten worden op de vertex cover eigenschap. Iedere check kan in constante tijd worden uitgevoerd omdat zo'n deelverzameling een begrensd aantal knopen heeft. Derhalve kan een vertex cover ter grootte van minimaal 500 in constante tijd gevonden worden.

Vraag 15. Welke evidentie bestaat er voor het vermoeden dat quantum computing, ten opzichte van klassieke berekeningen voor de oplossing van **NP**-problemen, voor sommige problemen in die klasse een exponentiële speed-up zou kunnen geven?

Antwoord

Het factorizeringsprobleem is een probleem in NP waarvoor geen polynomiale algoritmen bekend zijn, echter wel polynomiale kwantum algoritmen bekend zijn.

OPEN VRAGEN II (20 PUNTEN)

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

Vraag 16. (10 punten)

We nemen het volgende LANDINGS-probleem:

Er moeten N vliegtuigen $v_1, v_2, \dots, v_N$ landen. Hiervoor zijn M landingsbanen beschikbaar. Op iedere baan kan slechts één vliegtuig tegelijk landen. De tijd benodigd voor het landen van v_i is afhankelijk van de baan die gebruikt wordt: als v_i op baan j landt, dan is de benodigde tijd gelijk aan $t_{ij} \in \mathbb{Z}^+$. Er is bovendien een maximaal beschikbare tijd $B \in \mathbb{Z}^+$ waarop de banen operationeel zijn.

Gevraagd wordt of er een landingsschema (een toewijzing van vliegtuigen aan landingsbanen) bestaat zodat de benodigde (doorloop)tijd voor het laten landen van alle vliegtuigen niet meer dan B bedraagt.

Voorbeeld: stel $B = 6$ en er zijn 2 banen en 3 vliegtuigen met de volgende landingstijden:

	<i>baan 1</i>	<i>baan 2</i>
v_1	2	5
v_2	6	3
v_3	4	7

Een oplossing is dat v_1 en v_3 gebruikmaken van baan 1 (totale tijd voor baan 1: $2+4 = 6$) en v_2 van baan 2 (totale tijd voor baan 2: 3). De totaal benodigde tijd is dan $6 = \max\{6, 3\}$.

- a. Laat zien dat voor $M=2$ dit probleem een **NP**-hard probleem is (≤ 10 regels).

Hint: pas restrictie van het probleem toe.]

Antwoord

Laat $t_{i1} = t_{i2}$ voor $j = 1, 2$ en $B = \lfloor \sum t_{ij} / 2 \rfloor$. Dan is een LANDINGS instantie identiek aan een PARTITION instantie. Neem immers $A = \{v_1, v_2, \dots, v_N\}$ en $g(a_i) = g(v_i) = t_{i1}$. Hiermee is aangetoond dat PARTITION een restrictie is van een NP-compleet probleem, dus LANDING is NP-hard

- b. Hoe kun je eenvoudig laten zien dat het probleem voor elke waarde van $M \geq 2$ een **NP**-hard probleem is? (5 regels)

Antwoord

We weten dat het landingsprobleem voor 2 landingsbanen NP-hard is. Neem nu een willekeurige 2 landingsbaan instantie en voeg $M-2$ nieuwe vliegtuigen toe die voor iedere landingsbaan B landingstijd nodig hebben. Dan is de gecreeerde instantie een yes-instantie alss de oorspronkelijke 2 landingsbaan instantie een yes-instantie is. Daarmee is de NP-hardheid voor iedere waarde van $M \geq 2$ bewezen.

Vraag 17. (10 punten)

Laat FAC het volgende beslissingsprobleem zijn: Gegeven integers n, m , heeft n een priemfactor die kleiner is dan m ?

- a. Laat zien dat FAC een probleem is in **NP**.
b. Laat zien dat FAC een probleem is in **coNP**.

Hint: Maak gebruik van het gegeven dat er een efficient algoritme bestaat om te beslissen of een integer priem is of niet (Dit algoritme is niet triviaal en is pas in 2003 ontdekt.).

Antwoord

Ad a. Gok twee gehele getallen p en q , verifieer dat $p \cdot q = n$ en verifieer dat $\min\{p, q\} < m$. Als dit geverifieerd is, weten we zeker, en in polynomiale tijd, dat n een priemfactor heeft $< m$. Derhalve is FAC een **NP**-probleem.

Ad b. Gok k gehele getallen $p_1, p_2, \dots, p_k$ en verifieer dat

- 1) elk getal $p_i \geq m$ is.
- 2) elk getal p_i een priemgetal is.
- 3) het product $p_1 \cdot p_2 \cdot \dots \cdot p_k = n$.

Nu geldt dankzij PRIEM in P, en het feit dat ieder getal een unieke ontbinding heeft in priemgetallen, dat deze verificatie in polynomiale tijd kan worden uitgevoerd.

Als deze verificatie slaagt dan weten we ook dat n geen priemfactor heeft $< m$. Dus FAC in **coNP**.

EINDE TENTAMEN