Spracherkennung bei Klammersprachen: Kellerautomaten

a) Konstruiere einen endlichen Automaten, der die Sprache aller Klammerausdrücke bis Tiefe 2 erkennt.
Zur Sprache gehören z.B. (()), ()(), (), (()()).
Nicht zur Sprache gehören z.B. ((())), ((())), ()),)(,)()(

b) Gib einen endlichen Automaten A an, der die allgemeine Klammersprache, also Klammerungen beliebiger Tiefe erkennt.

Aufgabe 1: Die Idee des Kellerautomaten

Die folgende Darstellung zeigt einen Kellerautomaten.

a) Wie "funktioniert" der Kellerautomat?
b) Welche Funktion hat der „Stapel“ bzw. „Keller“ oder (engl.) „Stack“?
c) Woran erkennt die Verarbeitungseinheit, ob ein Klammerausdruck korrekt ist?

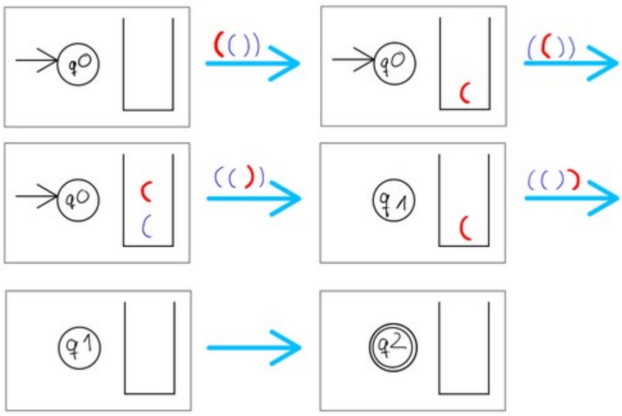


Abbildung 1: Klammerautomat der Tiefe 2 (Eigene Darstellung)

Ein **Kellerautomat** ist also eine Verarbeitungseinheit, die Symbole eines Eingabeworts verarbeitet, sich dabei stets in einem bestimmten Zustand befindet und die zum Zwischenspeichern von Symbolen einen Stapel / Keller benutzt.

Die Menge der Eingabesymbole eines Kellerautomaten kann als Alphabet einer Sprache aufgefasst werden.

Unter der **Sprache eines Kellerautomaten** versteht man die Menge aller Wörter aus Eingabesymbolen, die den Kellerautomaten vom Anfangszustand in einen Endzustand überführen.

Wenn `K` ein gegebener Kellerautomat ist, dann schreiben wir `L(K)` für die Sprache des Kellerautomaten `K`.

Ablaufprotokoll eines Kellerautomaten

Den „Lauf“ eines Kellerautomaten kann man in folgender Darstellung dokumentieren, bei der die durchlaufenen Zustände, die Entwicklung des Stack und das Verbrauchen der Eingabe sichtbar werden.

Dabei bedeutet die Notation an den Übergängen:

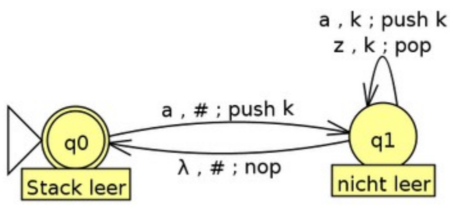
<gelesenes Zeichen> , <oberstes Zeichen auf dem Stack> ; <auszuführende Stackoperation>

Eingabe *Ausgabe/Operation*

Beispiel:

```
a, # ; push k  » in der Eingabe wird das Zeichen a gelesen
                 auf dem Stack liegt ein # (=leerer Stack)
                 → das Zeichen k wird auf den Stack gelegt
```

Für den rechts abgebildeten Automaten zur Erkennung von Klammerungen (a = Klammer auf, z = Klammer zu) und die Eingabe *aazazz* ergibt sich:



Stack ---→								
			k		k			
		k	k	k	k	k		
	#	#	#	#	#	#	#	#
Zustand	q0	q1	q1	q1	q1	q1	q1	q0
Vom Übergang verbrauchtes Eingabezeichen		a	a	z	a	z	z	Eingabe endet, q0 ist Endzustand: Wort wird akzeptiert!

Im Diagramm sieht man, wie der Stack mehrfach wächst und schrumpft. Unter der Rechtsachse wird der jeweils erreichte Zustand notiert und in der letzten Zeile das Eingabezeichen, das der jeweilige Zustandsübergang verbraucht.

Wie beim DFA gibt es einen **Fehlerzustand qF**, der im Graphen weggelassen wird, in der Tabellendarstellung sowie im Ablaufdiagramm aber vorkommt. Alle Eingabe/Stackzeichen-Kombinationen, die an einem Zustand nicht im Graphen enthalten sind, führen in den Fehlerzustand.

Übungsaufgaben

Aufgabe 1: Eine Grammatik für gültige Klammerausdrücke

Im Folgenden sollen Klammerausdrücke betrachtet werden:

`()(), (()(())), ()(())(()()), ...`

bzw. in abstrahierter Form: abab, aabaabbb, abaabbaababb, ...

Beschreibe diese Klammerausdrücke mit einer Grammatik.

Aufgabe 2[1]: Kontextfreie Sprachen?

Entscheide mit Hilfe des Kellerautomaten, ob die folgenden Sprachen kontextfrei sind. Wenn die Sprache kontextfrei ist, dann erläutere, wie Kellerautomaten arbeiten würde. Falls nicht, dann begründe, warum diese nicht mit einem Kellerautomaten erkannt werden kann.

a) $L = \{ a^n b^n \mid n \geq 0 \}$

b) $L = \{ a^n b^n c^n \mid n \geq 0 \}$

c) $L = \{ w \in \{a, b\}^* \mid \#a = \#b \}$ *die alle Wörter enthält, die aus gleich vielen a und b bestehen, die – im Gegensatz zur Sprache aus Aufgabenteil a) - in beliebiger Reihenfolge auftreten dürfen*

> *. Man kann mit zwei Zeichen im Stackalphabet arbeiten: ist ein "a" auf dem Stack und kommt ein weiteres "a", wird dieses obenauf gelegt. Folgt hingegen ein "b", wird das "a" abgebaut. Ist bei Eingabe "b" kein "a" zum abbauen vorhanden, wird das "b" auf den Stack gelegt und seinerseits von "a"s abgebaut.*

d) $L = \{ w \in \{a, b, c\}^* \mid \#a = \#b = \#c \}$

> *nicht kontextfrei - es kann nur ein Aufbau mit einem Abbau kombiniert werden. Die Sprache aus b) ist eine Teilmenge dieser Sprache - und da diese nicht kontextfrei ist kann die Sprache aus d) dies auch nicht sein.*

e) $L = \{ a^n b^m c^n \mid n, m \geq 0 \}$

Hinweis: #a bezeichnet die Anzahl der a´s in dem betreffenden Wort.

1 Aufgabenstellung entnommen Becker (Hrsg.): *inf-schule.de – Sprachen und Automaten,* CC-BY-SA 4.0

*Aufgabe 3: Abgrenzung regulär – kontextfrei – kontextsensitiv**

In Aufgabe 2 haben Sie sich mit wichtigen Beispielen aus unterschiedlichen Sprachklassen auseinandergesetzt. Entscheiden Sie, welche der folgenden Aussagen korrekt sind:

a) Immer wenn Zeichen in einem Wort gezählt werden müssen, ist die Sprache kontextfrei.

b) Immer wenn Zeichen in einem Wort gezählt werden müssen, ist die Sprache nicht regulär.

c) Immer wenn es auf die Reihenfolge der Zeichen ankommt, ist die Sprache nicht kontextfrei.

d) Ein DFA kann unendlich viele Zustände haben.

e) Ein DFA kann unendlich viele Übergänge haben.

f) Eine reguläre Sprache kann unendlich lange Wörter haben.

g) Eine reguläre Sprache kann unendlich viele Wörter haben.

h) Eine reguläre Sprache kann beliebig viele Wörter haben.

i) Eine reguläre Sprache kann beliebig lange Wörter haben.

j) Jede Sprache enthält mindestens ein Wort.

k) Jedes Wort enthält mindestens ein Alphabetzeichen.

l) Jede Sprache, die nicht regulär ist, ist kontextfrei.

m) Jede Sprache, die nicht kontextfrei ist, ist nicht regulär.

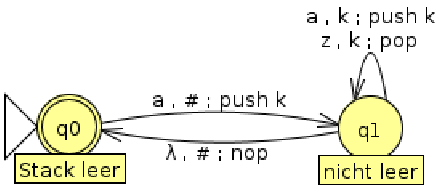
n) Jede Sprache, die kontextfrei ist, ist nicht regulär.

o) Jede Sprache ist entweder regulär oder kontextfrei.

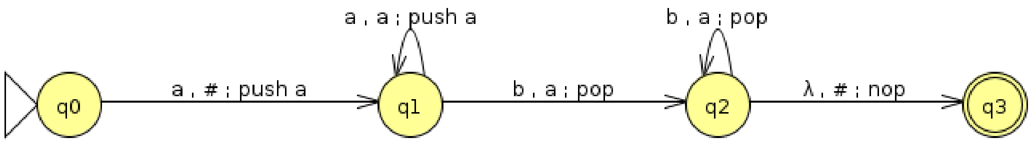
Aufgabe 4 – Definition und Ablaufprotokolle

Gegeben sind folgende Kellerautomaten. Beschreiben Sie für jeden der beiden Automaten die Wörter, die dieser erkennt und erstellen Sie jeweils Ablaufprotokolle für zwei aussagekräftige Beispieleingaben – ein Wort, das akzeptiert wird und eines, das verworfen wird.

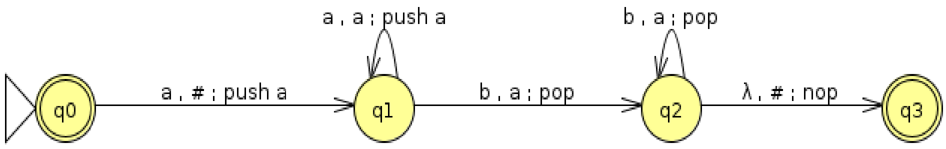
a)



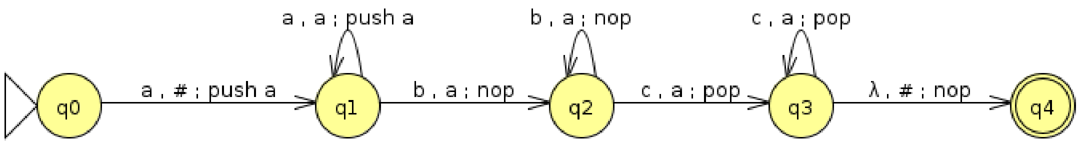
b)



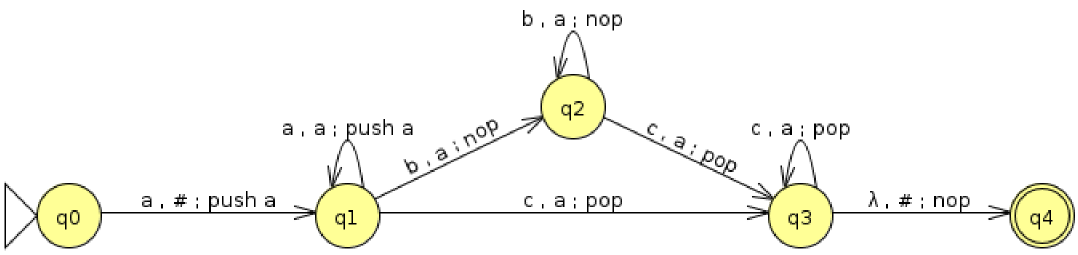
c)



d)



e)



f)