Antwoorden Languages & Machines (Oefening 2)

1. Originele grammatica G (uit opgave 4.4 van het boek):

$$G = \begin{cases} S & \to & A\,B \mid B\,C\,S \\ A & \to & a\,A \mid C \\ B & \to & b\,B \mid \lambda \\ C & \to & c\,C \mid \lambda \end{cases}$$

G_1: (nieuw startsymbool, S_0, niet recursief maken)

$$G_1 = \begin{cases} S_0 & \to & A\,B \mid B\,C\,S \\ S & \to & A\,B \mid B\,C\,S \\ A & \to & a\,A \mid C \\ B & \to & b\,B \mid \lambda \\ C & \to & c\,C \mid \lambda \end{cases}$$

G_2: (non-contracting maken door null-regels te elimineren).
De null regels zijn: $\{C, B, A, S, S_0\}$

$$G_2 = \begin{cases} S_0 & \to & A \mid B \mid C \mid S \mid A\,B \mid B\,C \mid B\,S \mid C\,S \mid B\,C\,S \mid \lambda \\ S & \to & A \mid B \mid C \mid S \mid A\,B \mid B\,C \mid B\,S \mid C\,S \mid B\,C\,S \\ A & \to & a\,A \mid a \mid C \\ B & \to & b\,B \mid b \\ C & \to & c\,C \mid c \end{cases}$$

G_3: (chain rules elimineren).
De niet-triviale kettingen zien er als volgt uit:

$$\begin{array}{rcl} chain(S) & = & \{A, B, C, S\} \\ chain(S_0) & = & \{A, B, C, S, S_0\} \end{array}$$

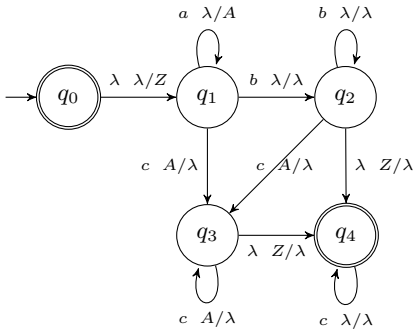
$$G_3 = \begin{cases} S_0 & \to & a\,A \mid a \mid c\,C \mid c \mid b\,B \mid b \mid A\,B \mid B\,C \mid B\,S \mid C\,S \mid B\,C\,S \mid \lambda \\ S & \to & a\,A \mid a \mid c\,C \mid c \mid b\,B \mid b \mid A\,B \mid B\,C \mid B\,S \mid C\,S \mid B\,C\,S \\ A & \to & a\,A \mid a \mid c\,C \mid c \\ B & \to & b\,B \mid b \\ C & \to & c\,C \mid c \end{cases}$$

G_4: (rechterkanten fatsoeneren)

$$G_4 = \begin{cases} S_0 & \to & X\,A \mid a \mid Z\,C \mid c \mid Y\,B \mid b \mid A\,B \mid B\,C \mid B\,S \mid C\,S \mid U\,S \mid \lambda \\ S & \to & X\,A \mid a \mid Z\,C \mid c \mid Y\,B \mid b \mid A\,B \mid B\,C \mid B\,S \mid C\,S \mid U\,S \\ A & \to & X\,A \mid a \mid Z\,C \mid c \\ B & \to & Y\,B \mid b \\ C & \to & Z\,C \mid c \\ X & \to & a \\ Y & \to & b \\ Z & \to & c \\ U & \to & B\,C \end{cases}$$

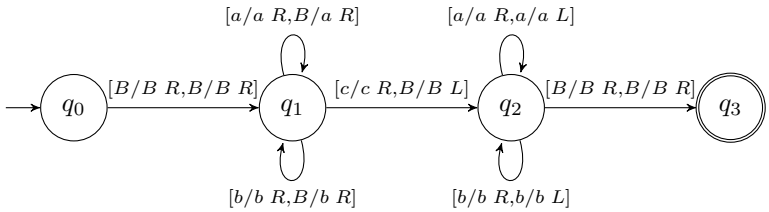
2. $L = \{a^i\, b^*\, c^j \mid j \geq i \geq 0\}$

   (a) Een deterministische PDA voor L:



   Er wordt eerst een Z op de stack gezet, om later te testen dat de stack (bijna) leeg is. In q_1 worden a's gelezen, en geteld op de stack. Dan kan er een willekeurig aantal b's volgen, die verder niet hoeven te worden geteld. Na de eerste c vanaf q_1 of q_2 springen we naar q_3. Hier worden net zoveel c's gelezen als er a's waren. Als de stack leeg is springen we naar q_4. Daar mogen nog meer c's volgen. De automaat is deterministisch, omdat er vanuit geen enkele toestand overlappende transities mogelijk zijn.

3. (a)



   (b) Er wordt eerst een maximaal woord $w \in \{a, b\}^*$ van tape 1 naar tape 2 gekopieerd. Na een c wordt gecheckt of het resterende woord w^R op tape 1 matcht met het omgekeerde woord op tape 2. De berekening is:

$[q_0; *BaabcbaaB; *BBBBB]$
$[q_1; B * aabcbaaB; B * BBBB]$
$[q_1; Ba * abcbaaB; Ba * BBB]$
$[q_1; Baa * bcbaaB; Baa * BB]$
$[q_1; Baab * cbaaB; Baab * B]$
$[q_2; Baabc * baaB; Baa * bB]$
$[q_2; Baabcb * aaB; Ba * abB]$
$[q_2; Baabcba * aB; B * aabB]$
$[q_2; Baabcbaa * B; *BaabB]$
$[q_3; BaabcbaaB*; B * aabB]$

De TM termineert in de accepterende toestand q_3, dus het *aabcbaa* wordt geaccepteerd.

(c) Deze TM termineert altijd (na 1 pass over het woord op tape 1), dus hij beslist deze taal.

(d) Deze TM is deterministisch, want het symbool op tape 1 bepaalt uniek welke transitie wordt genomen.