

TENTAMEN FORMELE TALEN EN BEREKENBAARHEID

Vrijdag 9 januari 2026, 09.00 - 12.00 uur

Dit tentamen bestaat uit acht 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.

Als we het in dit tentamen over een eindige automaat hebben (zonder verdere toevoeging), bedoelen we een deterministische eindige automaat zonder Λ -transities (wat elders *DFA* genoemd wordt).

1. [8 pt] Laat

$$L = \{x \in \{a, b\}^* \mid |x| \text{ is oneven en } x \text{ bevat (minstens) een substring } bb\}$$

Teken een eindige automaat M , zó dat $L(M) = L$.

2. [10 pt] Het pomplemma voor reguliere talen luidt als volgt:

Stel L is een taal over het alfabet Σ .

Als L geaccepteerd wordt door een eindige automaat $M = (Q, \Sigma, q_0, A, \delta)$,

en als n het aantal toestanden van M is,

dan zijn er voor elke $x \in L$ met $|x| \geq n$ drie strings u, v en w

zó dat $x = uvw$ en de volgende drie beweringen waar zijn:

1. $|uv| \leq n$.
2. $|v| \geq 1$ (dwz: $v \neq \Lambda$).
3. Voor elke $m \geq 0$ zit ook de string $uv^m w$ in L .

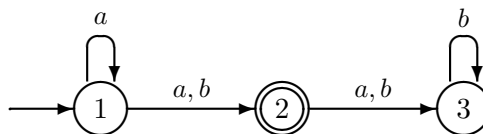
Gebruik dit pomplemma om aan te tonen dat de taal

$$L = \{a^i b^j a^k \mid 0 \leq i + j < k\}$$

niet door een eindige automaat geaccepteerd kan worden.

Ofwel: veronderstel dat L wél door een eindige automaat geaccepteerd kan worden, kies dan een geschikt woord $x \in L$ en toon aan dat x niet opgepompt en/of weggepompt kan worden. Vergeet ook niet om de conclusie te trekken.

3. [8 pt] Beschouw de volgende niet-deterministische eindige automaat M :



Verwijder het niet-determinisme van M met behulp van de subsetconstructie. Je hoeft alleen de resulterende automaat te geven. Zorg dat de oorspronkelijke toestanden nog duidelijk in je automaat te herkennen zijn. Je hoeft de niet-bereikbare toestanden die je volgens de algemene constructie zou krijgen, met hun transities, niet te tekenen. Verder moet je de constructie letterlijk toepassen.

4. [10 pt] Bij deze opgave wordt je vier keer gevraagd om een antwoord ‘ja’ of ‘nee’ te geven. Als het antwoord ‘ja’ is, hoef je dat niet toe te lichten.

Laat $L = \{x \in \{a, b\}^* \mid x \text{ bevat geen substring } bb\}$.

- (a) Laat L_1 de taal zijn die beschreven wordt door de volgende reguliere expressie:

$$(\Lambda + b)(a + aba)^*(\Lambda + b)$$

- i. Is $L_1 \subseteq L$? Zo nee, geef een string x die wel in L_1 zit, maar niet in L .
- ii. Is $L \subseteq L_1$? Zo nee, geef een string x die wel in L zit, maar niet in L_1 .

Inderdaad, als het antwoord twee keer ‘ja’ is, hebben we een correcte reguliere expressie voor taal L .

- (b) Laat L_2 de taal zijn die beschreven wordt door de volgende reguliere expressie:

$$\Lambda + (a + b)(aa^*b)^*a^*$$

- i. Is $L_2 \subseteq L$? Zo nee, geef een string x die wel in L_2 zit, maar niet in L .
- ii. Is $L \subseteq L_2$? Zo nee, geef een string x die wel in L zit, maar niet in L_2 .

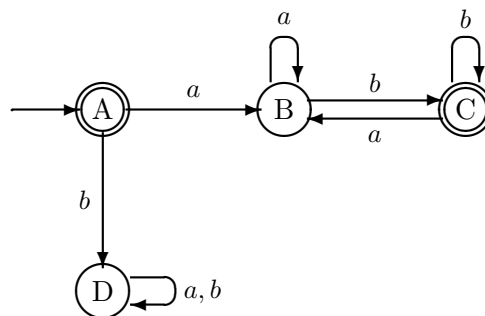
Inderdaad, als het antwoord twee keer ‘ja’ is, hebben we een correcte reguliere expressie voor taal L .

5. [10 pt]

- (a) Wanneer noemen we een context-vrije grammatica $G = (V, \Sigma, S, P)$ *regulier* (wat elders *rechts-lineair* genoemd wordt).

Als je dit antwoord niet weet, kun je het ‘kopen’ van de docent, zodat je het volgende onderdeel wellicht wel kunt maken.

- (b) Tijdens college hebben we een constructie behandeld om voor een gegeven eindige automaat M een reguliere grammatica G te vinden, zó dat $L(G) = L(M)$. Pas deze constructie toe op onderstaande automaat M , dat wil zeggen: geef de resulterende grammatica voor deze automaat:



6. [23 pt] Laat opnieuw

$$L = \{a^i b^j a^k \mid 0 \leq i + j < k\}$$

Bijvoorbeeld, de string $aa \in L$, omdat die geschreven kan worden als $a^i b^j a^k$ met $i = j = 0$ en $k = 2$.

- (a) Geef een context-vrije grammatica G , zó dat $L(G) = L$. Probeer ervoor te zorgen dat G ondubbelzinnig is. Als dit niet lukt, kun je nog wel het grootste deel van de punten verdienen.

Als je context-vrije grammatica dubbelzinnig is, geef dan twee verschillende afleidingsbomen voor een string $x \in L$.

- (b) Teken een stapelautomaat M , zodat $L(M) = L$.

Deze stapelautomaat moet rechtstreeks gebaseerd zijn op eigenschappen van de taal. Hij moet dus niet het resultaat zijn van een standaardconstructie om bijvoorbeeld een context-vrije grammatica om te zetten in een stapelautomaat.

Probeer ervoor te zorgen dat M deterministisch is en geen Λ -transities bevat. Als dit niet lukt, kun je nog wel het grootste deel van de punten verdienen.

Leg ook duidelijk uit hoe M zijn toestanden en stapel(symbolen) gebruikt om precies de juiste taal te accepteren.

7. [17 pt] Laat de functie $f : \{a, b\}^* \rightarrow \{a, b\}^*$ gedefinieerd zijn, door $f(x) = a^{n_a(x)} b^{n_b(x)}$. Dat wil zeggen: f ordent de letters in haar argument x , bijvoorbeeld $f(aababa) = aaaabb$.

Construeer een gewone (deterministische, 1-tape) Turingmachine T die deze functie f berekent. Als je voor T gebruik wilt maken van componenten¹, zul je die componenten ook moeten tekenen.

Leg duidelijk uit hoe T werkt.

8. [14 pt] Beschouw de volgende twee beslissingsproblemen:

Accepts:

Gegeven een Turing machine T_1 en een string x , is $x \in L(T_1)$?

AcceptsEverything:

Gegeven een Turingmachine T_2 , met invoeralfabet Σ , is $L(T_2) = \Sigma^*$?

Toon aan dat $\text{Accepts} \leq \text{AcceptsEverything}$. Doe dit door een geldige reductie tussen de twee beslissingsproblemen te beschrijven. Laat uiteraard ook zien dat aan alle eisen van een reductie is voldaan.

Als je geen geschikte reductie tussen de twee beslissingsproblemen weet, kun je een deel van de punten verdienen door uit te leggen hoe je voor algemene beslissingsproblemen P_1 en P_2 aantoont dat $P_1 \leq P_2$.

einde tentamen

¹Bij het vak Formele Talen en Berekenbaarheid hebben we dit jaar het onderwerp componenten niet eens behandeld.