

Voornaam:

Achternaam:

AB: gequoteerde oefenzitting 1 22 nov 2022 09u00 tot 10u30

Richtlijnen Elke vraag wordt gevolgd door een kader waarin je antwoord moet komen: wat je buiten dat kader schrijft wordt niet beschouwd als deel van je antwoord en je schrijft er best niks. Je geeft enkel deze opgavebladen af. Structureer je antwoord en schrijf duidelijk: voor klad dienen de kladbladen die je kan krijgen: die mag je weggooien nadien. Op je werkplek liggen enkel de kladbladen, deze opgavenbladen, je schrijfgerief en je cursustekst waarin geen opgeloste oefeningen (behalve voor de zelf-doens) mogen staan. Zet je gsm uit en steek deze in je boekentas. Leg je studentenkaart klaar met foto en naam naar boven. Bij het afgeven zorg je dat je naam op je bladen staat, en dat je alles aan elkaar niet of laat nieten. Bij elke vraag staat het gewicht - het totaal is 20.

1 Algebra van talen (4p)

Gegeven de taal $L_d(n)$ van alle binaire voorstellingen van delers van n en een contextvrije taal L_C over hetzelfde alfabet, bespreek de volgende uitspraken (altijd juist, altijd fout, soms juist, soms fout): vink het juiste kastje aan en bewijs jouw antwoord (we gebruiken de notatie $\bar{L}$ voor het complement van een taal L en L^R voor het omgekeerde van een taal).

1. $L_C \cup \bar{L}_d(n)$ is contextvrij

☒ altijd juist ☐ altijd fout ☐ soms juist, soms fout

Voor elk geheel getal n geldt dat het aantal delers eindig is, dus dat $L_d(n)$ een eindige taal is. Iedere eindige taal is regulier, want de taal wordt bepaald door een exhaustieve opsomming van alle elementen als reguliere expressie. Het complement van een reguliere taal is opnieuw regulier, dus $\bar{L}_d(n)$ is opnieuw regulier. Iedere reguliere taal is een contextvrije taal en de unie operatie is gesloten over contextvrije talen, dus $L_C \cup \bar{L}_d(n)$ is altijd contextvrij.

Vaak gemaakte fouten:

- Je moet bewijzen/verklaren waarom $L_d(n)$ een reguliere taal is. Gewoon zeggen dat $L_d(n)$ regulier is of dat je er een DFA voor kunt opstellen, is onvoldoende. Het bewijs moet ook juist zijn.
- Als je aangeeft 'soms juist, soms fout', moet je beide kanten bewijzen. Dit kan door een voorbeeld te geven voor beide. In dit geval zou je natuurlijk geen correct voorbeeld kunnen geven waar deze unie niet contextvrij is.

2. $\bar{L}_C \cap L_d^R(n)$ is regulier

☒ altijd juist ☐ altijd fout ☐ soms juist, soms fout

Aangezien $L_d(n)$ een eindige taal is en $L_d^R(n)$ de omgekeerde taal is, weten we dat $L_d^R(n)$ ook eindig moet zijn, want voor iedere string s in $L_d(n)$ bestaat er exact 1 string s' in $L_d^R(n)$ zodat s het omgekeerde is van s' . Dus het aantal strings in de twee talen is gelijk. De doorsnede van eender welke taal met een eindige taal is opnieuw eindig, dus $\bar{L}_C \cap L_d^R(n)$ is altijd eindig. Iedere eindige taal is regulier, dus $\bar{L}_C \cap L_d^R(n)$ is altijd regulier.

Vaak gemaakte fouten:

- De doorsnede van een taal met eender welke reguliere taal is niet altijd regulier.
- Analooq: 'soms juist, soms fout' bewijzen vraagt een voorbeeld voor beide.

2 Typfouten (6p)

Iedereen maakt wel eens een typfoutje. Eén van de vaakst voorkomende typfouten betreft twee opeenvolgende letters die van plaats gewisseld worden. Een andere vaak voorkomende typfout komt voor wanneer een aangrenzende toets wordt aangeslagen in plaats van de bedoelde toets.

Gegeven een taal L over een alfabet Σ kunnen we de volgende nieuwe talen construeren:

$$Switched(L) = \{w' | w \in L \wedge w' = w \text{ op twee opeenvolgende symbolen na die van plaats gewisseld zijn}\}$$

$$Ernaast(L) = \{w' | w \in L \wedge w' = w \text{ op één symbool } s \text{ na dat vervangen werd door een } s' \in \text{buren}(s)\}$$

met $\text{buren}(s)$ de set symbolen waarvoor de toets grenst aan de toets voor s .

1. Stel dat L regulier is, bewijs dat $Switched(L)$ dan ook regulier is.

Aangezien L regulier is, weten we dat er een DFA bestaat voor L : neem de DFA $D_1 = \langle Q_1, \Sigma, \delta_1, q_{s1}, F_1 \rangle$.

Nu construeren we een NFA $N = \langle Q, \Sigma, \delta, q_s, F \rangle$ voor $Switched(L)$.

Eerst maken we een kopie van D_1 : $D_2 = \langle Q_2, \Sigma, \delta_2, q_{s2}, F_2 \rangle$ waarbij

$$Q_2 = \{q^* | q \in Q_1\} \text{ (we hernoemen de toestanden in de kopie)}$$

$$q_{s2} = q_{s1}^*$$

$$F_2 = \{q^* | q \in F_1\}$$

$$\forall q_i^*, q_j^* \in Q_2 : \forall \sigma \in \Sigma : \delta_2(q_i^*, \sigma) = q_j^* \text{ iff } \delta_1(q_i, \sigma) = q_j$$

We beginnen in de eerste DFA D_1 door het pad te volgen tot ‘een fout’ gemaakt wordt. Bij het maken van ‘een fout’ moeten we 2 symbolen $\alpha\beta$ van plaats kunnen verwisselen ($\beta\alpha$). Hiervoor introduceren we een extra set van $|\Sigma| \cdot |Q|$ toestanden waarin we kunnen bijhouden in welke toestand we zouden uitgekomen zijn via de substring $\alpha\beta$ en welk symbool er nog doorlopen moet worden (α). Vanuit deze toestand wordt met α het pad verder gezet naar D_2 om daar de rest van de string te doorlopen en een eindtoestand te bereiken. Volgens deze constructie zal er altijd exact 1 fout plaatsvinden, ondanks mogelijke lussen in je oorspronkelijke DFA.

Deze constructie wordt formeel gegeven door D te nemen zodat:

$$Q = Q_1 \cup Q_2 \cup \{q^\sigma | q \in Q_1, \sigma \in \Sigma\}$$

$$q = q_{s1}$$

$$F = F_2$$

en δ de transitiefunctie zodat

$$\forall q_i \in Q_1 : \forall \sigma \in \Sigma : \delta(q_i, \sigma) = \{\delta_1(q_i, \sigma)\} \cup \{q_j^\alpha | \exists \alpha \in \Sigma \setminus \{\sigma\} : \exists q_k \in Q_1 : \delta_1(q_i, \alpha) = q_k \wedge \delta_1(q_k, \sigma) = q_j\}$$

$$\forall q_i \in Q_1 : \forall \sigma \in \Sigma : \delta(q_i^\sigma, \sigma) = \{q_i^*\}$$

$$\forall q_i \in Q_1 : \forall \sigma, \alpha \in \Sigma \setminus \{\sigma\} : \delta(q_i^\sigma, \alpha) = \{\}$$

$$\forall q_i \in Q_2 : \forall \sigma \in \Sigma : \delta(q_i, \sigma) = \{\delta_2(q_i, \sigma)\}$$

Aangezien we een NFA kunnen opstellen voor $Switched(L)$, is $Switched(L)$ regulier.

Vaak gemaakte fouten:

- Velen maakten geen kopie van de DFA, maar verwisselden gewoon bogen in de originele DFA. Hier ontstaan er dan problemen als er lussen in de originele DFA zijn. Dan kunnen er oneindig veel fouten toegelaten worden in de aanvaarde strings.
- De woorden/strings die tot L behoorden, behoren in het algemeen niet meer tot $Switched(L)$.
- We kunnen niet slechts één extra toestand gebruiken om van de ene DFA naar de andere over te gaan. We hebben er voldoende nodig om informatie bij te houden over de toestanden en symbolen die verwisseld werden.
- Over de transitiefunctie moet goed nagedacht worden: de signatuur, precieze overgangen ...
- We hebben een NFA, geen DFA, door het maken van extra overgangen in D_1 .

2. Stel dat L regulier is, bewijs dat $Ernaast(L)$ dan ook regulier is.

We kunnen een gelijkaardige NFA opstellen gegeven de DFA $D_1 = \langle Q_1, \Sigma, \delta_1, q_{s1}, F_1 \rangle$ voor L .

We maken opnieuw een kopie D_2 analoog als bij de vorige deelvraag.

Net als bij het vorige deel zullen we D_1 doorlopen zolang er geen fout gemaakt is, na het maken van de fout gaan we over naar D_2 . De overgang tussen de 2 DFA's is nu echter veel eenvoudiger, aangezien de fout maar 1 symbool betreft en we dus geen tussentoestand nodig hebben om een tweede symbool te verwerken.

Zo krijgen we formeel als NFA $N = \langle Q, \Sigma, \delta, q_s, F \rangle$ voor $Ernaast(L)$:

$$Q = Q_1 \cup Q_2$$

$$q_s = q_{s1}$$

$$F = F_2$$

en δ de transitiefunctie zodat

$$\forall q_i \in Q_1 : \forall \sigma \in \Sigma : \delta(q_i, \sigma) = \{\delta_1(q_i, \sigma)\} \cup \{q_j^* | \exists \alpha \in \Sigma : \sigma \in \text{buren}(\alpha) \wedge \delta_1(q_i, \alpha) = q_j\}$$

$$\forall q_i \in Q_2 : \forall \sigma \in \Sigma : \delta(q_i, \sigma) = \{\delta_2(q_i, \sigma)\}$$

Aangezien we een NFA kunnen opstellen die $Ernaast(L)$ bepaalt, is $Ernaast(L)$ regulier.

Vaak gemaakte fouten:

- **Analoog als bij vorige: we hebben een kopie nodig om het probleem met lussen te ontlopen.**
- **Analoog als bij vorige: $L \not\subseteq Ernaast(L)$**
- **Analoog als bij vorige: we hebben een NFA, geen DFA**
- **Analoog als bij vorige: Denk goed na over je transitiefunctie: signatuur, precieze overgangen etc.**

3. Welke taal heb ik nodig als ik alle woorden wil aanvaarden die maximaal 1 voorkomen van elke fout hebben. Is deze taal ook regulier?

De set van alle woorden met maximum 1 voorkomen van elke fout
= de unie van de sets met alle woorden

- zonder fouten
- met exact één wissel van opeenvolgende symbolen
- met exact één symbool vervangen door een buursymbool
- met exact één wissel en één vervangen symbool door een buursymbool

Of formeel: $L \cup Switched(L) \cup Ernaast(L) \cup Switched(Ernaast(L))$ Merk op dat $Switched(Ernaast(L)) = Ernaast(Switched(L))$.

Deze taal is regulier als L regulier is, want de reguliere talen zijn gesloten onder de $\cup$ -, $Switched$ - en $Ernaast$ -operaties.

Vaak gemaakte fouten:

- **maximaal 1 van elke fout laat toe dat er 2 fouten zijn, namelijk van elk 1.**
- **één fout van elk wordt niet gegeven door $Ernaast(L) \cap Switched(L)$ of door $Ernaast(L) \setminus Switched(L) \cup Switched(L) \setminus Ernaast(L)$ of dergelijke, maar wel door $Switched(Ernaast(L))$ of $Ernaast(Switched(L))$.**

3 Bepaal de taal (6p)

Zijn de volgende talen 1) regulier; 2) niet regulier maar wel context-vrij; of 3) niet context-vrij? Vink het juiste kastje aan en bewijs.

$L = \{a^i b^j a^k \mid i, j, k \in \mathbb{N} \wedge j < 15 \wedge \text{if } j < 5 \text{ then } k = i + j\}$ met alfabet $\Sigma = \{a, b\}$.

☐ regulier ☒ niet regulier maar wel context-vrij ☐ niet context-vrij

L is niet regulier, bewijs via pompend lemma:

Neem een string $s = a^p b a^{p+1} \in L$ met onbekende pomplengte p .

Dan kunnen we s opsplitsen in xyz volgens de eigenschappen van het pompend lemma. Aangezien $|xy| < p$ weten we dat xy volledig binnen de eerste groep a 'tjes ligt. Dus y is van de vorm a^l met $1 \leq l < p$.

Als we y pompen krijgen we $xy^2z = a^{p+l} b a^{p+1} \notin L$, want $l \neq 0$.

Het pompend lemma geldt niet, dus L is niet regulier.

$L = L_1 \cup L_2$ met $L_1 = \{a^i b^j a^{i+j} \mid i, j, k \in \mathbb{N} \wedge j < 5\}$ en $L_2 = \{a^i b^j a^k \mid i, j, k \in \mathbb{N} \wedge 5 \leq j < 15\}$

L_2 is regulier want er bestaat een reguliere expressie voor de taal: $a^*(b^5|b^6|\dots|b^{14})a^*$

L_1 is contextvrij want er bestaat een CFG voor de taal:

$\langle \{S\}, \{a, b\}, R, S \rangle$ met $R = \{S \rightarrow aSa\} \cup \{S \rightarrow b^i a^i \mid 0 \leq i < 5\}$

Dus L is contextvrij als unie van reguliere en contextvrije talen.

Vaak gemaakte fouten:

- Voor het pompend lemma moet je een string kiezen die in de taal zit. Bijvoorbeeld de string $ab^p a^{p+1}$ met onbekende pomplengte p is niet per se een string uit L , want p kan groter zijn dan 15. Anders maak je assumpties over je pomplengte.
- Je kunt het pompend lemma enkel gebruiken om aan te tonen dat een taal niet regulier/niet contextvrij is, niet om aan te tonen dat een taal wel regulier of contextvrij is.
- Opnieuw: enkel zeggen dat je er een CFG voor kunt opstellen is onvoldoende om te bewijzen dat de taal contextvrij is.
- Je moet zowel aantonen dat L niet regulier is, als dat L contextvrij is, één van de twee is onvoldoende.
- Een intuïtieve redenering voor waarom de taal niet regulier kan zijn, is geen bewijs.

$L = \{a^i b^j a^k \mid i, j, k \in \mathbb{N} \wedge j < 15 \wedge \text{if } i < 5 \text{ then } k = i + j\}$ met alfabet $\Sigma = \{a, b\}$.

☒ regulier ☐ niet regulier maar wel context-vrij ☐ niet context-vrij

$L = L_1 \cup L_2$

met $L_1 = \{a^i b^j a^{i+j} \mid i, j, k \in \mathbb{N} \wedge j < 15 \wedge i < 5\}$

en $L_2 = \{a^i b^j a^k \mid i, j, k \in \mathbb{N} \wedge j < 15 \wedge i \geq 5\}$

L_2 is regulier want er bestaat een reguliere expressie voor de taal: $a^5 a^*(\epsilon|b^1|b^2|\dots|b^{14})a^*$

L_1 is regulier want dit is een eindige taal met slechts 75 strings.

Dus L is regulier als unie van twee reguliere talen.

Vaak gemaakte fouten:

- Gelijkaardig aan de vorige wat betreft beide punten rond het pompend lemma.
- Opnieuw, enkel zeggen dat je een DFA kunt opstellen, is onvoldoende als bewijs. Je moet op zijn minst de DFA in kwestie kunnen beschrijven en geven.

4 Minimale DFA (4p)

Minimaliseer onderstaande DFA aan de hand van een matrix.

Eerst schrappen we de onbereikbare toestanden q_7 en q_8 . Dan stellen we de matrix op met de overige toestanden als kolommen en rijen:

	1	2	3	4	5	6
1		-	0	0	ϵ	-
2			0	0	ϵ	-
3				-	ϵ	0
4					ϵ	0
5						ϵ
6						

waarbij we een string x invullen in cel (i, j) als toestanden i en j met string x in f-verschillende toestanden uitkomen.

Na 2 iteraties bekommen we 2 sets f-gelijke toestanden:

1, 2 en 6

3 en 4

Na het samennemen van die sets bekommen we de minimale DFA rechts.

Vaak gemaakte fouten:

- Eerste stap is altijd het verwijderen van onbereikbare toestanden.
- Er is geen reden om de begintoestand anders te behandelen dan de andere toestanden.
- Toestanden moeten niet in exact dezelfde toestanden uitkomen voor alle symbolen in het alfabet, maar in f-gelijke toestanden.

