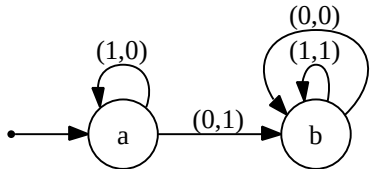
Aufgabe 1.: *Mealy–Automaten*

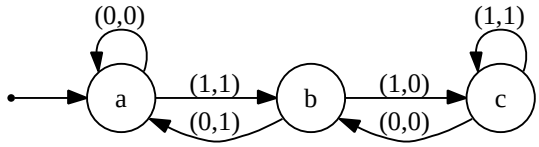
Frage: Warum gibt es zwei Funktionen? δ und λ sind doch nicht unabhängig!? Handelt es sich nicht vielmehr um die selbe Funktion mit $\delta_\lambda : Z \times \Sigma \to Z \times \Delta$?

Hinweis: Die Pfeilbeschriftung in den Graphen bedeutet (Eingabe,Ausgabe).

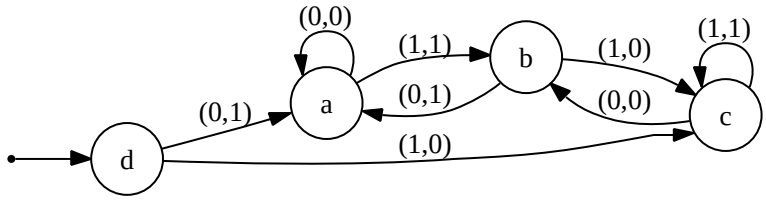
a) Berechnet die Funktion $f : rbin(x) \mapsto rbin(x+1)$
Der Automat sucht in der Binärfolge die erste 0 (Null) und ersetzt diese durch eine 1 (Eins). Die Einsen vor der ersten Null werden zu Nullen, die Bits nach der Null bleiben bestehen.
$M = (\{a, b\}, \{0, 1\}, \{0, 1\}, \delta, \lambda, a)$



b) Berechnet die Funktion $f : rbin(x) \mapsto rbin(3 \cdot x)$
Dieser Automat muss zwei Übertragszustände speichern können. Die Multiplikation mit 3 lässt sich in der Binärdarstellung gut als $2 \cdot x + x$ nachvollziehen. Die Multiplikation ist ein „Bitshift“ richtung MSB, die Addition überträgt im Fall von 1+1 auf die nächste Stelle. Kommen also zwei Einsen in Folge, verändern sie bei einer Multiplikation mit 3 die übernächste Stelle, deswegen 2 Übertragszustände.
$M = (\{a, b, c\}, \{0, 1\}, \{0, 1\}, \delta, \lambda, a)$



c) Berechnet die Funktion $f : rbin(x) \mapsto rbin(3 \cdot x + 1)$
Dieser Automat unterscheidet sich zum Automat aus b) nur durch das Lesen des ersten Bits.
$M = (\{a, b, c, d\}, \{0, 1\}, \{0, 1\}, \delta, \lambda, d)$



Aufgabe 2.
Wenn L das leere Wort ϵ enthält ist $MIN(L) = \emptyset$, und eine leere Sprache ist immer kontextfrei. Wenn $\epsilon \notin L$ und L *deterministisch* kontextfrei ist, gibt es einen deterministischen Kellerautomaten M_L *mit Endzuständen*, welcher L erkennt. Wenn der Kellerautomat nun so modifiziert wird, dass jeder Endzuständ *ausschließlich* als letzter Zustand erreicht werden kann, so werden keine Wörter mehr akzeptiert, welche ein Präfix aus L enthalten. Hierfür müssen aus δ alle Übergänge entfernt werden, die von einem Endzustand abgeleitet werden. Es gilt also:

$$M_L = (Z, \Sigma, \Gamma, \delta, z_0, \#, E)$$

$$M_{MIN(L)} = (Z, \Sigma, \Gamma, \delta', z_0, \#, E)$$

Die neue Übergangsfunktion δ' ist gleich δ, bis auf folgende Ableitungen:

$$\forall z \in E, \forall a \in \Sigma, \forall A \in \Gamma : \delta'(z, a, A) = \emptyset$$

Erreicht der Automat somit einen Endzustand, ohne das Wort vollständig gelesen zu haben (sondern eben nur einen Präfix), so blockiert er und akzeptiert das Wort nicht. Da der Automat bis auf die Reduzierung der Deltafunktion nicht verändert wurde, ist er (und somit $MIN(L)$) natürlich immer noch *deterministisch kontextfrei*. q.e.d.

Für die Sprache

$$L = \{a^i b^j c^k | i, j, k > 0, k \geq i \vee k \geq j\}$$

ist das „präfixfreie“ Minimum $MIN(L)$ die Sprache

$$MIN(L) = \{a^i b^j c^k | k > 0, k = min(i, j)\}$$

Die die Sprache $MIN(L)$ *nicht* kontextfrei ist, kann gemäß dem oben bewießenen Satz die Sprache L nicht deterministisch kontextfrei sein.

Aufgabe 3.
Grammatik für korrekt geklammerte Ausdrücke in bison–Syntax:

```
S:   '(' ')'
   | '[' ']'
   | '(' S ')'
   | '[' S ']'
   | S S
  ;
```

Die Datei `thomas-B9A3.tar.gz` enthält drei `yacc`–Programme. Eines für Teilaufgabe a) und zwei für Teilaufgabe b). Die Aufgabenstellung forderte, dass für b) keine `shift/reduce`–Konflikte auftreten, da sonst die Klammertiefe nicht ermittelt werden könne. Ich habe eine Version mit, und eine ohne Konflikt geschrieben, und beide können die Klammertiefe korrekt ermitteln (glaube ich zumindest)!