

Opgave 1

a) Een *niet-deterministische Turing-machine* is een 4-tuple (Q, Σ, δ, q_0) met

- Q de verzameling toestanden,
- Σ het alfabet,
- $\delta \subseteq Q \times (\Sigma \cup \{\#\}) \times (Q \cup \{q_{\text{YES}}, q_{\text{NO}}, q_{\text{HALT}}\})$,
- $q_0 \in Q$

met $\#$ het blank-symbool niet in Σ en $q_{\text{YES}}, q_{\text{NO}}, q_{\text{HALT}}$ nieuwe toestanden niet in Q .

Een *counter program* is een lijst van mogelijk gelabelde instructies van de vorm

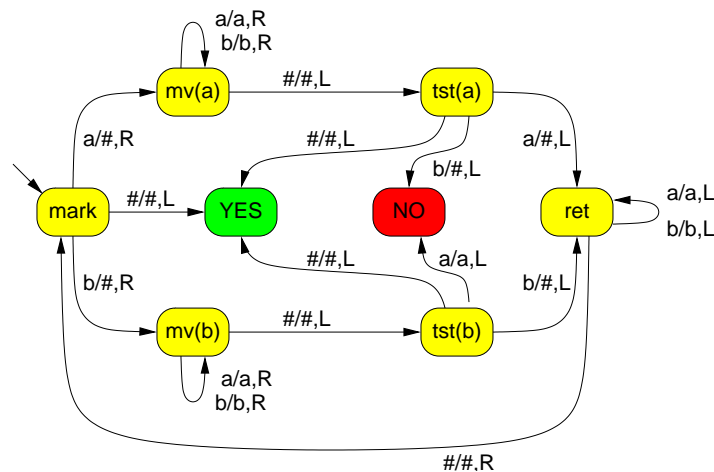
- $X \leftarrow 0$
- $X \leftarrow Y + 1$
- $X \leftarrow Y - 1$
- if** X **goto** L .

Een counter program heeft eindig veel variabelen die niet-negatieve, geheeltallige waarden kunnen krijgen.

b) Het volgende counter program slaat de som van de variabelen X en Y op in de variabele Z :

- $U \leftarrow 0$
- L: **if** $X=0$ **goto** M
- $X \leftarrow X-1$
- $Y \leftarrow Y+1$
- M: **if** $U=0$ **goto** L
- $Z \leftarrow Y+1$
- $Z \leftarrow Z-1$

c) De gevraagde Turing-machine heeft de volgende STD:



Opgave 2

a) Het woordcorrespondentieprobleem (wcp) over een alfabet Σ vraagt of een n -tal paren

$$(x_1, y_1), \dots, (x_n, y_n)$$

van woorden over Σ het mogelijk is een aantal paren één of meerdere keren aan te wijzen zodat de woorden die in de eerste en tweede coördinaat ontstaan gelijk zijn, i.e.

$$\exists k \exists i_1, i_2, \dots, i_k : x_{i_1} x_{i_2} \dots x_{i_k} = y_{i_1} y_{i_2} \dots y_{i_k}$$

- b) Kies een alfabet Σ en een n -tal paren $P = (x_1, y_1), \dots, (x_n, y_n)$ van woorden over Σ . Construeer nu een grammatica G als volgt:

Kies nieuwe variabelen S, S_X, S_Y .

Kies n nieuwe terminalen a_1, a_2, \dots, a_n .

Definieer producties

$$\begin{aligned} S &\rightarrow S_X \mid S_Y \\ S_X &\rightarrow x_i S_X a_i \mid x_i a_i \quad 1 \leq i \leq n \\ S_Y &\rightarrow y_i S_Y a_i \mid y_i a_i \quad 1 \leq i \leq n \end{aligned}$$

Bewering: G is ambigu precies als het wcp voor Σ en P en oplossing heeft.

(\Rightarrow) Stel G is ambigu. Dus G produceert een woord w op twee essentieel verschillende manieren. Wegens de vorm van G hebben we dan

$$S \rightarrow S_X \rightarrow^* w \text{ and } S \rightarrow S_Y \rightarrow^* w.$$

Dus we hebben

$$w = x_{i_1} x_{i_2} \cdots x_{i_k} \cdot a_{i_k} \cdots a_{i_2} a_{i_1} \text{ and } w = y_{j_1} y_{j_2} \cdots y_{j_\ell} \cdot a_{i_\ell} \cdots a_{i_2} a_{i_1}$$

voor geschikte indices $i_1, i_2, \dots, i_k, j_1, j_2, \dots, j_\ell$. Omdat de a 's vers zijn geldt $k = \ell$ en $i_h = j_h$ voor $1 \leq h \leq k, \ell$. Hieruit volgt $x_{i_1} x_{i_2} \cdots x_{i_k} = y_{j_1} y_{j_2} \cdots y_{j_\ell} = y_{i_1} y_{i_2} \cdots y_{i_k}$. Dus, het wcp heeft een oplossing.

(\Leftarrow) Stel het wcp heeft een oplossing. Kies dan i_1, i_2, \dots, i_k zodat

$$x_{i_1} x_{i_2} \cdots x_{i_k} = y_{i_1} y_{i_2} \cdots y_{i_k}.$$

Dan geldt

$$\begin{aligned} S &\rightarrow S_X \rightarrow x_{i_1} S_X a_{i_1} \rightarrow x_{i_1} x_{i_2} S_X a_{i_2} a_{i_1} \rightarrow \dots \rightarrow x_{i_1} x_{i_2} \cdots x_{i_k} S_X a_{i_k} \cdots a_{i_2} a_{i_1} \\ S &\rightarrow S_Y \rightarrow y_{i_1} S_X a_{i_1} \rightarrow y_{i_1} y_{i_2} S_X a_{i_2} a_{i_1} \rightarrow \dots \rightarrow y_{i_1} y_{i_2} \cdots y_{i_k} S_X a_{i_k} \cdots a_{i_2} a_{i_1}. \end{aligned}$$

Dus G is ambigu. Dit bewijst de bewering.

Stel dat ambigüiteit van een context-vrije grammatica beslist kan worden. Dan kan het wcp ook beslist worden door een instantie van wcp om te zetten in een context-vrije grammatica als boven en daarvan de ambigüiteit te beslissen.

Opgave 3

- a) Een beslissingsprobleem C is NP-volledig als
- Er een polynomiaal algoritme is dat beslist of een instantie van het probleem C wel of niet een oplossing heeft.
 - Voor elk ander probleem in NP een polynomiale reductie naar C bestaat.
- b) SUBSET-SUM is in NP want de deelverzameling is een polynomiaal certificaat. Laat $\varphi = \bigwedge_{1 \leq i \leq n} a_i \vee b_i \vee c_i$ een 3CNF-formule in de variabelen x_1, \dots, x_m zijn. Construeer dan de verzameling getallen S en grens t als volgt: $S = \{ y_j, z_j \mid 1 \leq j \leq m \} \cup \{ g_i, h_i \mid 1 \leq i \leq n \}$ waarbij

$$\begin{aligned} y_j &= 2^{j+n-1} + \sum_{i=1}^n \delta_{ij} \cdot 2^i \\ z_j &= 2^{j+n-1} + \sum_{i=1}^n (1 - \delta_{ij}) \cdot 2^i \\ g_i &= 2^{i-1} \\ h_i &= 2^{i-1} \end{aligned}$$

waarbij $1 \leq i \leq n, 1 \leq j \leq m$ en $\delta_{ij} = 1$ als x_j positief voorkomt in $a_i \vee b_i \vee c_i$ en $\delta_{ij} = 0$ anders. Zet

$$t = \sum_{j=1}^m 2^{j+n-1} + 3 \cdot \sum_{i=1}^n 2^{i-1}.$$

Deze constructie is polynomiaal. Bewering: φ is vervulbaar precies als S een deelverzameling van omvang t heeft.

(\Rightarrow) Zij α een bedeling die φ waar maakt. Definiëer de deelverzameling $S' \subseteq S$ als volgt

$y_j \in S'$ als $\alpha(x_j)$, $z_j \in S'$ als $\neg\alpha(x_j)$ voor $1 \leq j \leq m$;

$g_i \in S'$ als α in $a_i \vee b_i \vee c_i$ niet alle literals waarmaakt ($1 \leq i \leq n$);

$h_i \in S'$ als α in $a_i \vee b_i \vee c_i$ slechts één literal waarmaakt ($1 \leq i \leq n$).

Omdat voor elke clause tenminste één literal door α vervuld wordt, zal het $(i - 1)$ -ste cijfer van t precies 3 worden met bovenstaande keuze. Dus $\Sigma S' = t$

(\Leftarrow) Wegens de vorm van t, y_j, z_j geldt dat precies één van y_j dan wel z_j in S' zit. Voor deze keuze van y_j of z_j geldt wegens constructie dat $\delta_{ij} = 1$ als $y_j \in S'$ dan wel $\delta_{ij} = 0$ als $z_j \in S'$. Laat α de bedeling zijn met $\alpha(x_j) = \text{true} \iff y_j \in S'$. Deze bedeling vervult φ .

Opgave 4

a) Het benaderingsalgoritme `next_fit` heeft de volgende pseudo-code:

```

k=1
for i=1 to n do
  if s_i fits in bin_k then
    add s_i to bin_k
  else
    close bin_k
    k=k+1
    add s_i to bin_k
  end
end
return k

```

b) Stel het aantal bins opgegeven door `next_fit` is $nf(S)$ en het minimaal aantal bins is $opt(S)$. Laat vervolgens de bins achtereenvolgend $c_1, c_2, \dots, c_{nf(S)}$ door `next_fit` gevuld worden. Bewering: $nf(S) \leq 2opt(S) - 1$. Stel niet, dus neem aan $nf(S) \geq 2opt(S)$. Dan geldt

$$\begin{aligned}
& \sum_{i=1}^n s_i \\
&= \sum_{i=1}^{nf(S)} c_i \\
&\geq \sum_{i=1}^{opt(S)} c_i \\
&= \sum_{i=1}^{opt(S)} c_{2i-1} + c_{2i} \\
&< opt(S) \cdot B
\end{aligned}$$

Tegenspraak, want er geldt $\sum_{i=1}^n s_i \leq opt(S) \cdot B$. Conclusie: $nf(S) \leq 2 \cdot opt(S) - 1$ en

$$opt(S) \leq nf(S) \leq 2 \cdot opt(S).$$