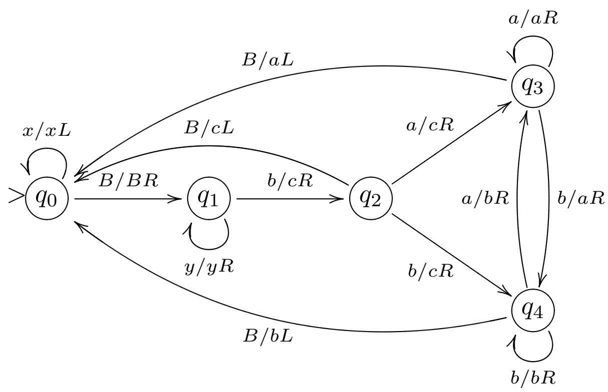
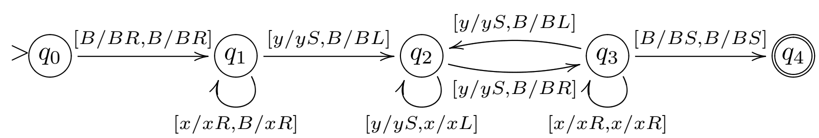
Berekenbaarheid 2010, uitwerkingen tentamen

1.



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

2.



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

3. $\Rightarrow$: Een Turing machine met unieke eindtoestand herkent een recursief opsombare taal.

   Het is een speciaal geval van een standaard Turing machine en recursief opsombare talen zijn per definitie de talen die door een standaard Turing machine worden geaccepteerd.

   $\Leftarrow$: Gegeven een recursief opsombare taal. We moeten laten zien dat er een Turing machine met unieke eindtoestand bestaat die die taal herkent.

   Per definitie is er een standaard Turing machine M (met niet noodzakelijk unieke eindtoestand) die de taal herkent. We maken bij deze M een Turing machine M' met unieke eindtoestand die dezelfde taal herkent.

   M' heeft een extra toestand q_f, en transities x/xR van iedere voormalige eindtoestand naar q_f als er nog geen transitie met x vanuit deze toestand bestond, en de enige eindtoestand van M' is q_f.

   Het is duidelijk dat M' precies dan in q_f stopt als M in een eindtoestand gestopt zou zijn, en dus geldt $L(M') = L(M)$.

Een wiskundige beschrijving van M' in termen van M:
Laat

$$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$$

Kies een q_f niet in Q, en definieer δ' door:

$$\begin{array}{rcll} \delta'(q_i, x) & = & \delta(q_i, x) & \text{als } q_i \in Q \setminus F \text{ of } \delta(q_i, x)\downarrow \\ \delta'(q_i, x) & = & [q_f, x, R] & \text{als } q_i \in F \text{ en } \delta(q_i, x)\uparrow \\ \delta'(q_f, x) & = & \uparrow & \end{array}$$

Dan nemen we

$$M' = (Q \cup \{q_f\}, \Sigma, \Gamma, \delta', q_0, \{q_f\})$$

4. L_H is recursief opsombaar, want wordt herkend door stoppen de universele Turing machine, maar is niet recursief want anders was het Halting probleem beslisbaar.

   $\overline{L_H}$ is niet recursief opsombaar, want als een taal en zijn complement allebei recursief opsombaar zijn is die taal ook recursief, en L_H was niet recursief. Hij is daarom ook niet recursief want dat is nog sterker.

5. Dit volgt uit het feit dat het blank tape probleem naar dit probleem reduceert:

   Laat gegeven een machine M waarvoor we willen weten of $M(\lambda)\downarrow$. Definieer uit M een machine M' die

   1. zijn input wist
   2. M uitvoert
   3. als M termineert zijn output wist

   Dan geldt duidelijk dat

   $$\begin{array}{llll} \text{voor alle } w \in \Sigma^* \text{ geldt dat} & M'(w) = \lambda & \text{als} & M(\lambda)\downarrow \\ \text{voor alle } w \in \Sigma^* \text{ geldt dat} & M'(w)\uparrow & \text{als} & M(\lambda)\uparrow \end{array}$$

   Dit betekent dat het probleem uit de opgave met input $R(M')$ zou beslissen of M stopt met de lege tape.

   De transformatie van $R(M)$ naar $R(M')$ is duidelijk eenvoudig genoeg om door een Turing machine gedaan te worden.

   (We stappen in deze uitwerking over de subtiliteit heen dat M' moet weten hoeveel er van de output tape is gebruikt om dit allemaal te kunnen wissen. Eigenlijk moeten we daarom met een # aan het eind van de tape bijhouden tot hoever we gekomen zijn, en M overeenkomstig moeten verbouwen.)

6.
$$
\begin{aligned}
f_1(x) &= x+1 \\
f_2(x) &= 2x
\end{aligned}
$$

Ofwel:

$$
\begin{aligned}
f_1 &= s \\
f_2 &= \mathsf{mult} \circ (c_2^{(1)}, \mathsf{id})
\end{aligned}
$$

7.
$$
\begin{aligned}
f(x_1, \ldots, x_k, 0) &= g(x_1, \ldots, x_k) \\
f(x_1, \ldots, x_k, y+1) &= h(x_1, \ldots, x_k, y, f(x_1, \ldots, x_k, y))
\end{aligned}
$$

De ariteit van g is k en die van h is $k+2$.

8. Definieer p_1 en p_2 als

$$
\begin{aligned}
p_1(x) &= x \dot{-} \mu z \leq x.\, \mathsf{divides}(x, \mathsf{exp}(2, x \dot{-} z)) \\
p_2(x) &= x \dot{-} \mu z \leq x.\, \mathsf{divides}(x, \mathsf{exp}(3, x \dot{-} z))
\end{aligned}
$$

Dan is het duidelijk dat deze functies de machten van 2 en 3 in de priemfactorontbinding van hun argument berekenen, en op grond van de vorm van de definitie is het duidelijk dat ze primitief recursief zijn.