Vierde (en laatste) huiswerkopgave Complexiteit 2019
Inleveren: dinsdag 14 mei 2019
Geprinte versie → kamer 159 of (werk)college
Maak de uitwerking in LaTeX!

Geef een *duidelijke* toelichting/uitleg bij al je antwoorden!

a. (20 punten)
In deze opgave moet een programma voor een deterministische Turingmachine (DTM) worden geschrevenn. In Hoofdstuk 11 van het dictaat (p. 51–52) en de sheets van het bijbehorende college staan enkele definities en een voorbeeld van een DTM-programma. Zo'n programma wordt gegeven in de vorm van een transitiefunctie $\delta: Q \setminus \{q_Y, q_N\} \times \Gamma \longrightarrow Q \times \Gamma \times \{-1, 0, 1\}$, met Q de verzameling toestanden van de DTM, en Γ de verzameling gebruikte tape-symbolen met $\{0, 1, b\} \subseteq \Gamma$ (b staat voor "blanco"). Geef duidelijk aan welke karakters Γ bevat.
Schrijf een DTM-programma dat voor een gegeven invoerstring x bestaande uit nullen en enen het volgende doet. Eerst wordt nagegaan of de string leeg is, of uit louter nullen bestaat of uit louter enen bestaat. Als dat het geval is moet het programma termineren in de nee-toestand. Indien de string niet leeg is en zowel nullen als enen bevat moet het programma een kopie van deze string maken, en termineren in de ja-toestand. Na afloop moet de tape in dit geval twee exemplaren van x bevatten, en wel in de volgende vorm: $x\#x$ (en verder blanco). Het karakter # wordt als scheidingsteken gebruikt. In alle gevallen staat de lees- en schrijfkop na afloop op de meest linker positie van de uitvoerstring. Indien nodig mag de invoerstring tijdens de werking van het programma tijdelijk veranderd/overschreven worden.
Geef de transitiefunctie in de vorm van een tabel en beschrijf de werking van je DTM-programma (kort) in woorden door aan te geven waar elke toestand globaal voor dient.
Voorbeeld: voor invoerstring 1111 (nee-instantie) en 01011011 (ja-instantie) staan na afloop de volgende strings op de tape: 1111, respectievelijk 01011011#01011011.

Een logische formule ϕ staat in conjunctieve normaalvorm (CNF) als ϕ een AND van clausules is, waarbij elke clausule de OR van een willekeurig aantal literals is. Een *literal* is een x_i of een $\neg x_i$, met x_i een logische variabele.

We bekijken de volgende twee beslissingsproblemen:

SAT: Gegeven een logische formule ϕ in CNF. *Vraag*: bestaat er een waardering (waarheidstoekenning) van de in ϕ voorkomende variabelen x_i, zodanig dat ϕ wordt waargemaakt, dat wil zeggen dat per clausule ten minste één literal waar is?

$\text{SAT}_{\geq 2}$: Gegeven een logische formule ϕ in CNF. *Vraag:* bestaan er *minstens twee verschillende* waarderingen van de in ϕ voorkomende variabelen x_i die ϕ waarmaken?

Voorbeeld: $\phi_0 = (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \neg x_3)$ heeft drie logische variabelen, namelijk x_1, x_2 en x_3. Deze ϕ_0 is een ja-instantie voor $\text{SAT}_{\geq 2}$, want de twee waarderingen $\langle \texttt{false}, \texttt{false}, \texttt{false} \rangle$ en $\langle \texttt{false}, \texttt{true}, \texttt{true} \rangle$ zijn verschillend en maken allebei ϕ_0 waar.

b. (10 punten)
Toon aan dat $\text{SAT}_{\geq 2} \in \mathcal{NP}$ door een *niet-deterministisch polynomiaal* algoritme voor $\text{SAT}_{\geq 2}$ te geven. Leg ook uit waarom dit algoritme precies ja-instanties met ja-instanties verbindt voldoet en waarom het polynomiaal is.

We bekijken de volgende eenvoudige transformatie T, die instanties van SAT transformeert naar instanties van $\text{SAT}_{\geq 2}$. Gegeven is een logische expressie ϕ in conjunctieve normaalvorm, met variabelen $x_1, x_2, \ldots, x_n$. We construeren hieruit een logische formule $T(\phi)\phi \wedge (z \vee \neg z)$, waarin naast de x_i's tevens een frisse nieuwe logische variabele z wordt gebruikt.
Voorbeeld: $T(\phi_0) = (\neg x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \neg x_3) \wedge (z \vee \neg z)$

c. (5 punten)
Laat zien dat de constructie van ψ uit ϕ in $\text{O}(|\phi|^k)$ stappen kan (voor zekere $k \geq 0$).

d. (10 punten)
Toon aan dat geldt: ϕ is een ja-instantie van SAT $\iff$ $T(\phi)$ is een ja-instantie van $\text{SAT}_{\geq 2}$.

Uit dit alles volgt dat $\text{SAT} \leq_{\text{P}} \text{SAT}_{\geq 2}$.

e. (5 punten)
Waarom geldt ook dat $\text{SAT}_{\geq 2} \leq_{\text{P}} \text{SAT}$? Wat gebruik je om dit te bewijzen?

f. (5 punten)
Bewijs dat $\text{SAT}_{\geq 2} \in \mathcal{NPC}$. Geef daarbij duidelijk aan welke van de voorgaande onderdelen je gebruikt.

g. (5 punten)
Een deelverzameling V' van de knopen in een graaf heet een *kliek* als tussen elk tweetal knopen uit V' een tak zit We bekijken het volgende beslissingsprobleem:

3Kliek: Gegeven een ongerichte graaf $\mathcal{G} = (V, E)$. Heeft deze graaf een kliek bestaande uit 3 knopen?

Toon aan dat 3Kliek $\in \mathcal{P}$ door in woorden een polynomiaal algoritme te geven dat het probleem oplost. Leg daarbij uit waarom je algoritme polynomiaal is.

h. (5 punten)
Stel dat we weten dat $\text{SAT}_{\geq 2}$ NP-volledig is. Beantwoord nu de volgende twee vragen:
(i) Bestaat er een polynomiale reductie van 3Kliek naar $\text{SAT}_{\geq 2}$?
(ii) Bestaat er een polynomiale reductie van $\text{SAT}_{\geq 2}$ naar 3Kliek?
Er wordt alleen naar het bestaan gevraagd. Je hoeft dus geen reductie te geven.