

IN2505 II Berekenbaarheidstheorie
Proeftentamen

BELANGRIJK

Beschikbare tijd 3 uur.

Gebruik hulpmiddelen Bij het tentamen mogen geen boeken, aantekeningen, of andere bronnen worden geraadpleegd. Tevens mag er geen gebruik worden gemaakt van rekenmachines.

Puntentelling Dit tentamen bestaat uit 5 opgaven. Voor elke opgave zijn 10 punten te verdienen. Alle opgaven tellen even zwaar mee.

Algemeen Het de bedoeling om iedere opgave te beginnen op een nieuwe zijde.

1. Zij $L_1, L_2 \subseteq \Sigma^*$ talen over een alfabet Σ . Laat de *concatenatie* L_1L_2 van L_1 en L_2 gedefinieerd zijn als $L_1L_2 = \{w_1w_2 \mid w_1 \in L_1 \text{ en } w_2 \in L_2\}$.
 - (a) Geef een definitie van het begrip *beslisbare taal* (Eng.: decidable language). (2 punten)
 - (b) Bewijs dat voor elk paar *beslisbare* talen $L_1, L_2 \subseteq \Sigma^*$, de concatenatie L_1L_2 ook *beslisbaar* is. (8 punten)

Uitwerking

- (a) Een taal L is beslisbaar d.e.s.d.a. er Turing-machine M bestaat zodanig dat $L(M) = L$ en zodanig dat M voor elke invoer in een accepterende of verwerpende toestand komt.
- (b) Stel dat L_1 en L_2 beslisbare talen zijn en dat M_1 en M_2 respectievelijk beslissers zijn van L_1 en L_2 . Om te laten zien dat L_1L_2 beslisbaar is, construeren we een niet-deterministische TM M die L_1L_2 beslist. Het idee is dat M een invoerwoord w op niet deterministische wijze opsplijt in delen w_1 en w_2 zodanig dat $w = w_1w_2$. Vervolgens laten we M_1 op w_1 werken en M_2 op w_2 . Als beide beslissers accepteren, laten we M ook accepteren. In alle andere gevallen laten we w verwerpen.

Een high-level beschrijving van M is als volgt:

M = “Op invoerwoord w :

1. Als het gescande symbool geen spatie is: ga naar stap 2 òf lees het volgende symbool van de invoer en herhaal stap 1.
Als het gescande symbool een spatie is: ga naar stap 2.
2. Markeer het woord links van de lees-schrijfkop als w_1 en de rest van w als w_2 .
3. Draai M_1 op w_1 , en M_2 op w_2 .
4. *Accepteer* als zowel M_1 als M_2 accepteren, *verwerp* in alle andere gevallen.”

Dan geldt dat M een NDTM is die L_1L_2 beslist:

- M is zo geconstrueerd dat zij in principe ‘goed’ kan gokken, waarbij een goede gok de juiste w_1 en w_2 oplevert.
- Stap 1 kan slechts een eindig aantal keren worden herhaald vanwege de eindige lengte van invoer w .
- Zowel M_1 als M_2 is een beslisser, zodat de stappen 3 en 4 beide zullen eindigen.

2. Zij V de verzameling van alle functies $f : \mathbb{N} \rightarrow \mathbb{N}$ zodanig dat $f(x) \leq x$ voor alle x . Met andere woorden:

$$V = \{f : \mathbb{N} \rightarrow \mathbb{N} \mid f(x) \leq x \text{ voor alle } x\}.$$

- (a) Geef een definitie van het begrip *aftelbare verzameling* (Eng.: countable set). (2 punten)
- (b) Ga na of V aftelbaar is of niet (2 punten) en geef een bewijs van uw opvatting (indien de verzameling niet aftelbaar is, dient u een bewijs te geven met behulp van de diagonaalmethode). (6 punten)

Uitwerking

- (a) Een verzameling V is aftelbaar als V eindig is, of als er een bijjectie (correspondentie) bestaat tussen V en $\mathbb{N}$.
- (b) V is uiteraard niet aftelbaar.

Bewijs:

Stel dat V aftelbaar is en dat $f_0, f_1, \dots, f_n, \dots$ een aftelling van V is. Voor al deze functies moet gelden $f_n(0) = 0$ vanwege de definitie van V . Dit betekent dat een recht-toe-recht-aan diagonaalfunctie $g(x) = f_x(x) + 1$ niet werkt, omdat dan zou gelden $g(0) = 1$, waarmee g niet tot V zou behoren. Om deze reden ‘schuiven’ we de diagonaal een plaats op naar rechts:

$$g(x) = \begin{cases} 0 & \text{als } x = 0 \\ f_{x-1}(x) + 1 \bmod 2 & \text{anders} \end{cases}$$

Dan geldt enerzijds dat $g \in V$ (immers $g(0) = 0 \leq 0$ en $g(x) \leq 1$ voor alle $x \geq 1$) en anderzijds dat $g \neq f_n$ voor alle $n \in \mathbb{N}$ (door de diagonaalconstructie). Tegenspraak.

Uit deze tegenspraak volgt dat V niet aftelbaar is.

3. Beschouw de klasse van Deterministische Eindige Automaten (DFA's) over een gegeven alfabet Σ . Zij:

$$ALL_{\text{DFA}} = \{\langle M \rangle \mid M \text{ is een DFA en } L(M) = \Sigma^*\}.$$

Bewijs dat ALL_{DFA} *beslisbaar* (Eng.: decidable) is.

Uitwerking Beschouw voor een eindige automaat M de eindige automaat $\overline{M}$ die identiek is aan M behalve dat in $\overline{M}$ niet-accepterende toestanden van M acceptierend zijn en vice versa. Dan geldt $L(\overline{M}) = \overline{L(M)}$. Dit betekent dat als $L(M) = \Sigma^*$, dan $L(\overline{M}) = \emptyset$, ofwel:

$$\langle M \rangle \in ALL_{\text{DFA}} \iff \langle \overline{M} \rangle \in E_{\text{DFA}}.$$

Volgens Theorem 4.4 uit Sipser is het probleem E_{DFA} beslisbaar. Aangezien de afbeelding van M op $\overline{M}$ berekenbaar is, moet gelden dat ALL_{DFA} ook beslisbaar is.

4. Beschouw voor een gegeven alfabet Σ het volgende probleem:

$$D = \{\langle M \rangle \mid M \text{ is een TM en } L(M) \text{ bevat precies één woord}\}$$

Met andere woorden, D bestaat uit die (coderingen van) Turing-machines M die precies één woord accepteren.

Bewijs met behulp van *mapping reduction* dat D niet beslisbaar (Eng.: *undecidable*) is. Uw bewijs dient daarbij uit 2 delen te bestaan:

- (a) Een beschrijving van de *reductie(functie)* f . (5 punten)
- (b) Het bewijs dat deze f inderdaad aan de eisen van een mapping reduction voldoet. (5 punten)

Uitwerking

- (a) We voeren de volgende reductie uit: $HALT_{\text{TM}} \leq_m D$.

Zij M_1 een Turing-machine zodanig dat $L(M_1) = \{machine\}$, bijvoorbeeld:

M_1 = “Op invoerwoord x :

1. *Accepteer* als $x = machine$, anders *verwerp*.”

Dan geldt $\langle M_1 \rangle \in D$.

Zij $\langle M, w \rangle \in HALT_{\text{TM}}$. Definieer de volgende TM M' als volgt:

M' = “Op invoerwoord x :

1. Draai M op w .
2. Draai M_1 op x .
3. *Accepteer* als M_1 accepteert, *verwerp* als M_1 verwerpt.”

Definieer de reductie f van $HALT_{\text{TM}}$ naar D als volgt: $f(\langle M, w \rangle) = \langle M' \rangle$.

- (b) De hierboven gedefinieerde f is inderdaad een reductie. Allereerst is f berekenbaar. De volgende TM F berekent f :

F = “Op invoerwoord $\langle M, w \rangle$:

1. Construeer M' m.b.v. $\langle M, w \rangle$.
2. Print $\langle M' \rangle$ op tape en stop.”

Verder geldt:

Als $\langle M, w \rangle \in \text{HALT}_{\text{TM}}$ dan stopt M op invoer w . Hierdoor zal M' aan stap 2 toekomen en zich verder gedragen als M_1 zodat $L(M') = L(M_1) = \{\text{machine}\}$. Maar dit impliceert dat $\langle M' \rangle \in D$.

Als $\langle M, w \rangle \notin \text{HALT}_{\text{TM}}$, dan stopt M niet op invoer w en zal M' niet aan stap 2 toekomen. Er geldt dan $L(M') = \emptyset$, zodat $\langle M' \rangle \notin D$.

Hiermee is aangetoond:

$$\langle M, w \rangle \in \text{HALT}_{\text{TM}} \Leftrightarrow f(\langle M, w \rangle) = \langle M' \rangle \in D.$$

Uit bovenstaande volgt dat f een reductie is.

5. Zij P de klasse van alle Turing-machines met het zelfde tape-alfabet $\Gamma = \{0, 1, \sqcup\}$ die wanneer gestart op een lege tape stoppen. Beschouw de functie $BB : \mathbb{N} \rightarrow \mathbb{N}$ die gedefinieerd is door:

$BB(n)$ = het maximum aantal 1-en dat door een TM uit P met n toestanden kan worden geprint.

Bijvoorbeeld: $BB(3)$ is dus gedefinieerd als het maximum van alle aantallen 1-en die door Turing-machines met 3 toestanden uit P kunnen worden geprint, indien deze starten op een lege tape en stoppen in de accepterende toestand.

Laat zien dat de functie BB niet *berekenbaar* (Eng: computable) is.

Uitwerking Zij B een TM die BB berekent. Definieer nu de volgende TM F :

F = “Op iedere invoer:

1. Verwerf, m.b.v. recursiestelling, eigen beschrijving $\langle F \rangle$.
2. Bepaal het aantal toestanden n van F .
3. Bereken $m = BB(n)$ m.b.v. B .
4. Print $m + 1$ keer een “1” op de tape en stop.”

Dit leidt als volgt tot een tegenspraak:

F is een TM die n toestanden bezit (hetgeen door F zelf wordt vastgesteld met behulp van de recursiestelling). Volgens TM B die de busy-beaverfunctie BB berekent, zou F niet meer dan $BB(n)$ “1”-en op de tape kunnen printen alvorens te stoppen. In werkelijkheid print F er echter $BB(n) + 1$. Dit kan niet.

EINDE TENTAMEN