TENTAMEN INLEIDING FUNDAMENTELE INFORMATICA
vrijdag 2 augustus 2002, 14.00 - 17.00 uur

Dit tentamen bestaat uit 8 opgaven waarbij steeds de waardering tussen [en] vermeld staat. In totaal zijn er 100 punten te verdienen.

Opgave 1 [12 pt]
De taal $L \subseteq \{0,1\}^*$ wordt recursief gedefinieerd door:
(i) $01, 10 \in L$,
(ii) als $x \in L$, dan is $xx^R \in L$,
(iii) L bevat geen andere woorden.

(a) Ga voor de volgende woorden na of ze al dan niet tot L behoren:
0110 1010 01010 01011010

Laat $K = \{10, 01\} \cup \{w \in \{01, 10\}^+ : w = w^R\}$.

(b) Bewijs met inductie naar de definitie van L dat $L \subseteq K$.

(c) Is $L = K$? Zo ja, geef een bewijs. Zo nee, geef een woord $w \in K - L$.

Opgave 2 [13 pt]

(a) Definieer het begrip aftelbaarheid.

(b) A en B zijn twee oneindige verzamelingen die allebei aftelbaar zijn.
Gegeven is dat A en B disjunct zijn, d.w.z. $A \cap B = \emptyset$.
Bewijs dat ook $A \cup B$ aftelbaar is.

De verzamelingen $A_1, A_2, A_3 \ldots$ zijn aftelbare oneindige verzamelingen die bovendien onderling disjunct zijn, d.w.z. $A_i \cap A_j = \emptyset$ als $i \neq j$.

(c) Bewijs dat voor elke $k \geq 1$ geldt dat $\bigcup_{i=1}^{k} A_i$ aftelbaar is.

(d) Is de oneindige vereniging $\bigcup_{i=1}^{\infty} A_i$ aftelbaar? Waarom (niet)?

Opgave 3 [15 pt]

(a) Wanneer wordt een DTM M een beslisser genoemd?
Definieer de ja-taal en de nee-taal van beslisser M.

(b) Maak een beslisser M met invoeralfabet $\{a, b\}$ voor de taal
$L = \{a^k v : k \geq 0,\ v \in \{a, b\}^* \text{ en } |v|_a = k\}$.
Dus M moet beslissen of het aantal a's in een invoerwoord even is waarbij de eerste helft van de a's vooraan in het woord staat.
Geef M d.m.v. zijn toestandsdiagram en schets het idee achter de constructie.

Opgave 4 [8 pt]
Geef voor elk van de talen L_1, L_2, L_3, L_4 een eindige automaat die deze taal accepteert.

$L_1 = \{w \in \{a, b\}^* : w = xaaay \text{ met } x, y \in \{a, b\}^*\}$, de taal bestaande uit alle woorden over het alfabet $\{a, b\}$ die een subwoord aaa bevatten.

$L_2 = \{w \in \{a, b\}^* : |w| \text{ is even}\}$, de taal bestaande uit alle woorden van even lengte over het alfabet $\{a, b\}$.

$L_3 = L_1 \cap L_2$, de taal bestaande uit alle woorden van even lengte over het alfabet $\{a, b\}$ die een subwoord aaa bevatten.

$L_4 = \{a, b\}^* - L_1$, de taal bestaande uit alle woorden over het alfabet $\{a, b\}$ die geen subwoord aaa bevatten.

Opgave 5 [`12 pt`]
De eindige automaat M is gegeven door $M = (\{0,1,2,3\}, \{a,b\}, \delta, \{0\}, \{3\})$ met $\delta = \{(0,a,0), (0,a,1), (0,b,0), (1,b,2), (1,b,3), (2,a,3)\}$.

(a) Teken het toestandsdiagram van M.

(b) Bepaal met behulp van configuratierijen of de volgende woorden worden geaccepteerd door M: aba baa $baab$

(c) Construeer met behulp van de subsetconstructie een deterministische eindige automaat die equivalent is met M.

Opgave 6 [`14 pt`]
Gegeven is de context-vrije grammatica G met axioma S en producties:
$S \to AC \quad A \to aAb \mid a \quad C \to bCc \mid b$

(a) Kies drie woorden uit $L(G)$ en geef voor elk een afleiding.

(b) Bewijs met inductie naar de lengte van de afleiding:
als $S \Rightarrow^* w$, dan is $|w|_b = |w|_a + |w|_A + |w|_c - |w|_C$.
Herinnering: $|u|_d$ is het aantal malen dat de letter d voorkomt in het woord u.

(c) Is $L(G) = \{a^k b^n c^m : n = k + m \geq 1\}$? Waarom (niet)?

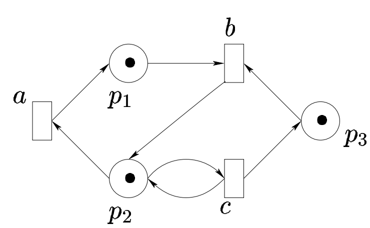
Opgave 7 [`11 pt`]
Bekijk de context-vrije grammatica G met axioma S en producties:
$S \to ABC \quad A \to aAB \mid \lambda \quad B \to Bb \mid b \quad C \to bCc \mid A$

(a) Bepaal alle λ-niet-terminalen van G. Is $\lambda \in L(G)$?

(b) Maak een CFG G' in λ-normaalvorm die equivalent is met G.

(c) Maak G' zonodig ketenvrij (met behoud van zijn λ-normaalvorm).

Opgave 8 [`15 pt`]
Gegeven is het Petri net $N = (P, T, F)$ met beginmarkering M:



(a) Welke transities hebben concessie in M?
Welke daarvan zijn in conflict en welke hebben concurrent concessie?

(b) Geef een vuurrij w van N die van M naar een deadlock leidt.

(c) Beredeneer dat voor *elke* markering M' van N geldt dat er vanuit M' een deadlock bereikbaar is.

(d) Is (N, M) safe? begrensd? levend? Waarom (niet)?

`einde tentamen`

TENTAMEN INLEIDING FUNDAMENTELE INFORMATICA
UITWERKINGEN
vrijdag 2 augustus 2002, 14.00 - 17.00 uur

Opgave 1

(a) — $0110 \in L$, want $01 \in L$ volgens (i) en $10 = (01)^R$ dus volgens (ii) is $0110 \in L$.
— $1010 \notin L$, want 1010 wordt niet genoemd in (i) en 1010 kan alleen met behulp van (ii) zijn geconstrueerd als $(10)^R = 10$, wat niet het geval is. Volgens (iii) bevat L geen andere woorden dan die beschreven in (i) en (ii).
— $01010 \notin L$, want 01010 wordt niet genoemd in (i) en 01010 kan ook niet met behulp van (ii) zijn geconstrueerd, omdat dat alleen woorden van even lengte oplevert. Volgens (iii) bevat L geen andere woorden dan die beschreven in (i) en (ii).
— $01011010 \notin L$, want 01011010 wordt niet genoemd in (i) en 01011010 kan alleen met behulp van (ii) zijn geconstrueerd, als $0101 \in L$ wat niet het geval is zoals we net hebben gezien. Volgens (iii) bevat L geen andere woorden dan die beschreven in (i) en (ii).

(b) Te bewijzen: voor alle $z \in L$ geldt $z \in K = \{10, 01\} \cup \{w \in \{01, 10\}^+ : w = w^R\}$.
Bewijs met inductie naar de definitie van L:
(basis) als $z \in L$ volgens (i), dan is $z = 01$ of $z = 10$ dus $z \in K$.
(inductiehypothese) neem nu aan dat $z = xx^R$ met $x \in L$ zodat $x \in K$
(inductiestap) dan is $z \in \{01, 10\}^+$ en $z^R = (xx^R)^R = (x^R)^R x^R = xx^R = z$, dus $z \in K$.
Dus door (ii) toe te passen op een woord $x \in L$ zodat $x \in K$ krijg je weer een woord uit K. Uit (iii) volgt nu dat elk woord van L ook een woord uit K is.

(c) $L \neq K$, want bijvoorbeeld $01011010 \in K - L$.
Immers $01011010 \in \{01, 10\}^+$ en $(01011010)^R = 01011010$ dus $01011010 \in K$, terwijl $01011010 \notin L$ volgens (a).

Opgave 2

(a) Een verzameling V heet aftelbaar als V ofwel eindig is ofwel gelijkmachtig met $\mathbb{N}$ (d.w.z. er bestaat een bijectie $f : V \longrightarrow \mathbb{N}$).

(b) A en B zijn beide oneindig en aftelbaar, dus er is een bijectie $f : A \to \mathbb{N}$ en er is een bijectie $g : B \to \mathbb{N}$. Omdat $A \cap B = \emptyset$ kunnen we deze bijecties combineren tot de functie $h : A \cup B \to \mathbb{N}$ gedefinieerd door: $h(a) = [2 \times f(a)] - 1$ als $a \in A$ en $h(b) = 2 \times g(b)$ als $b \in B$. Deze functie is een bijectie, immers
h is injectief: op de eerste plaats is h totaal, omdat $h(c)$ is gedefinieerd voor elk element $c \in A \cup B$ ofwel via f als $c \in A$ ofwel via g als $c \in B$ en f en g zijn totaal; ten tweede is $h(c) \neq h(d)$ als $c \neq d$, omdat als de een uit A afkomstig is en de ander uit B dan levert h voor de eerste een oneven waarde en voor de tweede een even waarde, terwijl als beide afkomstig zijn uit A (dan wel B) de injectiviteit van f (respectievelijk g) garandeert dat $h(c) \neq h(d)$.
h is surjectief: voor elk element van $n \in \mathbb{N}$ is er een $c \in A \cup B$ zodat $h(c) = n$, omdat als $n = 2k - 1$, dan is er een $a \in A$ zodat $f(a) = k$ en dus is $h(a) = 2k - 1 = n$ en als $n = 2k$, dan is er een $B \in B$ zodat $g(b) = k$ en dus is $h(b) = 2k = n$.

(c) Te bewijzen: voor alle $k \geq 1$, als de verzamelingen A_i, $1 \leq i \leq k$, aftelbaar en onderling disjunct zijn, dan is $\bigcup_{i=1}^{k} A_i$ aftelbaar.
Dat kan bijvoorbeeld door gebruik te maken van het vorige onderdeel en dan

inductie toe te passen:
(basis) $k = 1$: $\bigcup_{i=1}^{1} A_i = A_1$ is aftelbaar.
(inductiehypothese) Er is een $j \geq 1$ zodat $\bigcup_{i=1}^{j} A_i$ aftelbaar is.
(inductiestap) $k = j + 1$: $\bigcup_{i=1}^{k} A_i = \bigcup_{i=1}^{j+1} A_i = (\bigcup_{i=1}^{j} A_i) \cup A_{j+1}$. Volgens de inductiehypothese is $\bigcup_{i=1}^{j} A_i$ aftelbaar en gegeven is dat A_{j+1} aftelbaar is. Omdat $A_{j+1} \cap \bigcup_{i=1}^{j} A_i = \emptyset$, kunnen we uit onderdeel (b) concluderen dat $(\bigcup_{i=1}^{j} A_i) \cup A_{j+1}$ aftelbaar is.
In plaats van een inductiebewijs te geven kan je bijvoorbeeld ook expliciete bijecties definiëren (afhankelijk van k) of zoals in het volgende onderdeel de elementen in een tabel (nu met k rijen) opstellen en een Cantor(achtige) wandeling beschrijven (wat dus een beschrijving van een bijectie is).
Merk op dat op deze manier onderdeel (b) een speciaal geval ($k = 2$) is van dit onderdeel.

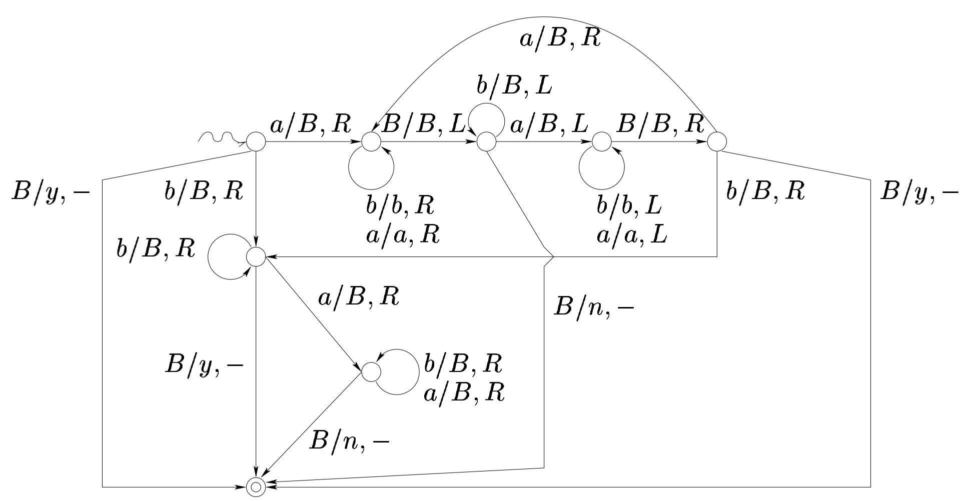
(d) $(\bigcup_{i=1}^{\infty} A_i)$ is aftelbaar. Immers we kunnen een oneindige tabel maken gebruikmakend van zowel de aftelbaarheid van de A_i en van het aftellen van de indexen en daar dan met een Cantorwandeling door heen lopen bij wijze van aftelling.

$A_1 =$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$\ldots$
$A_2 =$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$\ldots$
$A_3 =$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$\ldots$
$A_4 =$	$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	$a_{4,4}$	$\ldots$
$\ldots$	$\ldots$				

Opgave 3

(a) Een deterministische Turingmachine $M = (Q, \Sigma, \Gamma, \delta, s, f)$ is een beslisser als er twee onderscheiden symbolen $\mathbf{y}, \mathbf{n} \in \Gamma - \Sigma$ zijn zodat voor alle $x \in \Sigma^*$ ofwel $sx \vdash^* f\mathbf{y}$ ofwel $sx \vdash^* f\mathbf{n}$.
$Y(M) = \{x \in \Sigma^* : sx \vdash^* f\mathbf{y}\}$ is de ja-taal van M.
$N(M) = \{x \in \Sigma^* : sx \vdash^* f\mathbf{n}\} = \Sigma^* - Y(M)$ is de nee-taal van M.

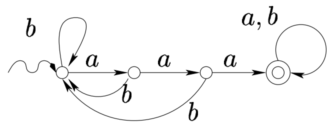
(b) Het lege woord voldoet aan de eisen, dus bij lege invoer mag M gelijk $\mathbf{y}$ afdrukken en stoppen.
Als het invoerwoord met een b begint, mogen daarna alleen nog b's volgen. De kop gaat van links naar rechts door het woord onderwijl alle symbolen uitvegend. Als daarbij geen a wordt gelezen, stopt de machine met $\mathbf{y}$ op de band, anders wordt $\mathbf{n}$ afgedrukt.



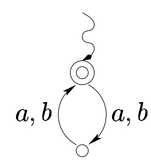
Als het invoerwoord met een a begint, wordt deze uitgeveegd en gaat de kop helemaal naar rechts. Vanuit het rechteruiteinde wordt dan naar links gelopen, terwijl alle b's worden uitgeveegd tot
OF een a wordt aangetroffen die als "match" met de net links uitgeveegde a ook wordt uitgeveegd. De kop gaat helemaal terug naar het begin van het beschreven deel van de band en het spel begint opnieuw;
OF geen a meer wordt gevonden in welk geval er geen "match" was. De band is nu leeg, de machine drukt **n** af en stopt.

Opgave 4

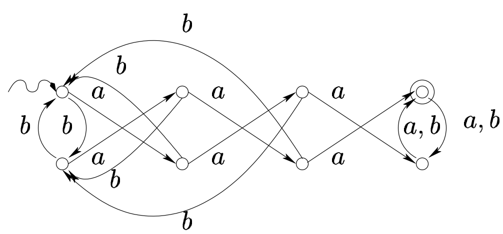
Voor L_1 maken we een standaard deterministische patroonherkenner voor aaa.
Voor L_2 zijn twee toestanden genoeg om bij te houden of het tot dan toe gelezen aantal letters even of oneven was. Alleen na een even aantal letters kan worden geaccepteerd.
Voor L_3 combineren we deze twee ideeën. Er wordt gezocht naar het patroon terwijl gelijktijdig even/oneven wordt bijgehouden door voor elke toestand uit de DFA voor L_1 nu een "even" en een "oneven" kopie te hebben. Alleen de even kopie van de eindtoestand is accepterend.
Het complement van L_1 wordt geaccepteerd door de DFA verkregen door in de DFA voor L_1 eindtoestanden en niet-eindtoestanden van rol te laten verwisselen (merk op dat onze automaat voor L_1 al van zichzelf volledig is!).



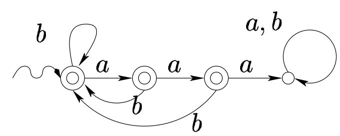
L_1



L_2



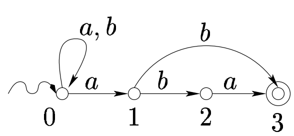
L_3



L_4

Opgave 5

(a) De automaat M ziet er zo uit:



(b) Het woord aba wordt geaccepteerd, want $0aba \vdash 1ba \vdash 2a \vdash 3$ is een accepterende configuratierij. (Overigens zijn er meer configuratierijen voor aba: $0aba \vdash 0ba \vdash 0a \vdash 0$, geen eindtoestand, $0aba \vdash 0ba \vdash 0a \vdash 1$, geen eindtoestand, $0aba \vdash 1ba \vdash 3a \nvdash$, hangt.)

baa wordt niet geaccepteerd, want **geen enkele** configuratierij voor baa is accepterend: $0baa \vdash 0aa \vdash 0a \vdash 0$, geen eindtoestand, $0baa \vdash 0aa \vdash 0a \vdash 1$, geen eindtoestand en $0baa \vdash 0aa \vdash 1a \not\vdash$, hangt.
$baab$ wordt geaccepteerd, want $0baab \vdash 0aab \vdash 0ab \vdash 1b \vdash 3$ is een accepterende configuratierij. (De andere configuratierijen voor $baab$ zijn: $0baab \vdash 0aab \vdash 0ab \vdash 0b \vdash 0$, niet accepterend, $0baab \vdash 0aab \vdash 0ab \vdash 1b \vdash 2$, niet accepterend, $0baab \vdash 0aab \vdash 1ab \not\vdash$, hangt.)

(c) Met de subsetconstructie maken we uit M de deterministische eindige automaat met begintoestand $\{0\}$, eindtoestanden $\{0,1,3\}$, $\{0,2,3\}$ en

	a	b
$\{0\}$	$\{0,1\}$	$\{0\}$
$\{0,1\}$	$\{0,1\}$	$\{0,2,3\}$
$\{0,2,3\}$	$\{0,1,3\}$	$\{0\}$
$\{0,1,3\}$	$\{0,1\}$	$\{0,2,3\}$

Opgave 6

(a) ab, a^2b^2 en ab^2c zijn voorbeelden van woorden uit $L(G)$:
$S \Rightarrow AC \Rightarrow aC \Rightarrow ab$, $\quad S \Rightarrow AC \Rightarrow aAbC \Rightarrow aabC \Rightarrow aabb = a^2b^2$ en
$S \Rightarrow AC \Rightarrow aC \Rightarrow abCc \Rightarrow abbc = ab^2c$

(b) Te bewijzen: voor alle $n \geq 0$ geldt dat
als $S \Rightarrow^n w$, dan is $|w|_b = |w|_a + |w|_A + |w|_c - |w|_C$.
Bewijs met inductie naar n, de lengte van de afleiding:
(basis, n=0) als $S \Rightarrow^0 w$, dan is $w = S$ en $|S|_b = 0$
terwijl $|S|_a + |S|_A + |S|_c - |S|_C = 0 + 0 + 0 - 0 = 0$ wat te bewijzen was.
(inductiehypothese) er is een $k \geq 0$ zodat
als $S \Rightarrow^k v$, dan is $|v|_b = |v|_a + |v|_A + |v|_c - |v|_C$.
(inductiestap, n = k+1) neem aan dat $S \Rightarrow^n w$. Dus $S \Rightarrow^k v \Rightarrow w$ voor een v.
Volgens de inductiehypothese is $|v|_b = |v|_a + |v|_A + |v|_c - |v|_C$. Er zijn 5 producties die gebruikt kunnen zijn om v te herschrijven in w:
$S \to AC$: dan is $|w|_b = |v|_b$ en $|w|_a + |w|_A + |w|_c - |w|_C = |v|_a + |v|_A + 1 + |v|_c - (|v|_C + 1) = |v|_b$;
$A \to aAb$: dan is $|w|_b = |v|_b + 1$ en $|w|_a + |w|_A + |w|_c - |w|_C = |v|_a + 1 + |v|_A + |v|_c - |v|_C = |v|_b + 1$;
$A \to a$: dan is $|w|_b = |v|_b$ en $|w|_a + |w|_A + |w|_c - |w|_C = |v|_a + 1 + |v|_A - 1 + |v|_c - |v|_C = |v|_b$;
$C \to bCc$: dan is $|w|_b = |v|_b + 1$ en $|w|_a + |w|_A + |w|_c - |w|_C = |v|_a + |v|_A + |v|_c + 1 - |v|_C = |v|_b + 1$;
$C \to b$: dan is $|w|_b = |v|_b + 1$ en $|w|_a + |w|_A + |w|_c - |w|_C = |v|_a + |v|_A + |v|_c - (|v|_C - 1) = |v|_b + 1$.

(c) Nee, het woord $bc = a^0b^1c^1$ voldoet aan de eis $1 = 0 + 1$, maar $L(G)$ bevat geen woorden zonder a's omdat in elke afleidng $S \Rightarrow AC \Rightarrow \ldots$ de niet-terminaal A altijd tenminste één a zal introduceren.

Opgave 7

(a) De λ-niet terminalen van G zijn: A, want $A \to \lambda$ is een productie van G, en C, want $C \to A$ is een productie. S is geen λ-niet-terminaal, dus $\lambda \notin L(G)$.

(b) Simuleren van herschrijven naar λ:
$S \to ABC$ wordt $S \to ABC \mid BC \mid AB \mid B \qquad A \to aAB$ wordt $A \to aAB \mid aB$
$C \to bCc$ wordt $C \to bCc \mid bc \qquad C \to A$ wordt $C \to A \mid \lambda$

Vervolgens worden de λ-producties verwijderd en we houden over:
$S \to ABC \mid BC \mid AB \mid B \quad A \to aAB \mid aB \quad B \to Bb \mid b \quad C \to bCc \mid bc \mid A.$

(c) De grammatica van het vorige onderdeel heeft twee ketenproducties:
$S \to B$ wordt vervangen door $S \to Bb \mid b$ $\qquad$ $C \to A$ door $C \to aAB \mid aB$.

Opgave 8

(a) In M hebben alle transities concessie (alle invoerplaatsen zijn gevuld). De transities a en b hebben concurrent concessie, evenals b en c, want ze hebben geen gemeenschappelijke invoerplaats. a en c zijn in conflict, want hun gedeelde invoerplaats p_2 bevat maar 1 teken.

(b) $(1,1,1)[b\rangle(0,2,0)[a\rangle(1,1,0)[a\rangle(2,0,0)$ waarin geen enkele transitie concessie heeft.

(c) Als p_1 en p_2 beide geen tekens bevatten, hebben we gelijk al een deadlock.
Als p_1 minstens een teken bevat: vuur zo vaak achter elkaar ba als er tekens zijn in p_3 en dan zo vaak a als er nog tekens zijn in p_2:
$(i+1,j,k)[(ba)^k\rangle(i+1,j,0)[a^j\rangle(i+1+j,0,0)$, een deadlock.
Als p_2 minstens een teken bevat, dan kan a vuren waarna we weer in de vorige situatie zijn: $(i,j+1,k)[a\rangle(i+1,j,k\rangle$.

(d) (N,M) is niet begrensd: c kan herhaald vuren waardoor het aantal tekens in p_3 willekeurig groot kan worden: $M[c^k\rangle(1,1,1+k)$. Dus (N,M) is ook niet safe.
(N,M) is niet levend, want zoals net aangetoond niet deadlock-vrij.