Theoretische Informatica II

Examen van 28 juni 2002

Vraag 1

Definieer het begrip NP-compleetheid tot op het bot. Je mag enkel de volgende termen als bekend onderstellen:

- string
- verzameling
- functie
- algoritme
- algoritme loopt in tijd polynomiaal in ...
- input en output van een algoritme

Oplossing

Een taal is een verzameling strings.

Een taal A is NP-compleet als A in NP zit, en voor elke andere taal B in NP geldt dat $B \leqslant^{\mathrm{P}} A$.

Een taal zit in NP als ze een polynomiale verifier bezit.

Een verifier voor een taal A is een algoritme V dat als input een koppel strings (w, c) verwacht, dat als output 'true' of 'false' geeft, en wel zodanig dat voor eender welke string w geldt:

$$w \in A \quad \Leftrightarrow \quad \exists c : V(w, c) = \text{true}$$

We noemen V polynomiaal als V loopt in tijd polynomiaal in de lengte van w.

Taal B kan polynomiaal gereduceerd worden naar taal A, wat genoteerd wordt als $B \leqslant^{\mathrm{P}} A$, als er een functie f van strings naar strings bestaat die berekenbaar is in polynomiale tijd, zodat voor eender welke string w geldt:

$$w \in B \quad \Leftrightarrow \quad f(w) \in A$$

Vraag 2: de stelling van Cook

1. Wat is de intuïtieve betekenis van de variabelen $x_{i,j,s}$?
2. Wat is het bereik van de variabelen i and j?
3. Construeer de formule ϕ_{cell} en leg de betekenis van deze formule uit.

Oplossing

In het bewijs van de stelling van Cook vertrekken we van een willekeurige taal L in NP, meer bepaald, met een polynomiale verifier V voor die taal. We moeten een reductie construeren die een willekeurige gegeven string w omzet in een booleaanse formule ϕ zodat $w \in L \Leftrightarrow \phi$ is satisfiable. We weten van de vorige vraag dat $w \in L$ hetzelfde is als $\exists c : V(w, c) = \text{true}$.

We moeten dus de run van V op input (w, c) simuleren, waarbij w dus gegeven is, maar c onbekend. Hiervoor beschouwen we de computation table van V op input (w, c). Omdat V polynomiaal is, loopt V op input (w, c) voor ten hoogste n^k stappen, waar n de lengte van w is, en k een vast getal. De computation table kan daarom voorgesteld worden als een $n^k \times n^k$ matrix. In elke cel van de matrix staat een symbool, dat een letter van het alfabet kan zijn, of een toestand van V. De verzameling van deze symbolen noemen we C.

De variabelen $x_{i,j,s}$ stellen nu de inhoud van de matrix voor op volgende wijze: $x_{i,j,s}$ betekent "de cel in rij i, kolom j, bevat het symbool s". Het bereik van de indices i en j is dus $1, \ldots, n^k$.

De formule ϕ_{cell} kan je letterlijk in het boek terugvinden. De betekenis is: "in elke cel staat minstens 1 symbool ingevuld, en ook hoogstens 1 symbool, dus exact 1 symbool."

Vraag 3

Zij f_{clique} en $f_{\text{vertex cover}}$ de functies uit de cursus die 3SAT naar CLIQUE respectievelijk VERTEXCOVER reduceren, en zij f_{3SAT} de functie uit de cursus die SAT naar 3SAT reduceert. Zij ϕ_1 de formule $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_1 \vee x_3)$.

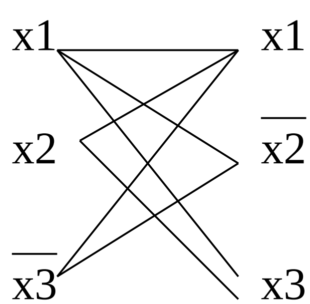
1. Construeer $f_{\text{clique}}(\phi_1)$.

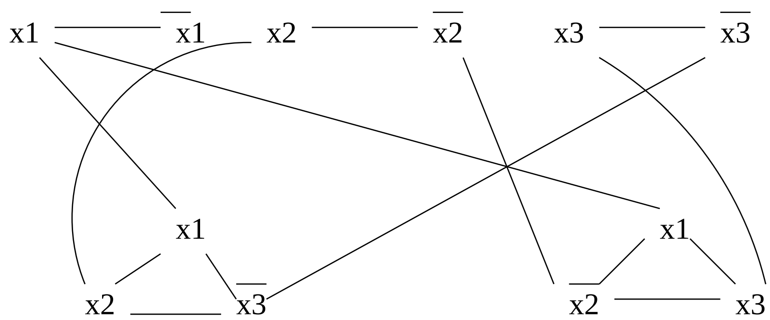
2. Construeer $f_{\text{vertex cover}}(\phi_1)$.

3. Zij ϕ_2 de formule $\neg(\neg(x_1 \vee x_2) \wedge x_3) \vee x_1$. Construeer $f_{\text{3SAT}}(\phi_2)$.

Oplossing

1.



2. 

3. De hulpformules zijn

$$\begin{array}{lcl}
\varepsilon_1 & \equiv & u_1 \leftrightarrow x_1 \\
\varepsilon_2 & \equiv & u_2 \leftrightarrow x_2 \\
\varepsilon_3 & \equiv & u_3 \leftrightarrow x_3 \\
\varepsilon_4 & \equiv & u_4 \leftrightarrow u_1 \vee u_2 \\
\varepsilon_5 & \equiv & u_5 \leftrightarrow \neg u_4 \\
\varepsilon_6 & \equiv & u_6 \leftrightarrow u_5 \wedge u_3 \\
\varepsilon_7 & \equiv & u_7 \leftrightarrow \neg u_9 \\
\varepsilon_8 & \equiv & u_8 \leftrightarrow u_7 \vee u_1
\end{array}$$

$f(\phi_2)$ is de formule $(\bigwedge_{i=1}^{8} \varepsilon_i) \wedge (u_8 \vee u_8 \vee u_8)$. Er rest nog uit te leggen hoe de formules ε_i moeten worden uitgewerkt. Dit gebeurt volgens het principe

$$u_j \leftrightarrow x \equiv (\neg u_j \vee x \vee x) \wedge (\neg x \vee u_j \vee u_j),$$

$$u_j \leftrightarrow \neg u_k \equiv (\neg u_j \vee \neg u_k \vee \neg u_k) \wedge (u_k \vee u_j \vee u_j),$$

$$u_j \leftrightarrow u_k \wedge u_l \equiv (\neg u_j \vee u_k \vee u_k) \wedge (\neg u_j \vee u_l \vee u_l) \wedge (\neg u_k \vee \neg u_l \vee u_j),$$

en

$$u_j \leftrightarrow u_k \vee u_l \equiv (\neg u_j \vee u_k \vee u_l) \wedge (\neg u_k \vee u_j \vee u_j) \wedge (\neg u_l \vee u_j \vee u_j).$$

Vraag 4

Defineer een zo klein mogelijke klasse van functies $\mathcal{F}$ zodat voor elke taal L in NP er geldt dat $L \in \bigcup_{f \in \mathcal{F}} \mathrm{TIME}(f(n))$. Beargumenteer waarom dit zo is.

Oplossing

Zij k de grootte van het alphabet. Zij L een taal in NP en zij V een polynomiale verifier die in tijd $t(n)$ werkt voor een polynoom $t(n)$. Een certificaat kan maximaal uit $t(n)$ symbolen bestaan. Het aantal mogelijke certificaten c is dan $k^{t(n)}$. Om na te gaan of een bepaalde input x tot L behoort moeten we dus elk certificaat beschouwen en voor elk certificaat het algoritme V uitvoeren. Dit kan allemaal in tijd $t(n)^2$. Aangezien deze truck voor elke taal in NP kan worden toegepast volstaat het als $\mathcal{F}$ de klasse van functies $\{k^{t(n)} \mid t(n) \text{ een polynoom}\}$ te nemen.

Vraag 5

Zij G een ongerichte graaf. Een verzameling knopen V van G is *onafhankelijk* als er tussen geen enkele twee knopen in V een boog is.

Toon de NP-compleetheid aan van volgend probleem:

INDEPENDENT SET = $\{\langle G, k\rangle \mid G$ heeft een onafhankelijke verzameling knopen van grootte $k\}$.

Oplossing

We construeren eerst een verifier V. Op input $\langle G, k, V\rangle$ doet V het volgende:

- Test of V een verzameling van k knopen van G is.
- Test of V onafhankelijk is.
- Als beide testen slagen dan aanvaardt V anders reject V.

Beide testen zijn efficiënt uit te voeren. De verifier is in P. We tonen vervolgens aan dat CLIQUE $\leqslant_P$ INDEPENDENT SET. De functie $f(\langle G, k\rangle) = (H, k)$ is als volgt gedefinieerd: H is het complement van G. D.w.z, H and G bestaan uit hetzelfde aantal knopen maar tussen twee knopen in H is er een boog a.s.a er geen boog is tussen deze knopen in G.

CORRECTHEID:$\langle G, k\rangle \in$ CLIQUE $\Leftrightarrow f(\langle H, k\rangle) \in$ INDEPENDENT SET.

$\Rightarrow$: Zij V een k-clique in G dan is V een onafhankelijke verzameling in H.

$\Leftarrow$: Zij V een onafhankelijke verzmaling met k elementen dan is V een k-clique in G.

Vraag 6

Zij G een ongerichte graaf. Een *kernel* is een verzameling knopen V van G zodat V onafhankelijk is (zie vorige vraag), én elke knoop buiten V verbonden is met tenminste één knoop in V.

Zij k een natuurlijk getal. Beschouw het volgend probleem

k-KERNEL = $\{\langle G\rangle \mid G$ heeft een kernel van grootte $k\}$.

Toon aan dat het probleem in P is ofwel dat het NP-compleet is.

Oplossing

Het probleem is in P aangezien k een vast getal is. Het volgende schematische algoritme toont dit aan:

```
for x1 in G do
  for x2 in G do
    for x3 in G do
    ...

      for xk in G do
         if {x1,...,xk} is een kernel then
                    accept and stop
      od
    ...
  od
od
reject
```

Testen of een verzameling een kernel is, kan efficiënt gedaan worden. Aangezien het algoritme verder uit een vast aantal (namelijk k) for-lussen bestaat is het in P.

Vraag 7

Toon de NP-compleetheid aan van volgend probleem:

NIETALLES = $\{\langle r \mid$

1. r is een reguliere expressie die een disjunctie is van expressies van de vorm $e_1 \cdots e_n$ waarbij elke e_i gelijk is aan 0, 1, of $(0+1)$; bijvoorbeeld, $01(0+1)1 + 0000 + (0+1)(0+1)00$;
2. r is niet equivalent met $\underbrace{(0+1)\cdots(0+1)}_{(n \text{ keer})}$.

$\}$.

HINT: Ga na hoe een string van nullen en enen een waarheidstoekenning kan encoderen. Wat drukt $\underbrace{(0+1)\cdots(0+1)}_{(n \text{ keer})}$ dan uit?

Oplossing

We construeren eerst een verifier V. Op input $\langle r, c\rangle$ doet V het volgende:

- Test of c een string is over het alpfabet $\{0,1\}$ van lengte n.
- Test of $c \notin L(r)$.
- Als beide testen slagen dan aanvaardt V anders reject V.

Deze tests kunnen efficiënt gedaan worden. De verifier is in P.

We tonen vervolgens aan dat 3SAT $\leqslant_P$ NIETALLES. Eerst observeren we dat een string $w = w_1 \cdots w_n \in \{0,1\}^n$ een waarheidstoekenning voor de variabelen $x_1, \ldots, x_n$ uitdrukt. Inderdaad, voor elke i, x_i is waar a.s.a. $w_i = 1$. Dus, elke regulier expressie r drukt een verzameling van waarheidstoekenningen uit. In het bijzonder moeten we nagaan of er een waarheidstoekenning/string is die *niet* in r zit. Zij ϕ een formule in 3CNF over de variabelen $x_1, \ldots, x_n$. M.a.w., ϕ is van de vorm $\bigwedge_{i=1}^m C_i$. Welke waarheidstoekenningen maken deze formule onwaar: elke toekenning die minstens één clause C_i onwaar maakt. Een clause is onwaar als elke literal (x_i of $\neg x_i$) die er in voorkomt onwaar is. We definiëren nu $f(\phi)$ als de reguliere expressie die alle waarheidstoekenningen definieert die ϕ onwaar maken. Dus, $f(\phi)$ definieert niet alles a.s.a ϕ satisfiable is.

We illustreren de functie f met een voorbeeld. Zij ϕ gelijk aan

$$(x_1 \vee x_2 \vee \neg x_4) \wedge (x_1 \vee \neg x_3 \vee x_4).$$

Dan is $f(\phi)$ gelijk aan

$$00(0+1)1 + 0(0+1)10.$$

CORRECTHEID: $\langle\phi\rangle \in$ 3SAT $\Leftrightarrow f(\langle\phi\rangle) \in$ NIETALLES.

$\Rightarrow$: Zij $w \in \{0,1\}^n$ de waarheidstoekenning die ϕ waar maakt, dan $w \notin L(f(\langle\phi\rangle))$ en $f(\langle\phi\rangle) \in$ NIETALLES.

$\Leftarrow$: Zij $w \in \{0,1\}^n$ en $w \notin L(f(\langle\phi\rangle))$ dan is w de waarheidstoekenning die ϕ waar maakt.