

Theoretische Informatik: Selbststudium - Blatt 1

Abgabe bis 23. Oktober 2015
Assistent: Sacha Krug, CHN D 42

Linus Fessler, Markus Hauptner, Philipp Schimmelfennig

Aufgabe S1

- (a) Vermutung: $L(G_1) = \{avawa \mid v \in \{b\}^+, w \in \{ab\}^*\}$

Beweis: Die Grammatik beginnt im Startsymbol S , für das es nur eine Ableitungsregel gibt: $S \rightarrow aXaYa$. Da das Wort $aXaYa$ immer noch die Nichtterminale X und Y enthält, müssen wir weiter ableiten. Es gibt in P_1 keine Ableitungsregel $QaR \rightarrow_{G_1} S$ für beliebige $Q, R, S \in (\Sigma_N \cup \Sigma_T)^*$, womit jedes Wort in der von der Grammatik erzeugten Sprache die Form $avawa$ aufweisen muss, für noch zu bestimmende v und u .

Im Folgenden beweisen wir, dass $v \in \{b\}^+$ und $w \in \{ab\}^*$.

1. $v \in \{b\}^+$: Wir haben $aXaYa$ gegeben. Die Ableitungsregeln, die X betreffen, sind $\{X \rightarrow bX, X \rightarrow b\}$. Wir zeigen mit Induktion, dass man mit X und den beiden Ableitungsregeln für X ausschliesslich die Wörter $v \in \{b\}^+ = \{b^i \mid i \in \mathbb{N}\}$ generieren kann. Sei n die Anzahl angewandter Ableitungsregeln.

- (i) *Induktionsanfang:*

Sei $n = 1$. Man kann mit einer Ableitung aus X entweder das Wort bX oder b herleiten (nicht aber λ).

- (ii) *Induktionsschritt:*

$n \rightarrow n + 1$: Es gibt zwei Fälle:

Das Wort endet auf X : Man kann wieder bX oder b herleiten.

Das Wort endet auf b : Das Wort kann nicht mehr weiter abgeleitet werden und man erhält nach dem n -ten Ableitungsschritt das Wort b^n .

2. $w \in \{ab\}^*$: Wir haben $aXaYa$ gegeben. Die Ableitungsregeln, die Y betreffen, sind $\{Y \rightarrow abY, Y \rightarrow ab, Y \rightarrow \lambda\}$. Mit Induktion kann man analog zu 1. beweisen, dass man mit Y und den drei Ableitungsregeln für Y ausschliesslich die Wörter $w \in \{ab\}^* = \{(ab)^i \mid i \in \mathbb{N}_0\}$ generieren kann. Der einzige Unterschied ist, dass man durch die dritte Ableitungsregel $Y \rightarrow \lambda$ auch das leere Wort herleiten kann.

- (b) Man beginnt in S . Für S gibt es nur eine Ableitungsregel $S \rightarrow 0X0$. Wir gehen also von $0X0$ aus. Für X gibt es nun die drei Ableitungsregeln $\{X \rightarrow AX, X \rightarrow BX, X \rightarrow Y\}$. Nun können wir beliebig viele A oder B vor das X schreiben, bevor wir X durch Y ersetzen.

Damit erhalten wir das Wort $0WY0$ für $W \in \{A, B\}^*$.

Jetzt können wir sukzessive die Ableitungsregeln $\{AY \rightarrow Y0, BY \rightarrow Y1\}$ anwenden, bis keine A oder B mehr im Wort enthalten sind.

Dann können wir noch einmal die letzte Ableitungsregel $\{0Y \rightarrow 0\}$ anwenden.

So entsteht das Wort $0w0$ für $w \in \{0, 1\}^*$, denn die Anwendung von $\{AY \rightarrow Y0, BY \rightarrow Y1\}$ bewirkt schlussendlich, dass jedes A durch eine 0 und jedes B durch eine 1 ersetzt wird und $\{0Y \rightarrow 0\}$ löscht das Y am Ende.

Eine äquivalente reguläre Grammatik ist $G'_2 = (\{S, X\}, \{0, 1\}, P'_2, S)$ mit

$$P'_2 = \{S \rightarrow 0X0, X \rightarrow \lambda, X \rightarrow 0, X \rightarrow 1, X \rightarrow X0, X \rightarrow X1\}.$$

Der Entwurf ist praktisch selbsterklärend. Wir starten in S , gehen über zu $0X0$ und können dann wählen, ob wir X zu λ , 0 oder 1 ableiten oder 0 bzw. 1 schreiben und danach weiter Buchstaben aus $\{\lambda, 0, 1\}$ hinzufügen wollen oder nicht (indem wir Regel $X \rightarrow X0$ oder $X \rightarrow X1$ anwenden). Damit generiert die Grammatik G'_2 auch die Wörter $0w0$ für $w \in \{0, 1\}^*$.

Aufgabe S2

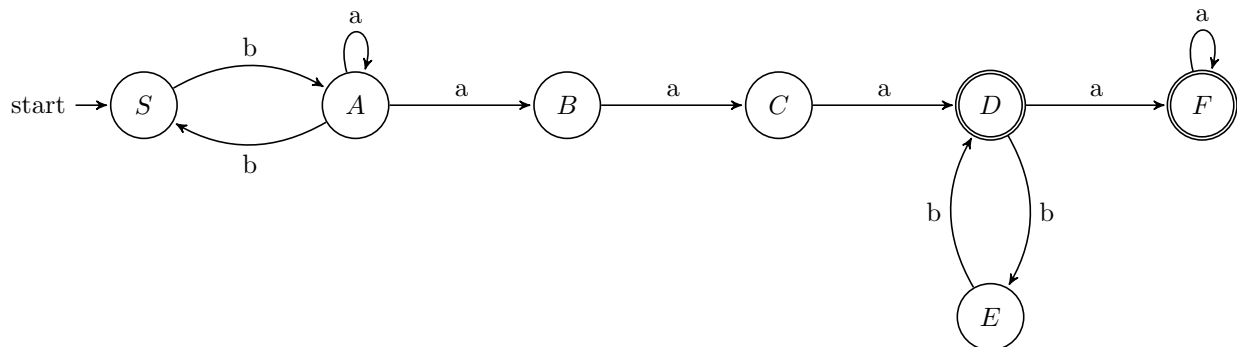
Die Sprache, die von G erzeugt wird, ist

$$\{vwaaaxy \mid v = b^{2k+1} \text{ für } k \in \mathbb{N}_0, \quad w, y \in \{a\}^*, \quad x \in \{bb\}^*\}.$$

Eine äquivalente reguläre Grammatik ist also $G' = (\{S, A, B, C, D, E, F\}, \{a, b\}, P', S)$ mit

$$P' = \{S \rightarrow bA, A \rightarrow bS, A \rightarrow aA, A \rightarrow aB, B \rightarrow aC, C \rightarrow aD, \\ D \rightarrow bE, E \rightarrow bD, D \rightarrow \lambda, D \rightarrow aF, F \rightarrow aF, F \rightarrow \lambda\}.$$

Der NEA, der die Sprache $L(G)$ erzeugt, ist damit:

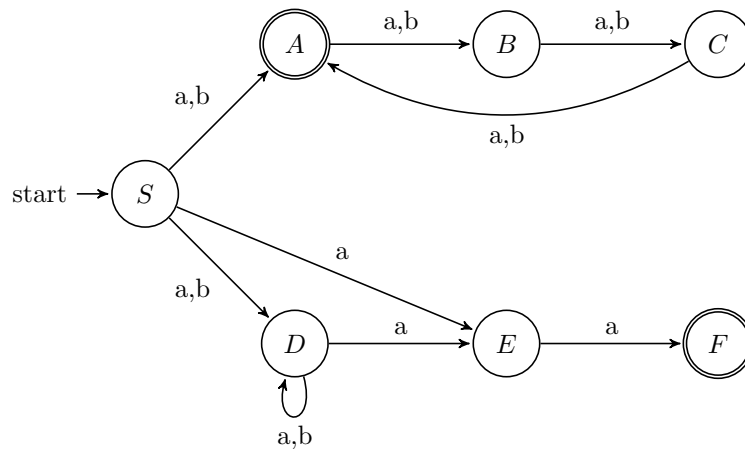


Aufgabe S3

(a) Eine reguläre Grammatik für die Sprache L_1 ist $G_1 = (\{S, A, B, C, D, E, F\}, \{a, b\}, P_1, S)$ mit

$$P_1 = \{S \rightarrow aA, S \rightarrow bA, A \rightarrow aB, A \rightarrow bB, B \rightarrow aC, B \rightarrow bC, C \rightarrow aA, C \rightarrow bA, B \rightarrow \lambda, \\ S \rightarrow aD, S \rightarrow bD, S \rightarrow aE, D \rightarrow aD, D \rightarrow bD, D \rightarrow aE, E \rightarrow aF, F \rightarrow \lambda\}.$$

Um den Entwurf zu begründen ist es einfacher, über den äquivalenten NEA zu argumentieren, der sich leicht von den Regeln ablesen lässt:

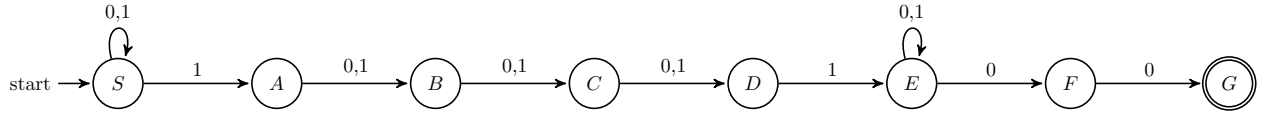


Der NEA wählt zuerst im Startknoten S , ob er die Bedingung $(|x|_a + |x|_b) \bmod 3 = 1$ überprüfen will (Knoten A, B, C) oder die Bedingung x endet mit aa (Knoten D, E, F). Dadurch wird L_1 nichtdeterministisch erzeugt und weil der NEA äquivalent zur Grammatik G_1 ist, ist $L_1 = L(G_1)$.

- (b) Eine reguläre Grammatik für die Sprache L_2 ist $G_2 = (\{S, A, B, C, D, E, F, G\}, \{0, 1\}, P_2, S)$ mit

$$P_2 = \{S \rightarrow 0S, S \rightarrow 1S, S \rightarrow 1A, A \rightarrow 0B, A \rightarrow 1B, B \rightarrow 0C, B \rightarrow 1C, C \rightarrow 0D, \\ C \rightarrow 1D, D \rightarrow 1E, E \rightarrow 0E, E \rightarrow 1E, E \rightarrow 0F, F \rightarrow 0G, G \rightarrow \lambda\}.$$

Wir begründen den Entwurf wieder über den äquivalenten NEA:



Es ist klar, dass der NEA und die Grammatik G_2 äquivalent sind und dass der NEA L_2 erzeugt. Also ist $L_2 = L(G_2)$.

- (c) Eine allgemeine Grammatik für die Sprache L_3 ist $G_3 = (\{S, A, X, Y, Z\}, \{0, 1, 2\}, P_3, S)$ mit

$$P_3 = \{S \rightarrow XYZ, Z \rightarrow A2Z, 2A \rightarrow A2, YA \rightarrow A1Y, 1A \rightarrow A1, XA \rightarrow 0X, \\ Z \rightarrow \lambda, Y \rightarrow \lambda, X \rightarrow \lambda\}.$$

Die Idee ist, dass, wenn vor dem Z eine 2 eingefügt wird, immer auch eine 1 vor dem Y und eine 0 vor dem X eingefügt wird. Dafür sorgt ein temporäres A , dass von hinten nach vorn durchgeschoben wird. Die letzten drei Regeln in P_3 sind dann dafür da, X , Y und Z aus dem Wort zu entfernen und das Wort $0^0 1^0 2^0 = \lambda$ zu ermöglichen.

Die Ableitung des Wortes 000111222 ist damit:

$$\begin{aligned} S &\Rightarrow_{G_3} XYZ \Rightarrow_{G_3} XYA2Z \Rightarrow_{G_3} XA1Y2Z \\ &\Rightarrow_{G_3} 0X1Y2Z \Rightarrow_{G_3} 0X1Y2A2Z \Rightarrow_{G_3} 0X1YA22Z \Rightarrow_{G_3} 0X1A1Y22Z \Rightarrow_{G_3} 0XA11Y22Z \\ &\Rightarrow_{G_3} 00X11Y22Z \Rightarrow_{G_3} 00X11Y22A2Z \Rightarrow_{G_3}^2 00X11YA222Z \Rightarrow_{G_3} 00X11A1Y222Z \Rightarrow_{G_3}^2 00XA111Y222Z \\ &\Rightarrow_{G_3} 000X111Y222Z \Rightarrow_{G_3} 000111222 \end{aligned}$$