

Tentamen IN3105

Complexiteitstheorie

31 maart, 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 $f = \sum f_i$ van dit aantal wordt bepaald ($0 \leq f \leq 40$)
 - o het aantal punten mp is dan gelijk aan $mp = 50 \times \max\{0, (40 - 1.5 \times f) / 40\}$. Hierbij is de factor 1.5 een milde gokcorrectiefactor.
- 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.
- 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 meerkeuze vragen 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 $NP \neq co-NP$, tenzij uitdrukkelijk het tegendeel wordt vermeld.

MEERKEUZEVRAGEN

Vraag 1. Het probleem om, gegeven een ongerichte graaf $G = (V, E)$ en een integer K te bepalen of G *géén* vertex cover ter grootte van K bevat, is een

- a. **NP**-volledig probleem.
- b. **co-NP**-volledig probleem.
- c. **P**-probleem.
- d. probleem in $\mathbf{NP} \cap \mathbf{co-NP}$.

Antwoord Het probleem is een co-NP probleem: alleen nee-instanties zijn polynomiaal verifieerbaar. Omdat het complement van het probleem een NPC probleem is, is het probleem een co-NPC probleem. Alternatief b. is juist en de overige alternatieven zijn onjuist.

Vraag 2. Als A een probleem in $\mathbf{NP} \cap \mathbf{co-NP}$ is, dan geldt dat

- a. het mogelijk is dat A een polynomiaal algoritme heeft, ook al is $\mathbf{P} \neq \mathbf{NP}$.
- b. yes-instanties en no-instanties van A polynomiaal verifieerbaar zijn.
- c. A geen **NP**-compleet probleem is, tenzij $\mathbf{NP} = \mathbf{co-NP}$.
- d. A geen **co-NP**-compleet probleem is, tenzij $\mathbf{NP} = \mathbf{co-NP}$.

Antwoord Omdat P bevat is in $\mathbf{NP} \cap \mathbf{co-NP}$ is alternatief a. juist; alternatief b. is juist op grond van de definitie van $\mathbf{NP} \cap \mathbf{co-NP}$. Indien A een **NPC** probleem zou zijn, dan zou gelden, dat $\mathbf{NP} = \mathbf{co-NP}$, evenzo als A een **co-NPC** probleem zou zijn, zou gelden $\mathbf{NP} = \mathbf{co-NP}$. Alternatief c. en d. zijn derhalve juist.

Vraag 3. Als $\mathbf{P} = \mathbf{NP}$ dan geldt in ieder geval dat:

- a. alle **NP**-problemen tevens **NP**-complete problemen zijn.
- b. $\mathbf{co-NP} = \mathbf{NP}$.
- c. $\mathbf{NP} = \mathbf{PSPACE}$.
- d. $\mathbf{APX} = \mathbf{NPO}$.

Antwoord Alternatief a. is niet juist omdat de P-problemen $\emptyset$ en Σ^* niet volledig kunnen zijn. Alternatief b. is juist omdat P gesloten is onder complement. Alternatief c. is niet juist: als $\mathbf{P} = \mathbf{NP}$ kan P nog steeds ongelijk zijn aan PSPACE. Alternatief d. is juist: als $\mathbf{P} = \mathbf{NP}$ geldt $\mathbf{NPO} = \mathbf{PO}$ en dus $\mathbf{APX} = \mathbf{PO} = \mathbf{NPO}$.

Vraag 4. Als A en B **NP**-problemen zijn, dan is volgens de definitie van Sipser:

- a. $A \cap B$ een **P**-probleem.
- b. $A \cup B$ een **NP**-probleem.
- c. $A \times B$ een **NP**-probleem.
- d. $A^c \cap B^c$ een **co-NP**-probleem.

Antwoord Alternatief a is onjuist (neem maar eens $A = B$ een NPC probleem). Alternatief b. en c. zijn juist: NP is gesloten onder vereniging en cartesisch product. Alternatief d. is ook juist, omdat A^c en B^c beide co-NP-problemen zijn en co-NP gesloten is onder doorsnede

Vraag 5. We noemen een optimaliseringsprobleem A *benaderbaar* als A een approximatie-algoritme heeft met approximatie-ratio $c < \infty$. Dan geldt:

- a. Er bestaat minstens één **NPO**-probleem dat niet benaderbaar is.
- b. Alle **APX**-problemen zijn benaderbaar.
- c. Alle **FPTAS**-problemen zijn benaderbaar.
- d. Als een **NPO**-probleem benaderbaar is, dan ligt het in de klasse **APX**.

Antwoord Omdat TSP geen approximatie-algoritme heeft waarvoor de approximatie-ratio naar boven door een constante begrensd is, is alternatief a. waar. Alternatief b. is per definitie van APX waar. Alternatief c. is per definitie van FPTAS waar evenals alternatief d.

Vraag 6. Welke van onderstaande beweringen zijn correct onder de aanname dat $\mathbf{NP} \neq \mathbf{P}$?

- a. Een probleem kan niet zowel in **NP** als in **co-NP** liggen.
- b. Sterk-**NP**-volledige problemen kunnen een pseudo-polynomiaal algoritme hebben.
- c. Voor ieder tweetal problemen A en B in **NP** geldt: $A \leq B$ of $B \leq A$.
- d. Voor ieder **NP**-probleem A bestaat er een sterk-**NP**-volledig probleem B waarvoor $A \leq B$.

Antwoord Omdat $\mathbf{P}$ omvat wordt door zowel $\mathbf{NP}$ als $\mathbf{co-NP}$ is alternatief a. onwaar. Alternatief b. is onwaar, evenals alternatief c. : neem maar $A = \emptyset$ en $B = \Sigma^*$. Omdat er minstens een sterk NP volledig probleem bestaat (zoals CLIQUE) is alternatief d. waar.

Vraag 7. 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?

De volgende reductie van het SAT-probleem kan gebruikt worden om aan te tonen dat dit probleem **NP**-hard is onder polynomiale-tijd reducties:

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

- $U = U_0 \cup \{z\}$, $C = \{c \cup \{z\} \mid c \in C_0\}$ en $c' = \{\neg z\}$, waarbij $z \notin U_0$
- $U = U_0 \cup \{z\}$, $C = C_0 \cup \{z\}$ en $c' = \{\neg z\}$, waarbij $z \notin U_0$
- $U = U_0 \cup \{y, z\}$, $C = \{c \cup \{y\} \mid c \in C_0\}$ en $c' = \{\neg y, \neg z\}$, waarbij $y, z \notin U_0$
- $U = U_0 \cup \{y, z\}$, $C = \{c \cup \{y\} \mid c \in C_0\} \cup \{z\}$ 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, omdat $(U, C \cup \{c'\})$ onvervulbaar is.

Alternatief c. is niet juist, omdat (U_0, C_0) onvervulbaar kan zijn terwijl met y is waar en z is onwaar $(U, C \cup \{c'\})$ vervulbaar is.

Alternatief d. is juist: (U, C) is triviaal vervulbaar en $(U, C \cup \{c'\})$ is vervulbaar desda z is waar en y is onwaar en (U_0, C_0) vervulbaar.

Vraag 8. Gegeven zijn n processoren P_1 t/m 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:

- een **NP**-volledig probleem, immers het is een restrictie van het PARTITION probleem.
- een **NP**-volledig probleem: PARTITION is een restrictie van dit probleem.
- een getal probleem.
- Voor $n = 1$ een polynomiaal probleem en voor $n \geq 2$ een **NP**-volledig probleem.

Antwoord Alternatief a. is onjuist, alternatief b. is juist. Alternatief c. is ook juist, evenals alternatief d.

Vraag 9. Als A een **NP**-probleem is, dan geldt dat

- voor A geen polynomiaal algoritme bestaat.
- voor iedere instantie van A een accepterend berekeningspad van polynomiale lengte bestaat.
- yes-instanties van A polynomiaal verifieerbaar zijn.
- er een backtracking algoritme voor A bestaat en een polynoom $p()$, zodanig dat iedere instantie x uit A in $O(2^{p(|x|)})$ -tijd kan worden opgelost.

Alternatief a. is onjuist, alternatief b. is onjuist, alternatief c. is juist, alternatief d. is juist.

Vraag 10. Welke van de onderstaande inclusierelaties zijn geldig:

- a. $\mathbf{NP} \subseteq \mathbf{RP}$
- b. $\mathbf{NC} = \mathbf{P}$
- c. $\mathbf{NPC} \subseteq \mathbf{PP}$
- d. $\mathbf{P} \subseteq \mathbf{BPP}$

Alternatief a. is onjuist, alternatief b. is onjuist, alternatief c. is juist, alternatief d. is juist.

OPEN VRAGEN I

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

Vraag 11. Geef precies aan wat het verschil is tussen **NP** en de klasse **P^{NP}**.

Antwoord In de klasse **P^{NP}** functioneert NP als een orakelklasse en zal een NP orakelalgoritme voor zowel yes- als no-instanties een antwoord geven in $O(1)$ -tijd. Er volgt onmiddellijk dat $P \subseteq P^{NP}$ en dat er geen verschil is tussen **P^{NP}** en **P^{co-NP}**. Derhalve omvat **P^{NP}** de klassen NP en co-NP.

Vraag 12. Het MAX-CLIQUE probleem is het probleem om, gegeven een graaf $G = (V, E)$ en een positieve integer K , te bepalen of de *maximale* grootte van een clique in G gelijk aan K is. Geef een **P^{NP}** algoritme voor dit probleem dat slechts twee orakelaanroepen doet.

Antwoord

```
begin
  if CLIQUE( $G, K$ )="yes" then
    if CLIQUE( $G, K+1$ )="no" then
      return "yes"
    return "no"
end
```

Hierin is $\text{CLIQUE}(G, K)$ een aanroep van een orakel voor het CLIQUE probleem met parameters G en K .

Vraag 13. In het door Sipser besproken Generalized Geography probleem zijn er twee spelers die om beurten een knoop kiezen in een gerichte graaf G . Speler 1 mag beginnen door uitgaande van een startknoop s een van de burens v van s te kiezen waarna speler 2 een buur van de door speler 1 gekozen knoop v mag kiezen etc. Een speler die als eerste gedwongen wordt een knoop te kiezen die al eerder gekozen is, verliest. De vraag is nu voor een gegeven graaf G en een gegeven startknoop s te bepalen of Speler 1 een winnende strategie heeft. Onderzoeker X denkt dat dit een **NP**-volledig probleem is. Immers als er een certificaat in de vorm van een sequentie van knopen $v_1, w_1, v_2, w_2, \dots, v_n, w_n$ of $v_1, w_1, v_2, w_2, \dots, v_n$ gegeven wordt waarbij v_i de i -de zet van Speler 1 en w_i de i -de zet van Speler 2 voorstelt kan eenvoudig worden vastgesteld wie van beide spelers een winnende strategie heeft.

Welke ernstige fout(en) maakt deze onderzoeker?

Antwoord Een dergelijk certificaat laat alleen maar zien dat Speler 1 voor de *gegeven sequentie van keuzes* gewonnen dan wel verloren heeft en niet of Speler 1 een winnende strategie heeft. In dat laatste geval moet er voor elke serie keuzes die Speler 2 kan maken een serie van keuzes voor Speler 1 bestaan die deze speler uiteindelijk zal laten winnen. Derhalve zou een potentieel exponentieel aantal certificaten in de vorm van een gegeven spelverloop nodig kunnen zijn om dit aan te tonen.

Vraag 14. Geef zo nauwkeurig mogelijk de relaties ($=$, $\subset$ of $\subseteq$) tussen de volgende klassen: **P**, **NP**, **NP^P**, **NPC**, **P^{NP}** en **NP^{NP}** (niet noodzakelijk in deze volgorde)

Antwoord $P \subseteq NP = NP^P \subseteq P^{NP} \subseteq NP^{NP}$. Verder geldt $NPC \subset NP$.

Vraag 15. Wat is de relatie tussen het *black-box* model in de complexiteitstheorie voor bepaling van de complexiteit van een probleem (zoals besproken in de behandeling van quantum computing) en het gebruik van *backtracking algoritmen* voor de oplossing van NP-complete problemen?

Antwoord Het black-box model definieert een klasse van problemen waarbij de complexiteit gemeten wordt in termen van het aantal keren dat een orakel f geraadpleegd moet worden om de oplossing voor een probleem te vinden. Als dit orakel polynomiale complexiteit heeft, meet het black-box model precies de complexiteit van uitputtende zoekmethoden zoals backtracking voor het oplossen van NPC problemen.

OPEN VRAGEN II

Beantwoord onderstaande vragen zo kort mogelijk.

Geef geen irrelevante informatie, dit kan tot puntenaftrek aanleiding geven.

Vraag 16.

Gegeven is een verzameling P van processoren en een verzameling $L \subseteq P \times P$ van links tussen processoren. Een link $(p, q) \in L$ betekent dat processor p een boodschap naar processor q kan sturen. Soms moet een boodschap van een processor p naar een processor q gestuurd worden via een specifieke verzameling T van processoren in P als tussenstations. Deze tussenstations mogen hierbij echter slechts één keer aangedaan worden.

Geef een expliciete constructie van de reductie van een bekend NPC probleem naar het volgende probleem:

Gegeven: een verzameling P van processoren, een verzameling $L \subseteq P \times P$ van links tussen processoren in P , twee gegeven processoren p en q in P en een verzameling $T \subseteq P$ van tussenstations,

Vraag: is het mogelijk een boodschap te sturen van p naar q via links in L zodat alle processoren uit T precies éénmaal worden aangedaan?

U kunt hierbij kiezen uit de volgende NPC problemen: VC, PARTITION, CLIQUE, HAMPAD, HAMCYCLE, SUBSUM, SAT en KNAPSACK.

Antwoord Gegeven een willekeurige instantie $I = (G = (V, E), s, t)$ van het HAMPAD probleem, construeer de volgende instantie $f(I) = (P, L, s, t, T)$ met $P = V$, $L = E$ en $T = P - \{s, t\}$. NB: Indien u het HAMPAD probleem hebt gedefinieerd als $\text{HAMPAD} = \{G = (V, E) : G \text{ heeft een hamilton's pad}\}$ wordt de reductie iets anders: Gegeven een instantie $I = (G = (V, E))$ construeer een instantie $f(I) = (P, L, s, t, T)$ met $P = V \cup \{s, t\}$, $L = E \cup \{\{s, v\}, \{t, v\} \mid v \in V\}$, $T = P$. Hierbij mogen s en t niet voorkomen in V .

Vraag 17.

Gegeven zijn $N > 1$ vliegtuigen en $K \geq 1$ startbanen. Van ieder vliegtuig v is bekend het aantal $st(v)$ seconden dat het vliegtuig nodig heeft om op te stijgen (dwz de startbaan bezet houdt).

Elke startbaan mag gedurende een totale hoeveelheid tijd B , gemeten in seconden vanaf een zeker tijdstip $t = 0$ gebruikt worden.

Gevraagd wordt of er een toewijzing van startbanen en vertrektijdstippen aan de N vliegtuigen bestaat zodanig dat alle vliegtuigen voor de deadline B kunnen starten.

Hierbij moet de volgende voorwaarden in acht worden genomen:

- (1) als vliegtuig v en vliegtuig w gebruik maken van dezelfde startbaan en w komt na v , dan moet altijd gelden: $s(w) \geq s(v) + st(v)$
- (2) voor alle vliegtuigen v geldt: $s(v) \leq B - st(v)$.

Hierbij duidt $s(v)$ het vertrektijdstip van een vliegtuig v aan.

Kies een geschikt **NP**-compleet probleem A en formuleer de reductie van dit probleem A naar bovengenoemd probleem met behulp van restrictie (≤ 4 regels).

U kunt hierbij kiezen uit de volgende problemen: SAT, PARTITION, VC, HAMPAD, HAMCYCLE, SUBSUM, CLIQUE, KNAPSACK.

Antwoord Beschouw instanties I' van het bovengenoemde start-toewijzingsprobleem met $I' = (V, S, st(), B)$ waarin $B = \lfloor st(V)/2 \rfloor$ en $S = \{s1, s2\}$. Er bestaat nu een simpele bijectie van PARTITION instanties naar deze restrictie van het start-toewijzingsprobleem: Voor PARTITION instanties $I = (A, g())$ kies $V = A$ en $st() = g()$. Dan volgt onmiddellijk dat de verzameling van alle instanties een NP-hard probleem is.