

HERTENTAMEN COMPUTABILITY

Dinsdag 17 juni 2025, 09.00 - 12.00 uur

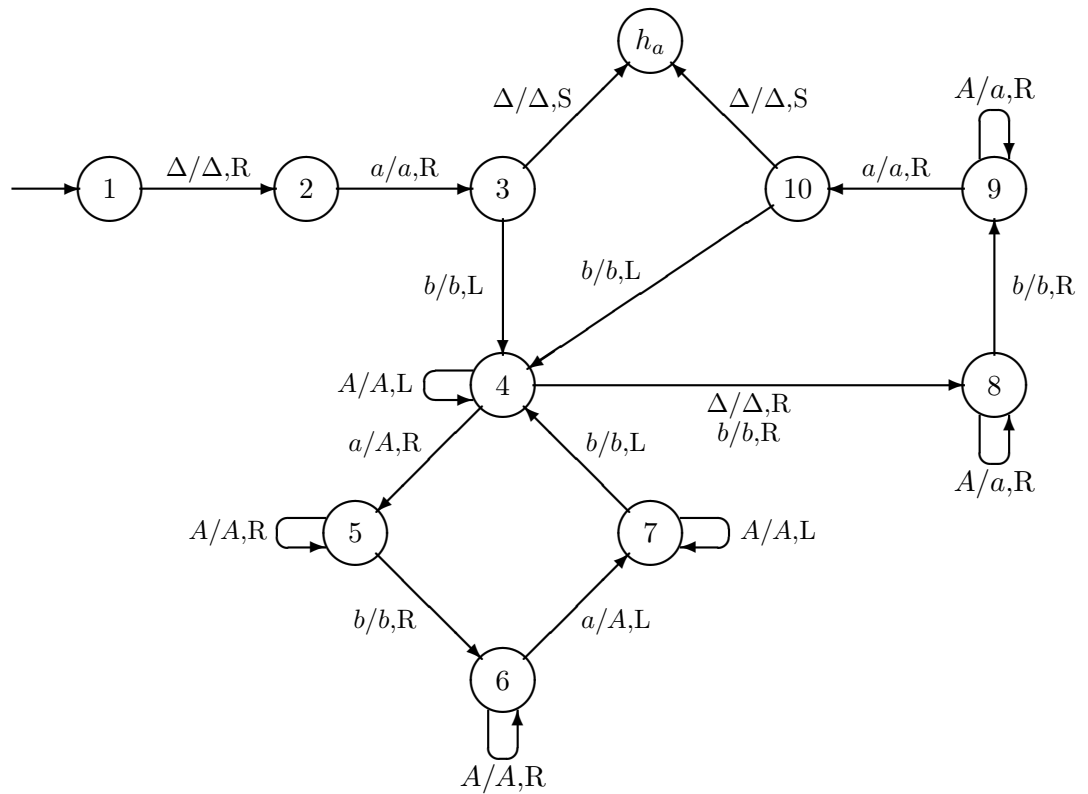
Dit tentamen bestaat uit zes opgaven, waarbij steeds tussen [en] staat hoeveel punten er ongeveer mee te verdienen zijn. In totaal zijn er 100 punten te verdienen. Wanneer er bij een vraag om uitleg, motivatie of toelichting gevraagd wordt, is het belangrijk om die ook te geven.

1. [20 pt] Laat

$$L_1 = \{a, abaa, abaabaaa, abaabaaaabaaa, \dots\}$$

Ofwel, een string $x \in L_1$ bestaat uit één of meerdere reeksen a 's gescheiden door steeds een b , waarbij de eerste reeks maar één a kent, en elke volgende reeks a 's één a langer is. Je kunt ook zeggen dat een string $x \in L_1$ van de vorm $a^1ba^2ba^3b \dots a^k$ is, met $k \geq 1$.

Bij het vorige tentamen vroegen we om een *unrestricted grammar* G te geven, zó dat $L(G) = L_1$. Beschouw nu de volgende Turingmachine T_1 , waarvoor gegeven is dat $L(T_1) = L_1$:



Een (te) korte toelichting bij deze Turingmachine: T_1 werkt door eerst te controleren of de invoer met a begint. Vervolgens, zo lang een reeks a 's gevolgd wordt door een b , wordt (in toestanden 4,5,6,7) voor elke a links van de b , een a rechts van de b gezocht, en beide worden gemarkeerd als hoofdletter. Daarna wordt (in toestanden 8,9,10) gecontroleerd of er rechts nog één extra a staat.

Strings over $\{a, b\}$ die niet in L_1 zitten, worden impliciet verworpen.

- (a) Een configuratie van een Turingmachine $T = (Q, \Sigma, \Gamma, q_0, \delta)$ is een string van de vorm xqy , met $x, y \in (\Gamma \cup \{\Delta\})^*$ en $q \in Q$, waarbij x de inhoud van de tape links van de leeskop beschrijft, y de inhoud van de tape onder en rechts van de leeskop beschrijft, en q de toestand is waarin T zich bevindt.

Een berekening van een Turingmachine T voor invoer x begint bij de initiële configuratie van T voor x , en volgt dan herhaaldelijk de (of een) mogelijke transitie in T , die een volgende configuratie oplevert. Ze voert ons dus van configuratie naar configuratie, enzovoort.

Geef de eerste zes stappen van de berekening van Turingmachine T_1 hierboven voor invoer $x = abaa$. Dit betekent dat je antwoord zeven configuraties moet bevatten.

- (b) Nu moet je uitzoeken waar en wanneer T_1 welke invoer impliciet verworpt. Concreet: Geef alle combinaties van een toestand en een tapesymbool (ook Δ tellen we hier mee) waar Turingmachine T_1 een bepaalde invoer $x \in \{a, b\}^*$ die niet in L_1 zit, impliciet verworpt. Geef bij elke combinatie van toestand en tapesymbool die je noemt, ook een invoer x die T_1 bij die combinatie zal verwerpen.

Bijvoorbeeld: In toestand $\dots$ zal T_1 invoer $x = \dots$ verwerpen, omdat hij op enig moment in die toestand het symbool $\dots$ op de tape ziet staan. (Natuurlijk hoeft je niet iedere keer deze hele zin op te schrijven.)

2. [18 pt] Laat

$$L = \{x \in \{a, b, c\}^* \mid n_a(x) < n_b(x) \text{ en } n_a(x) < n_c(x)\}$$

Dus bijvoorbeeld $cb, cabbc, aabbbcccc \in L$, maar $cbac \notin L$ omdat $n_a(cbac) = n_b(cbac)$.

Bij het vorige tentamen vroegen we om een gewone (deterministische, 1-tape) Turingmachine T_1 te tekenen, zó dat $L(T_1) = L$. Zo'n Turingmachine vereist voor een invoer x al gauw een aantal stappen dat kwadratisch is in de lengte van x .

We vragen nu: Teken een (deterministische) **2-tapes** Turingmachine T_2 die in een **lineair** aantal stappen bepaalt of een invoer $x \in \{a, b, c\}^*$ wel of niet in L zit. In het eerste geval moet T_2 (uiteraard) accepteren, in het tweede geval (impliciet of expliciet) verwerpen of crashen.

Als je voor T_2 gebruik wilt maken van componenten, moeten dat wel 2-tapes componenten zijn, en zul je ze ook moeten uitwerken (tekenen dus).

Leg ook duidelijk uit hoe T_2 werkt.

Wellicht ten overvloede: een lineair aantal stappen betekent dat het aantal stappen grofweg een constante keer de lengte van de invoer is.

3. [7 pt] Teken een gewone (deterministische, 1-tape) Turing machine T die de functie $f : \mathbb{N} \rightarrow \mathbb{N}$ berekent, gedefinieerd door

$$f(x) = \lfloor x/2 \rfloor$$

(‘ x gedeeld door 2, naar beneden afgerond’). Ga hierbij uit van de unaire representatie van de natuurlijke getallen.

Als je bij dit onderdeel gebruik wilt maken van componenten, zul je die ook moeten uitwerken (tekenen dus).

4. [21 pt] Laat

$$L = \{x \in \{a, b, c\}^* \mid n_a(x) = n_b(x) \text{ en } n_a(x) \neq n_c(x)\}$$

- (a) Geef de eerste zes strings in de canonieke volgorde van L .
 (b) Geef een *unrestricted grammar* G , zó dat $L(G) = L$.

Leg ook uit wat de functie is van de diverse variabelen en producties in je grammatica G .

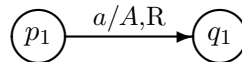
5. [15 pt] In het bewijs van Stelling 8.14 in het boek wordt beschreven, hoe je bij een willekeurige Turingmachine $T = (Q, \Sigma, \Gamma, q_0, \delta)$ een unrestricted grammar $G = (V, \Sigma, S, P)$ kunt construeren, zó dat $L(G) = L(T)$.

De grammatica G bevat drie soorten producties. De eerste soort is bedoeld om initiële configuraties van de Turingmachine na te bouwen, met twee kopieën van de tape-inhoud. De tweede soort producties is voor het simuleren van berekeningen van de Turingmachine.

- (a) Wat is het doel van de derde soort producties van G ?

Als je het antwoord op dit onderdeel niet weet, dan kun je het ‘kopen’ bij de docent. Wellicht kun je dan wel onderdeel (c) maken.

- (b) Geef alle producties van de tweede soort in G die volgens de constructie van Stelling 8.14 corresponderen met de volgende transitie van Turingmachine T :



Je mag de producties algemeen beschrijven in termen van elementen σ_i van Σ en/of Γ en/of $\{\Delta\}$. Maak dan wel duidelijk wat de σ_i 's kunnen zijn. Als alternatief mag je ook aannemen dat $\Sigma = \{a, b\}$ en dat $\Gamma = \{a, b, A\}$.

- (c) Geef alle producties van de derde soort in G (of, althans, producties die hetzelfde effect hebben). Leg ook uit wat de functie is van de diverse producties.

Opnieuw: je mag de producties algemeen beschrijven in termen van elementen σ_i van Σ en/of Γ en/of $\{\Delta\}$. Maak dan wel duidelijk wat de σ_i 's kunnen zijn. Als alternatief mag je ook aannemen dat $\Sigma = \{a, b\}$ en dat $\Gamma = \{a, b, A\}$.

6. [19 pt] Beschouw de volgende twee beslissingsproblemen:

AcceptsEverything:

Gegeven een Turingmachine T_1 , met invoeralfabet Σ ,
is $L(T_1) = \Sigma^*$?

AcceptsAtleastEven:

Gegeven een Turingmachine T_2 , met invoeralfabet Σ ,
accepteert T_2 (in ieder geval) alle strings $x \in \Sigma^*$ van even lengte
(en eventueel ook andere strings).

We willen aantonen dat $\text{AcceptsEverything} \leq \text{AcceptsAtleastEven}$, ofwel dat we *AcceptsEverything* kunnen reduceren naar *AcceptsAtleastEven*.

- (a) Beredeneer, met behulp van een tegenvoorbeeld, waarom we geen geldige reductie krijgen als we simpelweg $T_2 = T_1$ nemen (met T_1 de instantie van *AcceptsEverything* en T_2 de instantie van *AcceptsAtleastEven*).
- (b) Toon aan dat desondanks geldt dat $\text{AcceptsEverything} \leq \text{AcceptsAtleastEven}$. Doe dit door nu een geldige reductie tussen de twee beslissingsproblemen te beschrijven. Laat uiteraard ook zien dat aan alle eisen van een reductie is voldaan.

einde tentamen