

1. Betrachten Sie die Sprache $L(G_C)$, welche die einfache Deklaration und Initialisierung einer Variablen in der Programmiersprache C erzeugt, z.B.

`int aufgabe = 1;` $\in L(G_C)$,

`float pi = 3.14;` $\in L(G_C)$, und

`char * c = "Lichtgeschwindigkeit";` $\in L(G_C)$

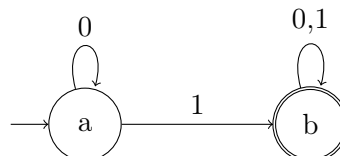
aber

`int c = 2.9;` $\notin L(G_C)$.

- Definieren Sie eine rechtslineare Grammatik G , welche zumindest den Datentyp `int` unterstützt. Beschränken Sie Variablennamen auf aus Kleinbuchstaben bestehende Zeichenketten und die Zuweisungen auf Literale vom passenden Typ.
- Beschreiben Sie, wie die Sprache um weitere Datentypen wie z.B. `float` und `char*` erweitert werden kann.
- Besteht die Möglichkeit einer rechtslinearen Grammatik G' , so dass allgemeine Rechenausdrücke zugewiesen werden können?

`int ergebnis = (4 + 6) * (27 - 18);` $\in L(G')$

- 2) Betrachten Sie folgenden endlichen Automaten A.



- Bestimmen Sie die akzeptierte Sprache $L(A)$ dieses Automaten.
 - Überprüfen Sie, mittels Induktion über die Wortlänge, Ihre Behauptung.
Hinweis: Für den Schrittfall bietet sich das Resultat aus der Zusatzübung an.
- 3) Wann nennt man eine formale Sprache *regulär*? Wie ist die *Sprache eines Automaten* definiert?

Gegeben sei die reguläre Sprache

$$L = \{ab^n a \mid n \text{ ungerade}\}.$$

über dem Alphabet $\Sigma = \{a, b\}$.

- a) Finden Sie eine Grammatik, welche diese Sprache erzeugt.
- b) Konstruieren Sie einen Automaten, der diese Sprache akzeptiert.

Zusatzübung. *Wie ist die erzeugte Sprache $L(M)$ eines Automaten M definiert?*

Sei $A = (Q, \Sigma, \delta, s, F)$ ein DEA. Beweisen Sie, dass

$$\hat{\delta}(q, yz) = \hat{\delta}(\hat{\delta}(q, y), z) \text{ für alle } q \in Q \text{ und } y, z \in \Sigma^*.$$

Hinweis: Verwenden Sie Induktion über die Länge von z , dh. über $|z|$.