

Tentamen TI3300

Complexiteitstheorie

15 april 2013,

9.00-12.00 uur

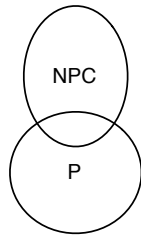
- Dit tentamen bestaat uit 10 meerkeuzevragen, 5 korte (open) vragen en 2 open vragen.
- Voor de meerkeuzevragen kunt u maximaal 50 punten behalen, voor de korte open vragen maximaal 30 punten en voor de open vragen maximaal 20 punten.
- Het totaal aantal punten voor de meerkeuzevragen (mp) wordt als volgt bepaald:
 - o per meerkeuzevraag i wordt het aantal foutief beantwoorde alternatieven f_i bepaald. aangezien per vraag 0,1,2,3 of 4 alternatieven juist kunnen zijn geldt: $0 \leq f_i \leq 4$;
 - o Laat $f = \sum f_i$ voor alle 10 meerkeuzevragen ($0 \leq f \leq 40$) ; het aantal punten voor de meerkeuzevragen (mp) is dan $mp = 50 \times \max\{0, (40 - 1.5 \times f) / 40\}$.
- Het eindcijfer c wordt bepaald door de som te nemen van mp, het totaal aantal punten behaald voor de 5 korte vragen (kv) en de som van het aantal punten behaald voor de twee open vragen (ov) en vervolgens het resultaat te delen door 10: $c = (mv + kv + ov) / 10$; dit cijfer wordt vervolgens afgerond op het dichtstbijzijnde halve of gehele getal.
- 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:

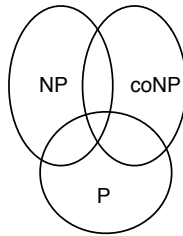
- $A \leq B$ betekent dat er een polynomiale-tijd reductie van A naar B bestaat.
- A^c is het complement van het probleem A.
- Er wordt steeds vanuit gegaan, dat **P** $\neq$ **NP** en **NP** $\neq$ **coNP**, tenzij uitdrukkelijk het tegendeel wordt vermeld.

MEERKEUZEVRAGEN (50 PUNTEN)

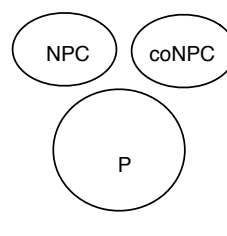
Vraag 1. Ongeacht of nu $P = NP$ of $P \neq NP$ geldt, sommige situaties kunnen zich *nooit* voordoen. Welke situaties in onderstaande Venndiagrammen zijn onmogelijk?



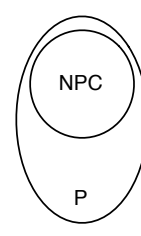
Figuur a



Figuur b



Figuur c



Figuur d

- Een situatie als in Figuur a is onmogelijk.
- Een situatie als in Figuur b is onmogelijk.
- Een situatie als in Figuur c is onmogelijk.
- Een situatie als in Figuur d is onmogelijk.

Antwoord: a. is onmogelijk omdat $NP \cap P \neq \emptyset$ impliceert $NP \subseteq P$.
b. is onmogelijk want er geldt altijd $P \subseteq NP$ en $P \subseteq coNP$.
c. is mogelijk (als $P \neq NP$).
d. is mogelijk (als $P = NP$).

Vraag 2. Sipser representeert beslissingsproblemen als talen over een alfabet. Zo'n taal is op te vatten als een deelverzameling van yes-instanties van het probleem. Ga na welke van de volgende uitspraken noodzakelijkerwijs juist zijn:

- Als A_1 en A_2 beide **NP**-complete problemen zijn, dan geldt $A_1 \leq A_2$ en $A_2 \leq A_1$.
- Als A_1 een **NP**-complete probleem is en $A_2 \leq A_1$ dan geldt ook $A_1 \leq A_2$.
- Als $A_1 \leq A_2$ en $A_2 \leq A_1$ en $A_1, A_2 \in \mathbf{NP}$, dan zijn A_1 en A_2 beide **NP**-complete.
- Als $A_1 \subseteq A_2$ dan geldt ook $A_1 \leq A_2$.

Antwoord Alternatief a. is juist omdat alle problemen in **NP** polynomiaal reduceerbaar zijn tot elkaar.
Alternatief b. is niet noodzakelijk juist: Kies bv. $A_2 = \emptyset$.
Alternatief c. is niet noodzakelijkerwijs juist: beide problemen kunnen problemen in **P** zijn en we veronderstellen $P \neq \mathbf{NP}$.
Alternatief d. is onjuist: neem maar eens $\emptyset$ en Σ^* . Er geldt $\emptyset \subseteq \Sigma^*$ maar niet $\emptyset \leq \Sigma^*$.

Vraag 3. A, B en C zijn drie beslissingsproblemen. Van A is bekend dat $A \in \mathbf{coNP} \cap \mathbf{NP}$. Voor B geldt $C \leq B$ en C is een **NP**-hard probleem. Er geldt nu

- Als $A \leq C$, dan geldt $P = \mathbf{NP}$.
- Als $C \leq A$, dan geldt $\mathbf{NP} = \mathbf{coNP}$.
- $A \leq B$.
- Als $B \leq A$ dan geldt $A \in \mathbf{NP-hard}$.

Antwoord a. is onjuist: ieder probleem in **NP** is polynomiaal reduceerbaar naar C, dus A ook, zonder dat dit consequenties heeft voor **NP** en **coNP**.
b. is juist: Als $C \leq A$, dan is A ook een **NP**-hard probleem. Maar dan geldt dat $A \in \mathbf{NPC}$ (immers $A \in \mathbf{NP}$). Derhalve geldt $\mathbf{coNP} \cap \mathbf{NPC} \neq \emptyset$. En daaruit volgt $\mathbf{NP} = \mathbf{coNP}$.
c. is juist: Uit $C \leq B$ volgt $B \in \mathbf{NP-hard}$ en aangezien $A \leq C$, geldt derhalve $A \leq B$.
d. is juist. Als $B \leq A$, dan is A een **NP**-hard probleem: $C \leq B$ en $B \leq A$ tezamen met het gegeven dat $C \in \mathbf{NP-hard}$ impliceert dat $A \in \mathbf{NP-hard}$.

Vraag 4. De klasse **NP** bestaat uit die problemen A waarvoor geldt dat een yes-instantie van A in polynomiale tijd verifieerbaar is. Laat de klasse **X** bestaan uit problemen B waarvoor geldt dat een yes-instantie van B in logaritmische tijd ($O(\log n)$) verifieerbaar is. Er geldt dan:

- a. $X \subseteq P$.
- b. $X \subseteq NP \cap \text{co-NP}$.
- c. $X \subseteq NP$.
- d. $X \subseteq EXP$.

Antwoord Alle alternatieven zijn juist: Als yes-instanties in $O(\log n)$ tijd verifieerbaar zijn kunnen alle certificaten voor yes-instanties in $2^{O(\log n)} = O(n)$ tijd opgesomd worden en één voor één gecheckt worden. Dit kost in totaal $O(n \log n)$ -tijd. Derhalve kan in polynomiale tijd beslist worden of een X-instantie een yes-instantie is. Dus $X \subseteq P$ en alternatief a. is juist. Maar dan zijn ook alle overige alternatieven juist omdat P een subset is van alle genoemde complexiteitsklassen.

Vraag 5. Welke van onderstaande beweringen is/zijn correct?

- a. Ieder **PSPACE**-compleet probleem is een **NP**-hard probleem.
- b. $P^{NP} \subseteq NP^{NP} \subseteq PSPACE^{PSPACE} = PSPACE$.
- c. Als A een **PSPACE**-compleet probleem is dan is A^c ook een **PSPACE**-compleet probleem.
- d. $P = PSPACE$ impliceert $NP = \text{coNP}$.

Antwoord a. is juist, want $NP \subseteq PSPACE$, dus ieder probleem in NP is reduceerbaar tot een PSPACE compleet probleem.
b is juist (bedenk dat een PSPACE orakel polynomiale ruimte kost: $PSPACE^{PSPACE} = PSPACE$).
c. is ook juist omdat PSPACE gesloten is onder complement.
d. is juist want $P = PSPACE$ impliceert $P = NP$.

Vraag 6. Neem het volgende probleem EXT-SAT:

Gegeven: een satisfiability instantie (U, C) waarvan bekend is dat C vervulbaar is en een nieuwe clause c' over U.

Vraag: is $C \cup \{c'\}$ vervulbaar?

Welke van de volgende reducties van het SAT-probleem naar dit probleem kan gebruikt worden om aan te tonen dat dit probleem **NP**-hard is onder $\leq$?

Neem een willekeurige instantie (U_0, C_0) van het SAT-probleem en creëer de volgende instantie van het EXT-SAT-probleem:

- a. $U = U_0 \cup \{z\}$, $C = \{c \cup \{z\} \mid c \in C_0\}$ en $c' = \{\neg z\}$, waarbij $z \notin U_0$
- b. $U = U_0 \cup \{z\}$, $C = C_0 \cup \{\{z\}\}$ en $c' = \{\neg z\}$, waarbij $z \notin U_0$
- c. $U = U_0 \cup \{y, z\}$, $C = \{c \cup \{y\} \mid c \in C_0\}$ en $c' = \{\neg y, \neg z\}$, waarbij $y, z \notin U_0$
- d. $U = U_0 \cup \{y, z\}$, $C = \{c \cup \{y\} \mid c \in C_0\} \cup \{\{z\}\}$ en $c' = \{\neg y, \neg z\}$, waarbij $y, z \notin U_0$

Antwoord Alternatief a. is juist: (U, C) is triviaal vervulbaar door z waar te maken en $(U, C \cup \{c'\})$ is vervulbaar dan en slechts dan als (U_0, C_0) vervulbaar is: de vervulbaarheid van $(U, C \cup \{c'\})$ vereist dat z onwaar wordt gemaakt en (U_0, C_0) vervulbaar is.
Alternatief b. is niet juist: $(U, C \cup \{c'\})$ is altijd onvervulbaar, ook al is (U_0, C_0) vervulbaar.
Alternatief c. is niet juist: (U_0, C_0) kan onvervulbaar zijn terwijl - met y is waar en z is onwaar - $(U, C \cup \{c'\})$ vervulbaar is.
Alternatief d. is juist: $(U, C \cup \{c'\})$ is vervulbaar desda z is waar en y is onwaar en (U_0, C_0) vervulbaar.

Vraag 7. In een schedulingsprobleem moet een aantal jobs met bekende processingtijden over twee sequentiële machines verdeeld worden zodat de processingtijd die beide processors nodig hebben om alle jobs te voltooien zo klein mogelijk is. We nemen aan dat we hiervoor een benaderingsalgoritme X kunnen gebruiken dat de jobs aan de machines toewijst. We gebruiken X voor een instantie waarbij de som van de processingtijden van alle jobs tezamen gelijk is aan 200. Voor deze instantie verdeelt X deze jobs zodanig over de machines dat de tijd die de twee machines nodig hebben om alle jobs uit te voeren gelijk is aan 110. Er geldt nu dat:

- a. de performance ratio van X voor deze instantie gelijk is aan 1.1
- b. de performance ratio van X voor deze instantie kleiner of gelijk is aan 1.1
- c. de approximatieratio van X gelijk is aan 1.1
- d. de approximatieratio van X nooit kleiner kan zijn dan 1.1

Antwoord Alleen b. is juist. Bedenk dat uit de gegevens alleen maar is af te leiden dat $100 \leq opt \leq 110$. De performance ratio van X voor de instantie is dan ook hoogstens gelijk aan 1.1; Over de approximatieratio valt niets te zeggen omdat de performance ratio van andere instanties niet bekend is.

Vraag 8. Van een **NPC** probleem is bekend, dat als de getallen die erin voornemen unair worden gecodeerd (dwz een getal n wordt gerepresenteerd door n^2 en), er een algoritme bestaat dat het probleem oplost in kwadratische tijd. Hieruit volgt:

- a. Dit probleem is een sterk-**NPC** probleem.
- b. Dit probleem kan niet sterk-**NPC** zijn, tenzij $P = NP$.
- c. Dit probleem heeft een pseudo-polynomiaal algoritme.
- d. De verzameling van instanties waarvoor geldt dat alle erin voorkomende getallen kleiner zijn dan de lengte van de instantie, is in polynomiale tijd oplosbaar.

Antwoord Uit het gegeven volgt dat het probleem voor polynomiale grootte van getallen in polynomiale tijd oplosbaar is. Het is derhalve geen sterk-NPC probleem (tenzij $P = NP$) en heeft een pseudo-polynomiaal algoritme:
a. is onjuist, b. is juist, c. is juist, d. is juist.

Vraag 9. Stel er bestaat een probabilistisch algoritme voor een **NP**-compleet beslissingsprobleem A . Voor een no-instantie van A geeft dit probleem altijd een correct antwoord, voor een yes-instantie van dit probleem geeft het algoritme met een kans 0.1 een goed antwoord. Wat volgt uit deze gegevens?

- a. A is een **P**-probleem.
- b. A is een **BPP**-probleem.
- c. Er bestaat een probabilistisch algoritme voor A dat met een kans van tenminste .67 altijd een goed antwoord geeft.
- d. $NP \subseteq BPP$.

Antwoord Bedenk dat *niet* gegeven is dat het probabilistisch algoritme een polynomiaal algoritme is! Wel geldt dat als je het algoritme m x herhaalt, de kans op een incorrect antwoord voor een yes-instantie gelijk is aan 0.9^m en een incorrect antwoord op een no-instantie gelijk is aan 0. Als m groot genoeg is, is de kans op een foutief antwoord $\leq .33$.

Conclusie:

- a. is onjuist
- b. is onjuist: er is niet gegeven dat het algoritme polynomiale tijd vergt.
- c. is juist: zie de redenering gevolgd onder b.
- d. is onjuist, zie b.

Vraag 10. In het college zijn drie thesen besproken: de Church-Turing These, de Sterke Church Turing These en de Sterke $P \neq NP$ These. Als we naast de klassieke fysica ook quantum fysische berekeningen betrekken, wat is dan de status van bovenstaande thesen?

- a. We moeten concluderen dat alle drie deze thesen onhoudbaar zijn geworden door recente resultaten uit de quantum computing.
- b. Op grond van resultaten uit de quantum computing zal vermoedelijk de Sterke Church Turing These verworpen moeten worden.
- c. Er zijn sterke aanwijzingen dat de Sterke $P \neq NP$ these verworpen zal moeten worden op grond van het algoritme van Grover.
- d. Het factorisatie-algoritme van Schor toont aan dat ook **NP**-complete problemen kunnen worden opgelost in polynomiale tijd met quantum computing. De Sterke Church Turing these en de Sterke $P \neq NP$ These kunnen derhalve als verworpen worden beschouwd.

Antwoord:

- a. is onjuist: de Church Turing these blijft sowieso onaangetast.
- b. is juist: vermoedelijk is het factorizeringsprobleem geen P-probleem en er bestaat een polynomiaal quantum computing algoritme voor dit probleem.
- c. is onjuist, het algoritme van Grover laat juist zien dat iedere brute-force aanpak van een NPC probleem slechts een kwadratische ipv exponentiele speed-up te zien zal geven met quantum computing.
- d. is onjuist. Het factoriseringsprobleem is voorozver we weten geen NP-compleet probleem. Voor NPC problemen zijn geen efficiënte quantum algoritmen bekend.

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. We weten dat geldt $P \subseteq NP \subseteq PSPACE \subseteq EXP$.

Welke stelling kan gebruikt worden om aan te tonen dat tenminste één van deze inclusierelaties een strikte inclusierelatie is?

Antwoord: Met de tijd-hierarchiestelling kunnen we laten zien dat P een strikte subset is van EXP .
Derhalve geldt $P \subsetneq EXP$.

Vraag 12. Een student heeft het volgende bewijs voor $P=NP$:

Het is bekend dat 2-SAT een P -probleem is. Neem een willekeurige 3-SAT instantie. Zo'n instantie bestaat uit clauses die ieder 3 literals hebben. Beschouw de volgende reductie:
Vervang iedere clause $c_i = x_i \vee y_i \vee z_i$ door een drietal clauses $x_i \vee b_i$, $b_i \vee y_i$ en $b_i \vee z_i$, waarbij voor iedere clause c_i een nieuwe unieke literal b_i gekozen wordt.
Deze transformatie is een polynomiale reductie naar 2-SAT. Omdat nu geldt $3-SAT \leq 2-SAT$ en P naar beneden gesloten is onder $\leq$, geldt $3-SAT \in NPC \cap P$. Maar dan moet gelden $P=NP$.
Geef exact aan (≤ 5 regels) waarom de voorgestelde reductie niet aan de voorwaarden voldoet.

Antwoord De transformatie is niet yes-instantie behoudend. Kies maar een 3-SAT instantie die onvervulbaar is. Neem een waarheidstoekenning die alle literals b_i waarmaakt in de gecreeerde 2-SAT instantie. Zo'n waarheidstoekenning maakt de 2-SAT-instantie waar. De transformatie voldoet derhalve niet aan de eisen van een polynomiale tijd-reductie.

Vraag 13. Welk bekend probleem kan gebruikt worden om onmiddellijk te laten zien dat het volgende probleem een **NP-hard** probleem is:

MINIMUM 2-BLAD OPSPANNENDE BOOM

Instantie: Een ongerichte graaf $G = (V, E)$

Vraag: Is er een opspannende boom T voor G met precies 2 bladeren?

Antwoord Een opspannende boom van een graaf G met 2 bladeren is een hamiltons pad van G .
Het bestaan van zo'n hamiltons pad is een NP-hard probleem: $HAMP \in NPC$.

Vraag 14. Welke twee eisen worden aan een probleem gesteld dat in de parallelle complexiteitsklasse klasse NC (Nick's Class) ligt?

Antwoord Zo'n probleem moet oplosbaar zijn in polylogaritmische (parallelle) tijd met een polynomiaal aantal processoren. Dwz. voor zo'n probleem moet een circuit familie bestaan met polylogaritmische diepte complexiteit en polynomiale grootte complexiteit.

Vraag 15. Noem het beste voorbeeld van een quantum algoritme dat de Sterke Church Turing These ondermijnt.

Antwoord Het factoriseringsalgoritme van Schor laat zien dat een probleem in $NP \cap coNP$ in polynomiale tijd oplosbaar is met een quantum algoritme. Dit is met klassieke berekeningsprocessen (nog) niet mogelijk gebleken.

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)

FINDPARTITION is het probleem om, gegeven een verzameling items $A = \{a_1, a_2, \dots, a_n\}$ en een gewichtsfunctie $g: A \rightarrow \mathbb{Z}^+$, een deelverzameling A' van A te bepalen (als zo'n deelverzameling bestaat) zodanig dat de som $g(A') = \sum_{a \in A'} g(a)$ van de gewichten van de items $a \in A'$ gelijk is aan $g(A - A')$, de som van de gewichten van items in $A - A'$. Als zo'n deelverzameling A' niet bestaat moet `nil` als antwoord worden gegeven.

Geef een polynomiaal algoritme voor FINDPARTITION waarbij een algoritme voor PARTITION hoogstens n -maal als orakel mag worden gebruikt.

Hint: Neem het eerste element a_1 uit A . Als er een oplossing bestaat, bevindt dit element zich in een helft (A' of $A - A'$). Ga nu verder met element a_2 . Als dit element zich in dezelfde helft als a_1 bevindt, dan moet de instantie met als verzameling items $A - \{a_1, a_2\} + \{b_{12}\}$, waarbij $g(b_{12}) = g(a_1) + g(a_2)$ ook een oplossing hebben. Dit kun je met het orakel controleren. Ga door met het onderzoeken van de volgende elementen a_i .

Antwoord

```
begin
  if PARTITION(A,g) = "NO" then nil
  else
    b = a1;
    A' = {a1};
    A := A - {a1};
    g(b) = g(a1);
    for i=2 to n do
      B := A - {ai} + {b};
      g(b) = g(b) + g(ai);
      if PARTITION(B,g) then
        A' := A' ∪ {ai};
        A := B;
      else
        g(b) := g(b) - g(ai);
      return A';
    end
```

Vraag 17. (10 punten)

Soms willen we nagaan of in een sociaal netwerk er een voldoende grote clique van personen bestaat die alle leden van een zekere subgroep omvat. Het vinden van zo'n clique blijkt niet triviaal. Beschouw hiervoor het volgende probleem:

$\text{SPECIALECLIQUE} = \{ \langle G = \langle V, E \rangle, W, k \rangle \mid G \text{ is een ongerichte graaf met een clique } C \subseteq V \text{ ter grootte van } k \text{ zodanig dat } \emptyset \neq W \subseteq C \}$

Laat zien, dat dit probleem **NP-hard** is door een polynomiale-tijdreductie te geven van CLIQUE naar SPECIALECLIQUE. Het is hierbij niet nodig de correctheid van de reductie te bewijzen. Het geven van een expliciet programma dat de reductie uitvoert is voldoende.

Hint: bedenk dat W een verzameling kan zijn die slechts één knoop bevat.

Antwoord Gegeven een instantie $(G = \langle V, E \rangle, k)$ van CLIQUE, construeer een instantie $G' = \langle V', E', W, k' \rangle$ van SPECIALECLIQUE waarbij

- $V' = V \cup \{x\}$,
- $E' = E \cup \{\{x, v\} : v \in V\}$
- $k' = k + 1$,
- $W = \{x\}$.

x is met alle knopen in V verbonden. $G = (V, E)$ heeft daarom een clique C ter grootte van k alss G' een clique C' ter grootte van $k+1$ heeft die x bevat: $C' = C \cup \{x\}$.

EINDE TENTAMEN