

Complexiteit

Uitwerkingen Opgaven

Opgave 59a

SAT: gegeven een logische formule ϕ in CNF. Bestaat er een waardering van de in ϕ voorkomende logische variabelen die ϕ waarmaakt?

Om te bewijzen dat $\text{SAT} \in \mathcal{NP}$ is het voldoende om een polynomiaal begreind niet-deterministisch algoritme A te geven voor SAT. De invoer voor A is een logische formule ϕ . Merk op dat n , het aantal verschillende in ϕ voorkomende logische variabelen $x_1, x_2, \dots, x_n$, in een polynomiaal aantal stappen kan worden bepaald. Immers (bijvoorbeeld): loop van links naar rechts door ϕ heen en voor elke logische variabele die je zo tegenkomt kijk je of er rechts van je in ϕ nog een exemplaar van staat. Zo ja dan tel je hem niet mee, zo nee, dan wel tellen. Zo tel je het meest rechter voorkomen van elke variabele in de formule, en daarmee krijg je precies het aantal verschillende logische variabelen in $O(|\phi|^2)$ stappen. Aangezien dit dus polynomiaal kan, heeft dit geen invloed op de polynomialiteit van Fase 2, en kunnen we ook wel (zonder beperking der algemeenheid) aannemen dat we weten dat er n verschillende logische variabelen in ϕ staan. Voor het gemak veronderstellen we dus dat n bekend is.

Intuïtief: een certificaat (waarmee je eenvoudig kunt laten zien dat een ja-instantie inderdaad een ja-instantie is) zal hier een waardering zijn die de invoerformule ϕ waarmaakt. Immers, precies het bestaan van zo'n waardering maakt ϕ tot een ja-instantie.

1. Fase 1 (gokfase)

Er wordt een string s gegenereerd, hierna te interpreteren als een rij booleans (waardering van de gebruikte logische variabelen).

// Hopelijk stelt dit een waarmakende waardering voor

2. Fase 2 (verificatiefase)

Er wordt gecontroleerd of s een waarmakende waardering voorstelt (interpreteer de i -de boolean als de waarheidswaarde voor de i -de logische variabele):

(1) controleer dat er precies n logische waarden staan: s aflopen en tellen (en meteen checken dat elke s_i een boolean T of F voorstelt): $O(|s|)$

(2) controleer dat de waardering voorgesteld door s een waarmakende waardering is voor ϕ . Hiertoe ϕ aflopen, clause voor clause. Voor elke clause kijken of een der voorkomende literalen (literals) volgens s true is. Daarvoor de waardering van elke literaal ophalen uit s . Dat kan samen zeker in $O(|\phi| \cdot |s|)$ stappen.

Als beide controles positief zijn retourneert het verificatie-algoritme True, anders wordt False geretourneerd of gaat het programma in een oneindige loop of blijft hangen, of ...

3. Fase 3 (uitvoerfase)

Als fase 2 True oplevert wordt “ja” uitgevoerd, anders geen uitvoer.

Er geldt: als ϕ een ja-instantie is dan bestaat er een waardering die ϕ waarmaakt, dus dan bestaat er een string s (het certificaat) die een waarmakende waardering voorstelt, hetgeen in Fase 2 nu juist gecontroleerd wordt. Ergo, dan *is er een* executie van A die

“ja” als uitvoer geeft (namelijk die waarin s correspondeert met een waarmakende waardering). Voor nee-instanties bestaat er geen waarmakende waardering en is er dus geen “goede” (= door Fase 2 goedgekeurde) string. Derhalve levert geen enkele executie van A daarop een uitvoer.

Hieruit volgt: het antwoord van A voor invoer x (dat is hier dus ϕ) is “ja” $\iff x$ is een ja-instantie voor SAT (en het antwoord van A voor x is “nee” $\iff x$ is een nee-instantie). Het algoritme A geeft derhalve voor elke invoer x het juiste antwoord.

En verder: A is polynomiaal begrensd.

Namelijk: als x ($= \phi$) een ja-instantie is, en s een goede string, dan bestaat s uit n waarheidswaarden, dus $|s| \in O(|x|)$, en de verificatiefase is dan $O(|x|^2)$, polynomiaal in $|x|$. Dus voor elke invoer x waarvoor het antwoord van A “ja” is, is er een executie die dat in $O(|x|)$ (Fase 1) + $O(|x|^2)$ (Fase 2) $\subseteq O(|x|^2)$ stappen doet.