Uitwerkingen Tentamen IN3105

Complexiteitstheorie

13 april 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:
  - per meerkeuzevraag *i* wordt het aantal foutief beantwoorde alternatieven f_i bepaald (*0-4*);
  - de som *f* van dit aantal voor alle 10 meerkeuzevragen wordt bepaald $(0 \leq f \leq 40)$
  - 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** ≠ **NP** en **NP** ≠ **coNP**, tenzij uitdrukkelijk het tegendeel wordt vermeld.

VEEL SUCCES!

MEERKEUZEVRAGEN (50 PUNTEN)

Vraag 1. Als **P** = **coNP** dan geldt**:**

a. **P = NP**.
b. **NPC** = **P**.
c. **NPC = coNPC**.
d. **NP = NPC**.

Uitwerking: Als P=coNP dan is P=NP, omdat P gesloten is onder complement en het complement van co-NP gelijk is aan NP. Omdat de lege taal $\varnothing$ en de volledige taal Σ^* nooit NP-volledig kunnen zijn, geldt NPC $\neq$ P, ook als P=NP. Omdat P= NP = coNP, geldt NPC =coNPC. Omdat P =NP en P $\neq$ NPC, geldt nu ook NP $\neq$ NPC.

Vraag 2. Gegeven zijn de beslissingsproblemen A, B en C met $A \leq B$ en $B \leq C$. Het is bekend dat B een **NPC** probleem is. Dan geldt in ieder geval, dat

a. A een **NP**-probleem is.
b. C een **NP**-hard probleem is.
c. als $B \leq A$, dan is A ook een **NPC**-probleem.
d. als C een **NP** probleem is, dan is A een **P**-probleem.

Uitwerking: Omdat NP naar beneden gesloten is onder $\leq$, geldt A in NP. C is een NP-hard probleem omdat B een NPC probleem is en B $\leq$ C. Als B $\leq$ A zijn A en B even moeilijk en omdat A een NP probleem is, is A nu een NPC probleem. Als C een NP probleem is, is C een NPC probleem en A een NP probleem. Hieruit is niet af te leiden dat A een P probleem is.

Vraag 3. Welke van onderstaande beweringen zijn correct onder de aanname dat **NP** $\neq$ **P**?

a. Een probleem kan niet zowel in **P** als in **NPC** liggen.
b. Alle **NP**-harde problemen zijn tevens **NPC** problemen.
c. Voor ieder tweetal problemen A en B in **NP** geldt: $A \leq B$ of $B \leq A$.
d. Voor ieder **P**-probleem A bestaat er een **NP** probleem B met $A \leq B$.

Uitwerking: Als P $\neq$ NP, dan is NPC disjunct met P, omdat NPC $\cap$ P $\neq \varnothing$ impliceert dat P=NP. Een NP-hard probleem A is alleen NPC probleem, als A een NP probleem is, dit verandert niet als NP$\neq$ P, dus b. is niet waar. Als we voor A de lege taal en B de volledige taal nemen, geldt c. niet, dus c. is onwaar. Alternatief d. is waar, omdat als keuze voor B een NPC probleem volstaat.

Vraag 4. Een optimaliseringsprobleem A heet *benaderbaar* als A een approximatie-algoritme heeft met approximatieratio r $< \infty$. Dan geldt:

a. **NPO** is de klasse van benaderbare optimaliseringsproblemen.
b. Alle **APX**-problemen zijn benaderbaar.
c. Alle **FPTAS**-problemen zijn benaderbaar.
d. Als een **NPO**-probleem niet benaderbaar is, dan ligt het in de klasse **NPO - APX**.

Uitwerking: Alle APX en FPTAS problemen zijn per definitie benaderbaar: b. en c. zijn juist. Als een NPO probleem niet in APX ligt, is het niet benaderbaar, derhalve is a. onjuist en d. juist.

Vraag 5. Welke van de volgende uitspraken zijn juist:

a. Een probleem is **NP**-compleet als er voor iedere instantie van het probleem een exponentieel aantal kandidaatoplossingen aanwezig is.
b. Als **PSPACE** = **P** dan is **P = NP = coNP**.
c. Als **A** $\in$ **PSPACE** dan geldt **A** $\in$ **NP**-hard.
d. **NC** $\subseteq$ **NP**.

Uitwerking: a. is niet juist: zelfs P-problemen als het kortste pad probleem, voldoen aan het gestelde criterium. Als PSPACE = P, geldt P= NP, omdat NP bevat is in PSPACE. Daarmee geldt ook P=NP=coNP: Als A een PSPACE probleem is, kan A ook een P-probleem zijn, het is derhalve onjuist om te stellen dat A een NP-hard probleem is. NC is de klasse van efficient parallelliseerbare problemen en een subklasse van P: alternatief d is juist.

Vraag 6. Welke van de onderstaande inclusierelaties zijn geldig:

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

Uitwerking: a. is juist; omdat $NP^P = NP \subseteq P^{NP}$, is alternatief b. ook juist. Alternatief c. is juist omdat NP$\subseteq$PP. Alternatief d. is ook juist omdat P de klasse van talen is die geaccepteerd worden door een probabilistische Tm met foutmarge 0.

Vraag 7. Gegeven zijn n processoren $P_1,...,P_n$ en een verzameling taken T. Van iedere taak t in T is de verwerkingstijd $v(t)$ in seconden bekend. Verder is er een deadline d (in gehele seconden) gegeven. Gevraagd wordt of het mogelijk is de taken uit T zodanig over de n processoren te verdelen, dat elke processor P_i niet meer dan d seconden nodig zal hebben om alle aan P_i toebedeelde taken achter elkaar uit te voeren.
Dit **NP**-probleem is:

a. een **NP**-volledig probleem, immers het is een restrictie van het PARTITION probleem.
b. een **NP**-volledig probleem, PARTITION, is een restrictie van dit probleem.
c. een **P**-probleem.
d. voor $n = 1$ een **P**-probleem en voor $n = 2$ een **NP**-volledig probleem.

Uitwerking: Het PARTITION probleem is een restrictie van dit probleem: Kies $n=2$ en $d = \lfloor$(som van de verwerkingstijden van de taken) /2 $\rfloor$. Omdat het probleem tevens polynomiaal verifieerbaar is, is dit probleem een NPC probleem. Voor n=1 is het polynomiaal oplosbaar: tel alle verwerkingstijden bij elkaar op en ga na of deze som $\leq d$ is. Hieruit volgt dat a. onwaar is, b. waar is, c. onwaar is en d. waar is.

Vraag 8. Het ZERO-WEIGHT-SIMPLE-CYCLE is het volgende probleem:

Gegeven: een graaf $G = (V,E)$ en een functie w die aan iedere kant e in E een positief of negatief geheeltallig gewicht $w(e)$ toekent.
Gevraagd: heeft G een simpele cycle waarvan de som van de gewichten op de kanten in de cycle gelijk is aan 0.

Om aan te tonen dat dit probleem **NP**-hard is, stelt iemand de volgende reductie voor:
Gegeven een instantie $G = (V, E)$ van HAMILTOONS CIRCUIT met v een willekeurige knoop in V en $|V| = n$, laat $f(G) = (V, E, w)$ de instantie zijn van ZERO-WEIGHT-SIMPLE-CYCLE waarbij $w(e) = n$-1 voor alle kanten e die knoop v als één van de eindpunten hebben en $w(e) = -1$ voor alle overige kanten e in E.

Er geldt nu:

a. Dit is een correcte polynomiale reductie van HAMILTOONS CIRCUIT naar ZERO-WEIGHT-SIMPLE-CYCLE.
b. Dit is een correcte reductie van HAMILTOONS CIRCUIT naar ZERO-WEIGHT-SIMPLE-CYCLE, maar helaas niet polynomiaal.
c. Dit is geen correcte polynomiale reductie van HAMILTOONS CIRCUIT naar ZERO-WEIGHT-SIMPLE-CYCLE.
d. Dit is zou een correcte polynomiale reductie van HAMILTOONS CIRCUIT naar ZERO-WEIGHT-SIMPLE-CYCLE zijn, als $w(e) = n$-2 voor alle kanten $e \in E$ die v als een eindpunt hebben.

Uitwerking: Deze reductie is polynomiaal maar niet correct. Als G een hamiltoons circuit (en dus $n \geq 3$) heeft, zijn er $n-2$ kanten met gewicht 1 in zo'n cycle en twee kanten waarin v voorkomt met gewicht $n-1$. Zo'n hamcycle kan derhalve nooit als cycle met gewicht 0 gekozen worden. Als $f(G)$ een simpele cycle heeft die v niet bevat, is het totaal gewicht <0. Als $f(G)$ een simpele cycle heeft die v wel bevat is het totale gewicht van de cycle $\geq (n-2)x-1 + 2(n-1) = n > 0$. Het resultaat is, dat f yes-instanties van HAMILTOONS CIRCUIT afbeeldt op no-instanties van ZERO-WEIGHT-SIMPLE-CYCLE. Alternatief a. en b. zijn derhalve onjuist. Alternatief d. is eveneens niet correct: een cycle in $f(G)$ die v *niet* bevat heeft een gewicht < 0 en anders een gewicht $\geq (n-2)x-1 + 2(n-2) = n - 2 > 0$. Het juiste alternatief is daarom c.

Vraag 9. Neem het volgende probleem

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

Dit probleem is een **NP**-hard probleem, omdat de volgende transformatie f een polynomiale tijdreductie is van het **NP**-complete VC naar SPECIALVC: zij $I = (V, E, k)$ een instantie van VC. Dan is $f(I) = <V', E', v, k'>$ waarbij (in alle gevallen geldt dat $v,w \notin V$):

a. $V' = V \cup \{v\}$, $E' = E$ en $k' = k$-1 .
b. $V' = V \cup \{v,w\}$, $E' = E \cup \{\{v,w\}\}$ en $k' = k$+1 .
c. $V' = V \cup \{v,w\}$, $E' = E \cup \{\{v,w\}\}$ en $k' = k$.
d. $V' = V \cup \{v\}$, $E' = E \cup \{\{w',v\}: w' \in V\}$ en $k' = k$.

Uitwerking: Alternatief a. is niet juist: als G een VC heeft ter grootte van k-1, maar niet ter grootte van k, is $f(G)$ zeker een yes-instantie voor $k'=k$-1, maar G een no-instantie. Om een instantie (V, E, k) van VC te reduceren naar SPECIALVC moet je er voor zorgen dat een speciale knoop v toegevoegd aan V deel kan uitmaken van een vertexcover ter grootte van $k+1$ alss de oorspronkelijke VC instantie een vertex cover heeft ter grootte van k. Dit lukt door een nieuwe kant $\{v,w\}$ toe te voegen aan E met v en w als nieuwe knopen: een vertex cover van G' moet nu altijd v of w bevatten en de vertex cover ter grootte van k uit G kan meegenomen worden: alternatief b. is juist. Alternatief c. is niet juist: als G een maximale vertex cover heeft van $k - 1$ heeft G' een VC heeft ter grootte van k. Alternatief d. is niet juist: neem maar eens een volledige graaf G en zij $k = |V|$-1; nu heeft G een VC ter grootte van k en is $G'=(V', E')$ opnieuw een volledige graaf, maar heeft geen VC ter grootte van k, echter wel van $k+1$.

Vraag 10. Stel dat we een algoritme OHAM hebben voor HAMCIRCUIT: Gegeven een ongerichte graaf G, heeft G een hamiltoons circuit?
Bij aanroep van OHAM voor een graaf G, retourneert OHAM de waarde *true* als G een hamiltoons circuit heeft en *false* anders.
Neem nu het volgende programma:

```
input: ongerichte graaf G = (V,E)
begin
   C := ∅;
   if ¬OHAM(G) then return C;
    while E(G)≠∅
       selecteer een kant e ∈E(G);
       E(G) := E(G)−{e};
       if ¬OHAM((V(G),E(G)∪C)) then C:=C ∪{e};
    return C;
end
```

Hierin is $E(G)$ de verzameling van kanten van G en $V(G)$ de verzameling van knopen van G. Het bovenstaande algoritme toont aan dat

a. het vinden van een hamiltoons circuit in een graaf een polynomiaal probleem is als iedere aanroep van OHAM polynomiale tijd kost.
b. als het algoritme een niet lege verzameling C retourneert, deze verzameling kanten een hamiltoons circuit van G is.
c. het vinden van een hamiltoons circuit in een graaf een **EXP**-hard probleem is.
d. het vinden van een hamiltoons circuit een probleem in $\mathbf{P}^{\mathbf{NP}}$ is.

Uitwerking: Als de while-loop niet doorlopen wordt, wordt een lege verzameling geretourneerd: G heeft dan *geen* hamiltoonse cycle. Na afloop van iedere doorgang door de while-loop is het volgende waar (een zgn *invariant* van de algoritme):
$(V(G), E(G) \cup C)$ bevat een hamiltoonse cycle en als $C \neq \emptyset$, dan geldt voor elke deelverzameling C' van C : $(V(G), E(G) \cup C')$ heeft geen hamiltoonse cycle.

Als de while-lus doorlopen is, is $E(G)$ leeg. Derhalve moet C een verzameling van kanten zijn die een hamiltoonse cycle van G vormen. Hieruit is in polynomiale tijd een hamiltoonse cycle te construeren. Als iedere aanroep van OHAM polynomiaal veel tijd kost, is dit ook een polynomiaal algoritme: alternatief a. en b. zijn juist. OHAM is een algoritme voor een NPC probleem. Derhalve is d. juist. Op grond van dit algoritme kan geen uitspraak over de EXP-hardheid van HAMCIRCUIT worden gedaan: c.is onjuist.

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. Stel dat SAT$\in$**P.** Laat zien, dat dan geldt: **P = NP^{NP}**.

Uitwerking: Als SAT een P probleem is, geldt P=NP. Daarmee is $NP^{NP} = P^{P} = P$.

Vraag 12. Toon aan, dat als L een **coNP-**compleet probleem is en $L \leq A$ voor een probleem A, dan is A^c een **NP**-hard probleem.

Uitwerking: Om aan te tonen, dat A^c NP-hard is moeten we laten zien dat voor ieder probleem X in NP geldt $X \leq A^c$ of te wel, dat $X^c \leq A$. Omdat L coNP-compleet is en X^c een coNP probleem is, geldt $X^c \leq L$ en aangezien $L \leq A$, geldt nu met transitiviteit van $\leq$ ook $X^c \leq A$.

Vraag 13[1]. Laat $L \subseteq \Sigma_1^*$ en $M \subseteq \Sigma_2^*$ twee talen zijn waarvoor geldt $L = M$ en $SAT \leq L$.
Geef precies aan waarom M een **NP**-hard probleem is.

Uitwerking: L is een NP-hard probleem omdat SAT een NPC probleem is en $SAT \leq L$. Omdat $L = M$, bestaat er een polynomiale reductie $f: \Sigma_1^* \rightarrow \Sigma_2^*$ van L naar M: Kies $f(x) = x$. Derhalve is geldt $L \leq M$ en is M ook NP-hard.

Vraag 14. De probabilistische complexiteitsklasse **RP** is als volgt gedefinieerd:
$L \in$ **RP** alss er een polynomiale PrTm M bestaat en een positieve constante $0 < \varepsilon \leq 1$, zodat voor iedere input x van M geldt:

- Pr(M accepteert $x \mid x \in L$) $\geq \varepsilon$
- Pr(M accepteert $x \mid x \notin L$) = 0

Toon aan dat **RP** $\subseteq$ **BPP**. (Let op: ε mag kleiner dan 0.5 zijn!)

Uitwerking: Als $L \in$ **RP**, dan bestaat er met toepassing van het *amplificatielemma* zeker een RP machine M' en een $0 < \partial < 0.5$ zodat Pr(M' accepteert $x \mid x \in L$) $\geq 1 - \partial$ en Pr(M' accepteert $x \mid x \notin L$) = 0:
Herhaal de runs van M hiervoor k maal en accepteer x alss x tenminste eenmaal geaccepteerd wordt. Dan geldt: Pr(M' accepteert $x \mid x \in L$) $\geq 1 - (1- \varepsilon)^k$ en Pr(M' accepteert $x \mid x \notin L$) = 0.
Voor k groot genoeg geldt altijd $1 - (1- \varepsilon)^k \geq 1 - \partial$. Tevens geldt dan: Pr(M verwerpt $x \mid x \notin L$) = $1 > 1 - \partial$.. Maar uit deze twee volgt onmiddellijk $L \in$ BPP.

Vraag 15. Waarom is het feit dat het factorizeringsprobleem met quantum computing in polynomiale tijd oplosbaar is, geen directe reden om aan te nemen dat alle **NPC** problemen met quantum computing in polynomiale tijd oplosbaar zijn?

Uitwerking: het is bekend dat het factorizeringsprobleem zich in de doorsnede van NP en coNP bevindt en zeer waarschijnlijk geen NP-volledig probleem is. Het feit dat quantum computing voor een dergelijk probleem een exponentiele speedup geeft, is onvoldoende reden om aan te nemen dat ook NP-complete problemen in polynomiale tijd met quantum computing oplosbaar zullen zijn.

[1] Deze vraag is om voor mij bijna onverklaarbare redenen in een eerdere verkeerde formulering in het tentamen terechtgekomen. Het antwoord is dan ook niet meegewogen voor het eindresultaat.

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)

De firma WebExploIT heeft een opdracht gekregen van een bedrijf dat reklameboodschappen op strategische plaatsen op hun website wil aanbrengen. Het bedrijf toont hen een map van hun website. Zo'n map van een website kan als een gerichte graaf
$G = (V, E)$ worden gemodelleerd. Tevens hebben zij van de bezoekers van hun website een verzameling van t *trails* bijgehouden. Zo'n trail is een reeks pagina's van de website die typische bezoekers van hun website volgen. In de graaf G kunnen we deze trails weergeven door een verzameling van t gerichte paden $p_1, ..., p_t$ in G.
De vraag van het bedrijf aan WebExploIT is nu of het mogelijk is om met k advertenties alle (typische) bezoekers van hun website te bereiken, dwz, gegeven een graaf $G=(V, E)$, t gerichte paden $p_1, ..., p_t$ in G, en een integer k, is het mogelijk k knopen in V te selecteren, zodanig dat elk pad p_j tenminste één geselecteerde knoop (=advertentie) bevat?

a. Toon aan, dat het bovengenoemde probleem een **NP**-probleem is (≤5 regels).

b. Laat zien dat in het algemeen dit probleem een **NP**-hard probleem is door een expliciete polynomiale reductie te geven van een bekend **NPC**-probleem naar dit probleem. Het is hierbij niet nodig een correctheidsbewijs te geven. (≤15 regels). (HINT: Gebruik een reductie van SAT)

Uitwerking: Het probleem is een NP-probleem: markeer k knopen en ga na voor ieder van de t paden of ze zo'n knoop bevatten. Dit kost per pad p_t, $O(|p_t| \times k)$ tijd. Derhalve in totaal $O(\Sigma_t|p_t| \times k) = O(|I|^2)$ –tijd. Voor de reductie van SAT nemen we een instantie (U,C) van SAT en construeren een volledige graaf $G = (U \cup \neg U, E)$ voor alle propositieknopen en hun negatie. We construeren $t = |U| + |C|$ paden: voor elke u uit U wordt het pad $\{u, \neg u\}$ in G en voor elke clause $\{x1, ..., xk\}$ het pad $\{x1, ..., xk\}$ in de graaf G opgenomen. Laat $k = |U|$. Nu moet op ieder pad $\{u, \neg u\}$ precies één knoop opgenomen worden. Ieder clausepad wordt overdekt als er een knoop u geselecteerd is zodat u in de clause voorkomt. Uiteraard wordt een reductie van VC ook goed gerekend.

Vraag 17. (10 punten)

Gegeven is het volgende zogenaamde bewijs voor de stelling dat **NP** ∩ **coNP** ⊆ **P**

1. Laat L een probleem (taal) zijn over een alfabet Σ met $|\Sigma| \geq 2$ met $|\Sigma| \geq 2$ en $L \in$ **NP** ∩ **coNP**.
2. Omdat $L \in$ **NP**, geldt dat er voor iedere $x \in L$ een certificaat y bestaat met $|y| \leq p(|x|)$ voor een polynoom p om aan te tonen dat x een element is van L.
3. Omdat $L \in$ **co-NP**, geldt dat er voor iedere $x \notin L$ een certificaat z bestaat met $|z| \leq q(|x|)$ voor een polynoom q om aan te tonen dat x geen element is van L.
4. Laat nu x een willekeurig rijtje zijn over Σ^*. Dan bestaat er derhalve een certificaat y van polynomiale lengte om aan te tonen dat x een element is van L óf een certificaat z van polynomiale lengte om aan te tonen dat x geen element is van L.
5. Test nu in lexicografische volgorde alle rijtjes $u \in \Sigma^*$ om na te gaan of u een certificaat is voor $x \in L$ of voor $x \notin L$. Als het eerst gevonden certificaat een certificaat is voor $x \in L$, *accepteer* en anders *verwerp*.

6. Het eerst gevonden certificaat wordt in $max\{|y|,|z|\}$ tijd gevonden, derhalve wordt L in polynomiale tijd beslist.

Geef precies aan welke van de genummerde beweringen *incorrect* zijn en motiveer uw antwoord kort en bondig. ($\leq$ 10 regels)

Uitwerking: Alleen uitspraak 6 is onjuist. Aangezien de opsomming lexicografisch is en $|\Sigma| \geq 2$, wordt een certificaat worst-case in $O(2^{\max\{|y|,|z|\}})$-tijd gevonden. Dit is niet polynomiaal in $|x|$. Derhalve kan niet worden geconcludeerd dat L in polynomiale tijd wordt beslist.

EINDE TENTAMEN