# Berekenbaarheid 2017
# Uitwerkingen Toets 1
# 22 september 2017

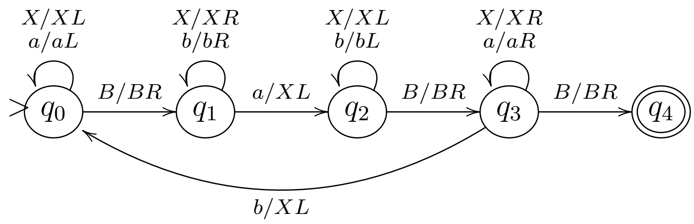
1. Definieer een standaard Turing-machine $M_1$ die de taal (2½ punten)

$$L_1 = \{w \in \{a,b\}^* \mid w \text{ bevat meer } a\text{'s dan } b\text{'s}\}$$

herkent door eindtoestand. Er geldt bijv. $aba \in L_1$ maar $abba \notin L_1$.

We gebruiken een hulpsymbool $X$, dus als tape alfabet $\Gamma = \{B, a, b, X\}$.

$M_1$:

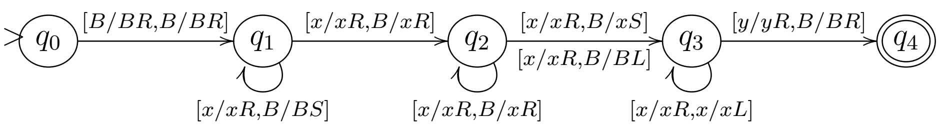



2. Definieer een non-deterministische Turing-machine $M_2$ die de taal (2½ punten)

$$L_2 = \{w \in \{a,b\}^* \mid w \text{ bevat een deelwoord } u \text{ met } |u| \geq 3 \text{ en } u = u^R\}$$

herkent door eindtoestand. Hierin is $|u|$ de lengte van het woord $u$, en $u^R$ is het woord $u$ achterstevoren. Er geldt bijv. $babbaa \in L_2$ want $babbaa$ bevat $abba$ als deelwoord, en $abba$ is een palindroom van tenminste drie symbolen.

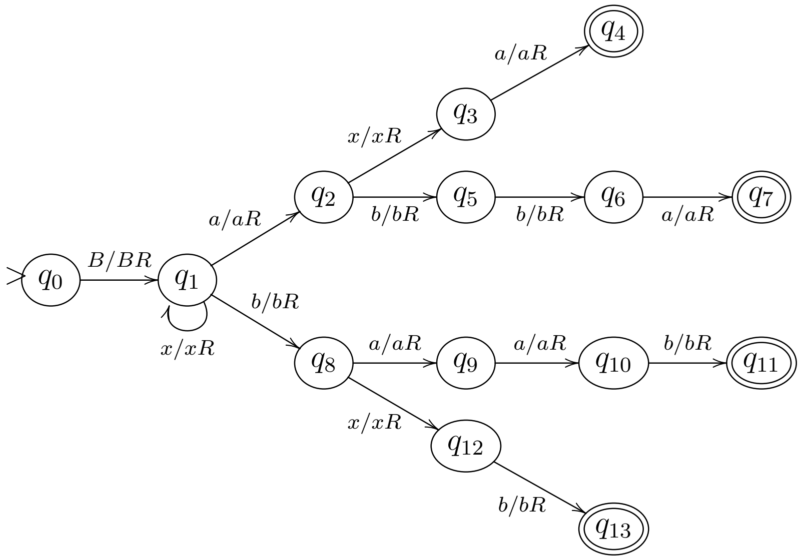
Zorg ervoor dat een input $w \in L_2$ wordt herkend in hoogstens $|w|+3$ stappen. Je mag zelf weten of je $M_2$ met één of meerdere tapes definieert.




Hierin is $x \in \Sigma$ en $y \in \Gamma$. Deze machine gebruikt hoogstens $|w| + 2$ stappen, want bij iedere stap behalve de eerste en de laatste ga je bij een symbool uit je input naar rechts.

Een leuke alternatieve oplossing gebruikt dat zo'n $u$ ook altijd wel te vinden is met $|u| = 3$ of $|u| = 4$ (neem 'het midden' van de $u$ waarvan gegarandeerd wordt dat deze bestaat). En verder is $aaa$ een deelwoord van $aaaa$ en $bbb$

van $bbbb$, waaruit volgt dat het voldoende is te testen op een deelwoord uit $\{aaa, bbb, aba, bab, abba, baab\}$:




Opnieuw met $x \in \Sigma$. Deze machine gebruikt hoogstens $|w| + 1$ stappen.

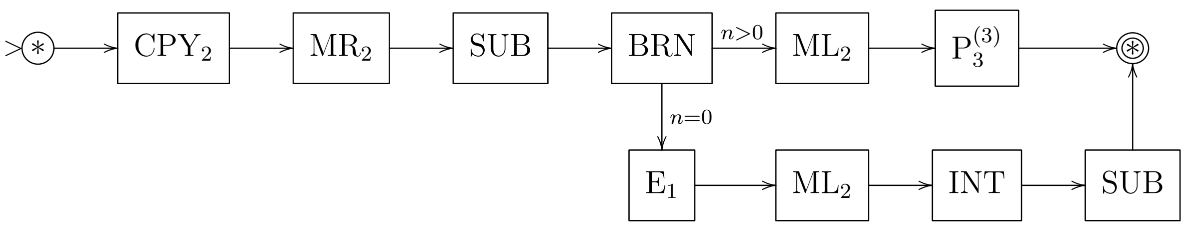
Wat verder nadenken leert overigens dat $L_2$ het complement is van

$$\{a, b, aa, bb, ab, ba, aab, bba, abb, baa, aabb, bbaa\}$$

3. Definieer een numerieke Turing-machine $M_3$ die de functie (2 punten)

$$f_3(n, m) = |n - m| = \begin{cases} n - m & \text{als } n \geq m \\ m - n & \text{als } n < m \end{cases}$$

uitrekent. Er geldt bijv. $f_3(4, 5) = |4 - 5| = |-1| = 1$. Je mag in de definitie van $M_3$ de macro's op de achterkant van dit blaadje gebruiken.




4. Gegeven een recursief opsombare taal $L_4$. Laat zien dat deze ook altijd kan worden herkend door een standaard Turing-machine $M_4$ *zonder hulpsymbolen*, dus een machine waarin $\Gamma = \Sigma \cup \{B\}$. (2 punten)

*Hint:* de constructie van een equivalente één-spoor Turing machine $M'$ uit een meer-sporen Turing-machine $M$ (zoals op het college behandeld) gebruikt geen extra hulpsymbolen, dus $M'$ en $M$ hebben daarbij dezelfde $\Gamma$.

Per definitie wordt een recursief opsombare taal herkend door een standaard Turing-machine, dus voor deze machine is er een standaard Turing-machine $M$ met $L_4 = L(M)$. Evenwel kan $M$ hulpsymbolen hebben, dus $\Gamma \supsetneq \Sigma \cup \{B\}$.

Maak nu een Turing-machine $M'$ met $n + 1$ sporen, waarbij $n$ het aantal hulpsymbolen van $M$ is, en waarbij de sporen nummer 2 tot en met $n + 1$ met de hulpsymbolen corresponderen. Kies een vast symbool $a \in \Sigma$, en codeer ieder hulpsymbool door een $a$ op het bijbehorende spoor te zetten en de rest van die positie op de tape blank te laten. Op deze manier kunnen we $M$ met $M'$ simuleren zonder hulpsymbolen (maar met extra sporen).

Volgens het college bestaat er dan ook een standaard Turing-machine $M_4$ met hetzelfde tape alfabet als $M'$, die dezelfde taal als $M'$ accepteert, dus met $L_4 = L(M) = L(M') = L(M_4)$. Dit is de machine waar in de opgave om wordt gevraagd.