Berekenbaarheid 2018
Uitwerkingen Tentamen
1 november 2018

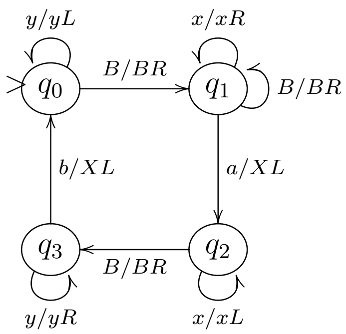
1. We definiëren de taal L_1 als

$$L_1 := \{w \in \{a,b\}^* \mid |w|_a > |w|_b\}$$

waarin $|w|_a$ het aantal a's in w en $|w|_b$ het aantal b's in w is. Er geldt bijvoorbeeld $babaa \in L_1$ want $|babaa|_a = 3 > |babaa|_b = 2$, maar $aabb \notin L_1$ want $|aabb|_a = 2 \not> |aabb|_b = 2$.

Geef een standaard Turing-machine M_1 die L_1 accepteert door stoppen.

M_1:



$x \in \{b, X\}$
$y \in \{a, X\}$

Als er meer a's dan b's in de input zijn, zal de machine geen b meer vinden in q_3 en daar stoppen. Als er niet meer a's dan b's in de input zijn, dan zal de machine op een gegeven moment geen a vinden in q_1, en daar eindeloos langs de blanks op de tape naar rechts gaan lopen.

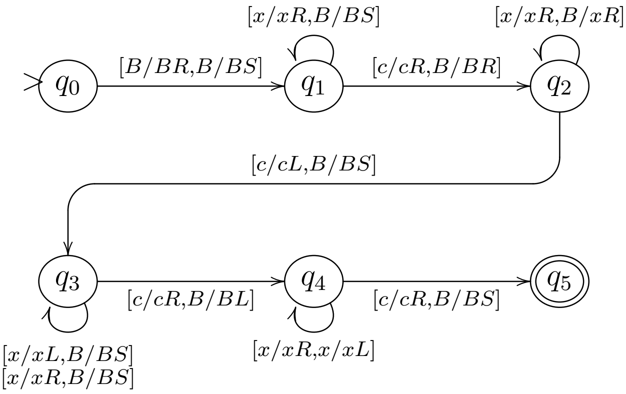
2. We definiëren de taal L_2 als

$$L_2 := \{w \in \{a,b,c\}^* \mid \text{er is een } u \in \{a,b,c\}^* \text{ zodat } w \text{ zowel } cuc \text{ als } cu^Rc \text{ bevat}\}$$

Er geldt bijvoorbeeld $acacbbcacb \in L_2$ want $acacbbcacb$ bevat onder meer zowel cuc als cu^Rc met $u = acbb$ (dat zijn dus de woorden $cacbbc$ en $cbbcac$: $a\underline{cacbbc}acb$ en $aca\underline{cbbcac}b$).

Geef een non-deterministische Turing-machine M_2 met twee tapes die de taal L_2 accepteert door eindtoestand. Een woord uit de taal $w \in L_2$ moet worden herkend in minder dan $3|w| + 2$ stappen.

M_2:



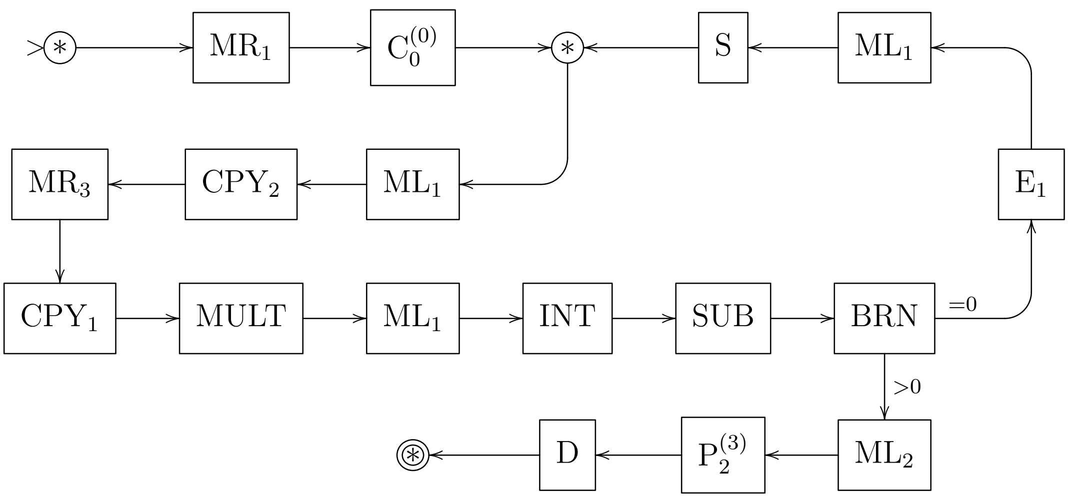
$x \in \{a, b, c\}$

3. We definiëren de numerieke functie sqrt door

$$\mathsf{sqrt}(x) := \lfloor \sqrt{x} \rfloor$$

Geef een numerieke Turing-machine die sqrt uitrekent. Je mag in je machine de macro's gebruiken uit de lijst op pagina 7.

Hint: Zoek in een lus de kleinste n met $n^2 > x$, en trek daar vervolgens nog één vanaf.



4. (a) Als een taal L recursief is, is het complement $\overline{L}$ dan ook altijd recursief?

Ja.

Als L recursief is wordt hij per definitie geaccepteerd door eindtoestand door een Turing-machine die voor iedere input stopt. Door bij alle toestanden het eindtoestand-zijn om te draaien krijgen we een machine die $\overline{L}$ herkent en ook voor iedere input stopt.

(b) Als een taal L recursief opsombaar is, is het complement $\overline{L}$ dan ook altijd recursief opsombaar?

Nee.

Als een taal L zowel als zijn complement $\overline{L}$ recursief opsombaar zijn, dan is de taal ook recursief. Maar er zijn recursief opsombare talen die niet recursief zijn.

Neem bijvoorbeeld als tegenvoorbeeld

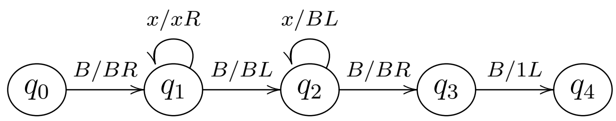
$$L_H := \{R(M)w \mid M(w)\downarrow\}$$

de taal van het halting-probleem. Deze taal is recursief opsombaar, want wordt herkend door stoppen door de universele Turing-machine, maar is niet recursief, want anders zou het halting-probleem beslisbaar zijn.

Geef in beide gevallen een verklaring van je antwoord, en als het antwoord negatief is daarbij ook een tegenvoorbeeld.

5. Geef een code $R(M_5)$ van een Turing-machine M_5 zodat $M_5(R(M_5)) = 1$. Teken bij je antwoord ook het toestandsdiagram van M_5.

M_5:



$x \in \{0, 1\}$

De code van deze machine is

$$\begin{aligned} R(M_5) = 000\,&\underbrace{101110110111011}_{\delta(q_0,B)=[q_1,B,R]}00\underbrace{110101101011}_{\delta(q_1,0)=[q_1,0,R]}00\underbrace{11011011011011}_{\delta(q_1,1)=[q_1,1,R]} \\ &00\underbrace{1101110111011101}_{\delta(q_1,B)=[q_2,B,L]}00\underbrace{1110101110111101}_{\delta(q_2,0)=[q_2,B,L]}00\underbrace{1110110111011101}_{\delta(q_2,1)=[q_2,B,L]} \\ &00\underbrace{11101110111101110 11}_{\delta(q_2,B)=[q_3,B,R]}00\underbrace{1111011101111101101}_{\delta(q_3,B)=[q_4,1,L]}000 \end{aligned}$$

ofwel, zonder de uitleg-decoratie

$$R(M_5) = \begin{array}{l}0001011101101110110011010110101100110110110110110011011101110111010011101\\0111011101001110110111011101001110111011110111011001111011101111101101000\end{array}$$

6. Het probleem P_6 heeft als input een code $R(M)$ van een Turing-machine, en vraagt hierover of de taal $L(M)$ van deze machine eindig is. Laat zien (zonder de stelling van Rice te gebruiken, want die is dit jaar niet behandeld) dat P_6 onbeslisbaar is.

   We bewijzen dat het omgekeerde probleem $\overline{P_6}$ dat vraagt of de taal *on*eindig is onbeslisbaar is. Daaruit volgt dat P_6 ook onbeslisbaar is.

   Het probleem $\overline{P_6}$ is onbeslisbaar, want het blank tape probleem B reduceert naar dit probleem. Als $\overline{P_6}$ beslisbaar was, was B ook beslisbaar, maar dat is niet zo. Dus is $\overline{P_6}$ onbeslisbaar.

   Voor de reductie moeten we bij iedere input $R(M)$ van B een input $R(M')$ voor $\overline{P_6}$ construeren zodat geldt

$$M(\lambda)\downarrow \iff L(M') \text{ is oneindig}$$

   want dan geeft $\overline{P_6}$ het antwoord op de vraag van B.

   De constructie van M' is:

   (a) wis de tape

   (b) doe M

   We hebben dan

$$\begin{array}{llll} M(\lambda)\downarrow & \implies M' \text{ stopt voor iedere input} & \implies L(M') = \Sigma^* & \text{is oneindig} \\ M(\lambda)\uparrow & \implies M' \text{ stopt voor geen enkele input} & \implies L(M') = \emptyset & \text{is eindig} \end{array}$$

   waaruit de gezochte equivalentie volgt.

7. Geef numerieke functies f_7 en g_7 met $f_7 \neq g_7$ maar $f_7 \circ g_7 = g_7 \circ f_7$. Verklaar je antwoord.

   Neem bijvoorbeeld

$$\begin{aligned} f_7 &= \mathsf{id} \\ g_7 &= e \end{aligned}$$

   Dan $f_7 \neq g_7$ want $f_7(0)\downarrow$ en $g_7(0)\uparrow$. Maar $f_7 \circ g_7 = e = g_7 \circ f_7$.

8. We definiëren de numerieke functie f_8 met primitieve recursie door de recursievergelijkingen

$$\begin{aligned} f_8(0) &= 0 \\ f_8(y+1) &= f_8(y) + \mathsf{eq}\big((f_8(y)+1)^2, y+1\big) \end{aligned}$$

(a) Geef de waarde van $f_8(4)$.

$$
\begin{aligned}
&f_8(0) = 0\\
&f_8(1) = f_8(0+1) = f_8(0) + \mathsf{eq}\big((f_8(0)+1)^2, 0+1\big) = 0 + \mathsf{eq}(1,1) = 0 + 1 = 1\\
&f_8(2) = f_8(1+1) = f_8(1) + \mathsf{eq}\big((f_8(1)+1)^2, 1+1\big) = 1 + \mathsf{eq}(4,2) = 1 + 0 = 1\\
&f_8(3) = f_8(2+1) = f_8(2) + \mathsf{eq}\big((f_8(2)+1)^2, 2+1\big) = 1 + \mathsf{eq}(4,3) = 1 + 0 = 1\\
&f_8(4) = f_8(3+1) = f_8(3) + \mathsf{eq}\big((f_8(3)+1)^2, 3+1\big) = 1 + \mathsf{eq}(4,4) = 1 + 1 = 2
\end{aligned}
$$

Dus

$$f_8(4) = 2$$

In feite is $f_8 = \mathsf{sqrt}$, de functie uit opgave 3.

(b) Schrijf f_8 als

$$f_8 = \textbf{primrec}(g, h)$$

en geef g en h als compositie van functies op pagina 8 van dit tentamen.

We hebben

$$
\begin{aligned}
g() &= 0\\
h(y,w) &= w + \mathsf{eq}\big((w+1)^2, y+1\big) = \mathsf{add}\big(w, \mathsf{eq}\big(\mathsf{exp}(s(w),2), s(y)\big)\big)
\end{aligned}
$$

Als compositie geschreven is dit

$$
\begin{aligned}
g &= c_0^{(0)}\\
h &= \mathsf{add} \circ \big(p_2^{(2)}, \mathsf{eq} \circ \big(\mathsf{exp} \circ (s \circ p_2^{(2)}, c_2^{(2)}), s \circ p_1^{(2)}\big)\big)
\end{aligned}
$$

9. We definiëren de numeriek functie digits als

$$\mathsf{digits}(x) = \text{het Gödel-getal van het rijtje van de cijfers van } x$$

Er geldt bijvoorbeeld

$$
\begin{aligned}
\mathsf{digits}(0) &= \mathsf{gn}_0(0) = 2^{0+1} = 2\\
\mathsf{digits}(1) &= \mathsf{gn}_0(1) = 2^{1+1} = 4\\
\mathsf{digits}(10) &= \mathsf{gn}_1(1,0) = 2^{1+1}3^{0+1} = 12\\
\mathsf{digits}(42) &= \mathsf{gn}_1(4,2) = 2^{4+1}3^{2+1} = 864\\
\mathsf{digits}(666) &= \mathsf{gn}_2(6,6,6) = 2^{6+1}3^{6+1}5^{6+1} = 21870000000
\end{aligned}
$$

Laat zien dat digits een μ-recursieve functie is. (Het is zelfs een primitief recursieve functie, maar dat hoef je niet te laten zien.)

Hint: Definieer eerst een functie f_9 die een Gödel-getal dat een output is van digits terugrekent naar het oorspronkelijke getal, en laat zien dat deze functie primitief recursief is. Definieer vervolgens digits in termen van deze functie. In beide definities kunnen de functies gdln en gdn nuttig zijn.

Het gevraagde volgt uit het feit dat we de functies f_9 en digits kunnen schrijven als

$$f_9(x) = \sum_{i=0}^{\mathsf{pred}(\mathsf{gdln}(x))} \mathsf{dec}(i, x) \cdot 10^{\mathsf{pred}(\mathsf{gdln}(x)) \dot{-} i}$$

$$\mathsf{digits}(n) = \mu x. \big(\mathsf{gdn}(x) \cdot \mathsf{eq}(f_9(x), n)\big)$$

We laten nu nog zien dat digits zelfs primitief recursief is. Dit hoort natuurlijk niet meer bij het antwoord van de opgave. Om dit te laten zien moeten we een bovengrens voor de minimalisatie geven, dus we moeten afschatten wat $\mathsf{digits}(n)$ maximaal kan zijn.

Het aantal cijfers van een getal n is maximaal $n+1$ (dat is een erg ruime afschatting, maar dat geeft niet), en het maximum van een cijfer is natuurlijk negen. Hieruit volgt dat we digits ook hadden kunnen definiëren als

$$\mathsf{digits}(n) = \mu x \leq \prod_{i=0}^{n} \mathsf{pn}(i)^{9+1}. \big(\mathsf{gdn}(x) \cdot \mathsf{eq}(f_9(x), n)\big)$$

Omdat de minimalisatie nu begrensd is, volgt uit deze schrijfwijze dat digits primitief recursief is.

Macro's voor Turing-machines voor numerieke berekeningen

S	$\underset{\uparrow}{B}\,\overline{n}\quad BB$	$\underset{\uparrow}{B}\,\overline{n+1}\,B$
Z	$\underset{\uparrow}{B}\,\overline{n}\quad B$	$\underset{\uparrow}{B}\,\overline{0}\ldots B$
E	$\underset{\uparrow}{B}\,\overline{n}\,B$	*termineert niet*
$\mathrm{P}_i^{(k)}$	$\underset{\uparrow}{B}\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B$	$\underset{\uparrow}{B}\,\overline{n_i}\;B\ldots\quad\ldots B$
$\mathrm{C}_m^{(k)}$	$\underset{\uparrow}{B}\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B$	$\underset{\uparrow}{B}\,\overline{m}\;B\ldots\quad\ldots B$
A	$\underset{\uparrow}{B}\,\overline{n}\,B\,\overline{m}\quad B$	$\underset{\uparrow}{B}\,\overline{n+m}\,BB$
D	$\underset{\uparrow}{B}\,\overline{n}\qquad B$	$\underset{\uparrow}{B}\,\overline{n\dot{-}1}\ldots B$
SUB	$\underset{\uparrow}{B}\,\overline{n}\,B\,\overline{m}\quad B$	$\underset{\uparrow}{B}\,\overline{n\dot{-}m}\ldots B$
MULT	$\underset{\uparrow}{B}\,\overline{n}\,B\,\overline{m}\,B\ldots$	$\underset{\uparrow}{B}\,\overline{n\cdot m}\,B\ldots$
MR_k	$\underset{\uparrow}{B}\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B$	$B\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,\underset{\uparrow}{B}$
ML_k	$B\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,\underset{\uparrow}{B}$	$\underset{\uparrow}{B}\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B$
FR	$\underset{\uparrow}{B}B^i\,\overline{n}\,B\qquad(i\geq 0)$	$B^i\underset{\uparrow}{B}\,\overline{n}\,B$
FL	$B\,\overline{n}\,B^i\underset{\uparrow}{B}\qquad(i\geq 0)$	$\underset{\uparrow}{B}\,\overline{n}\,B^iB$
E_k	$\underset{\uparrow}{B}\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B$	$\underset{\uparrow}{B}\ldots\qquad\ldots B$
T	$\underset{\uparrow}{B}B^i\,\overline{n}\,B\qquad(i\geq 0)$	$\underset{\uparrow}{B}\,\overline{n}\,B^iB$
INT	$\underset{\uparrow}{B}\,\overline{n}\,B\,\overline{m}\,B$	$\underset{\uparrow}{B}\,\overline{m}\,B\,\overline{n}\,B$
CPY_k	$\underset{\uparrow}{B}\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B\ldots\qquad\ldots B$	$\underset{\uparrow}{B}\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B$
$\mathrm{CPY}_{k,i}$	$\underset{\uparrow}{B}\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B\,\overline{n_{k+1}}\,B\ldots B\,\overline{n_{k+i}}\,B\ldots\qquad\ldots B$	$\underset{\uparrow}{B}\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B\,\overline{n_{k+1}}\,B\ldots B\,\overline{n_{k+i}}\,B\,\overline{n_1}\,B\ldots B\,\overline{n_k}\,B$
BRN ($n=0$ / $n>0$)	$\underset{\uparrow}{B}\,\overline{n}\,B$	$\underset{\uparrow}{B}\,\overline{n}\,B$

Codering van transities

Symbol	Encoding
0	1
1	11
B	111
q_0	1
q_1	11
$\vdots$	$\vdots$
q_n	1^{n+1}
L	1
R	11

Let $en(x)$ denote the encoding of a symbol x. A transition $\delta(q_i, x) = [q_j, y, d]$ is encoded by the string

$$en(q_i)0en(x)0en(q_j)0en(y)0en(d).$$

Primitief recursieve functies

$$\begin{aligned}
\mathsf{id}(x) &= x \\
z(x) &= 0 \\
s(x) &= x + 1 \\
p_i^{(k)}(x_1, \ldots, x_k) &= x_i \\
c_n^{(k)}(x_1, \ldots, x_k) &= n
\end{aligned}$$

$$\begin{aligned}
\mathsf{pred}(y) &= y \mathbin{\dot{-}} 1 \\
\mathsf{add}(x,y) &= x + y \\
\mathsf{mult}(x,y) &= x \cdot y \\
\mathsf{sub}(x,y) &= x \mathbin{\dot{-}} y \\
\mathsf{exp}(x,y) &= x^y \\
\mathsf{fact}(x) &= x! \\
\mathsf{sg}(x) &= \text{als } x \neq 0 \text{ dan 1 anders 0} \\
\mathsf{cosg}(x) &= \text{als } x \neq 0 \text{ dan 0 anders 1} \\
\mathsf{lt}(x,y) &= \text{als } x < y \text{ dan 1 anders 0} \\
\mathsf{gt}(x,y) &= \text{als } x > y \text{ dan 1 anders 0} \\
\mathsf{le}(x,y) &= \text{als } x \leq y \text{ dan 1 anders 0} \\
\mathsf{ge}(x,y) &= \text{als } x \geq y \text{ dan 1 anders 0}
\end{aligned}$$

$$\begin{aligned}
\mathsf{eq}(x,y) &= \text{als } x = y \text{ dan 1 anders 0} \\
\mathsf{ne}(x,y) &= \text{als } x \neq y \text{ dan 1 anders 0} \\
\mathsf{max}(x,y) &= \text{het maximum van } x \text{ en } y \\
\mathsf{min}(x,y) &= \text{het minimum van } x \text{ en } y \\
\mathsf{quo}(x,y) &= \text{als } y \neq 0 \text{ dan } \lfloor x/y \rfloor \text{ anders 0} \\
\mathsf{rem}(x,y) &= \text{als } y \neq 0 \text{ dan } x \bmod y \text{ anders } x \\
\mathsf{divides}(x,y) &= \text{als } y \neq 0 \text{ en } y \mid x \text{ dan 1 anders 0} \\
\mathsf{even}(x) &= \text{als } x \text{ even is dan 1 anders 0} \\
\mathsf{prime}(x) &= \text{als } x \text{ priem is dan 1 anders 0} \\
\mathsf{pn}(x) &= \text{het } x\text{-de priemgetal} \\
& \text{(dus } \mathsf{pn}(0) = 2,\ \mathsf{pn}(1) = 3, \text{ etc.)}
\end{aligned}$$

$$\begin{aligned}
\mathsf{gn}_n(x_0, \ldots, x_n) &= \mathsf{pn}(0)^{x_0+1}\mathsf{pn}(1)^{x_1+1} \cdots \mathsf{pn}(n)^{x_n+1} = \text{Gödel-getal van het rijtje } \langle x_0, \ldots, x_n \rangle \\
\mathsf{dec}(i,x) &= i\text{-de element in het rijtje bij het Gödel-getal } x
\end{aligned}$$

$$\mathsf{gdn}(x) = \begin{cases} 1 & \text{als } x \text{ het Gödel-getal is van een rijtje} \\ 0 & \text{als } x \text{ geen Gödel-getal is} \end{cases}$$

$$\mathsf{gdln}(x) = \begin{cases} n & \text{als } x \text{ het Gödel-getal is van een rijtje met lengte } n \\ 0 & \text{als } x \text{ geen Gödel-getal is} \end{cases}$$