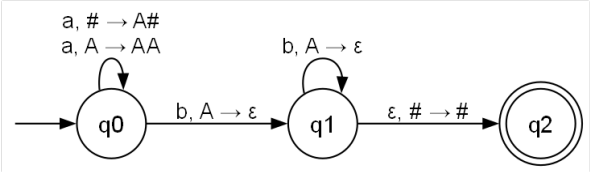
1. (4 Punkte) Der Kellerautomat akzeptiert die Sprache $L = \{a^n b^n \mid n \geq 1\}$.



(a) Gib die Abarbeitung des Wortes *aaabbb* als Folge von Konfigurationen (Zustand, Restwort, Keller) an.

(b) Was passiert bei der Eingabe *aab*?

Lösung:

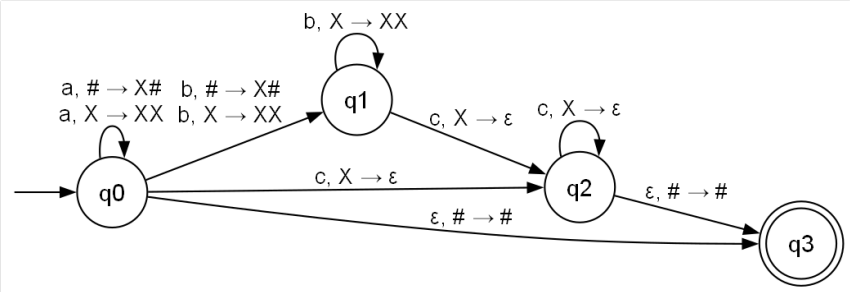
(a)

Zustand	Restwort	Keller
q_0	*aaabbb*	`#`
q_0	*aabbb*	`A#`
q_0	*abbb*	`AA#`
q_0	*bbb*	`AAA#`
q_1	*bb*	`AA#`
q_1	*b*	`A#`
q_1	ε	`#`
q_2	ε	`#`

Das Wort wird akzeptiert.

(b) Nach *aab* ist der Automat in q_1, im Keller liegt noch $A\#$. Das Wort ist gelesen, aber q_1 ist kein Endzustand, und nach q_2 kommt er nur mit # oben im Keller. *aab* wird nicht akzeptiert.

2. (5 Punkte) Gegeben ist der folgende Kellerautomat über $\Sigma = \{a, b, c\}$:



(a) Gib die Abarbeitung des Wortes *abcc* an.

(b) Welche der Wörter *acc*, *bc*, *aabccc*, *abc*, *bbcc*, *cab* werden akzeptiert?

(c) Beschreibe die Sprache des Automaten in Mengenschreibweise.

Lösung:

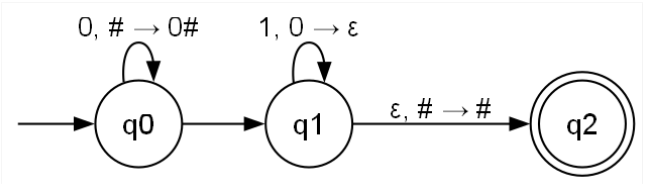
(a)

Zustand	Restwort	Keller
q_0	*abcc*	#
q_0	*bcc*	X#
q_1	*cc*	XX#
q_2	*c*	X#
q_2	ε	#
q_3	ε	#

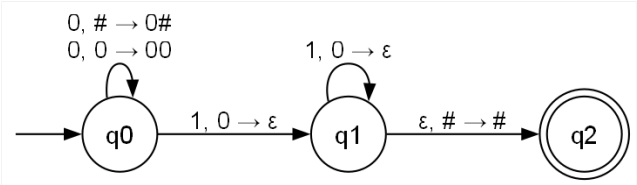
(b) *bc*, *aabccc*, *bbcc*. Bei *acc* und *abc* passt die Anzahl der c nicht, bei *cab* die Reihenfolge.

(c) $L = \{a^i b^j c^{i+j} \mid i, j \geq 0\}$. Für jedes a und jedes b liegt ein X im Keller, jedes c nimmt eins weg. Der Übergang von q_0 nach q_3 ist für das leere Wort.

3. (3 Punkte) Die Abbildung zeigt einen Kellerautomaten für die Sprache $L = \{0^n 1^n \mid n \geq 1\}$ über $\Sigma = \{0, 1\}$. Es fehlen genau zwei Übergänge. Ergänze die fehlenden Übergänge.

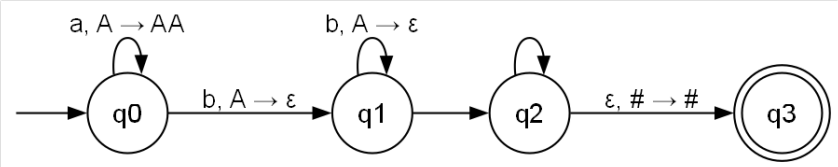


Lösung:

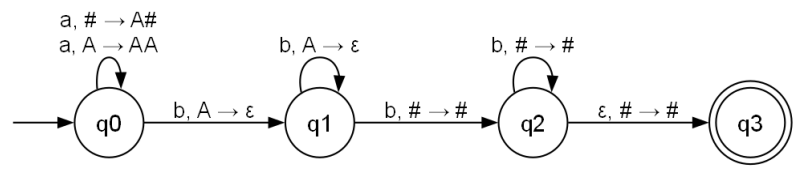


Ergänzt: $0, 0 \to 00$ bei q_0, $1, 0 \to \varepsilon$ von q_0 nach q_1.

4. (3 Punkte) Die Abbildung zeigt einen Kellerautomaten für die Sprache $L = \{a^n b^m \mid m > n,\ n \geq 1\}$ über $\Sigma = \{a, b\}$. Es fehlen genau drei Übergänge. Ergänze die fehlenden Übergänge.

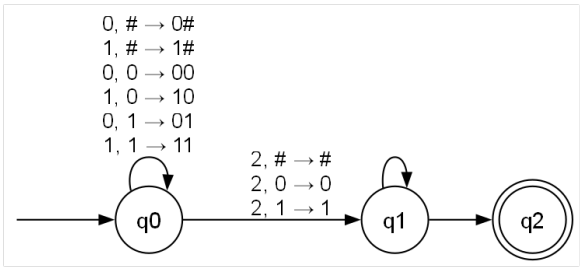


Lösung:

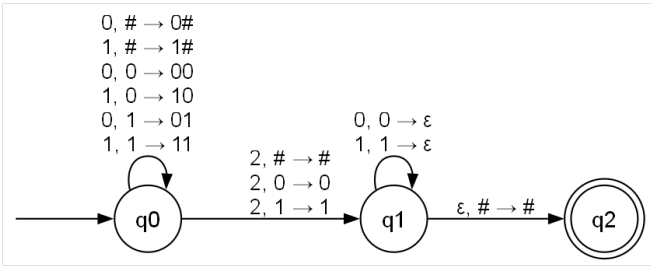


Ergänzt: $a, \# \to A\#$ bei q_0, $b, \# \to \#$ von q_1 nach q_2, $b, \# \to \#$ bei q_2.

5. (4 Punkte) Die Abbildung zeigt einen Kellerautomaten für die Sprache $L = \{w2w^R \mid w \in \{0,1\}^*\}$ über $\Sigma = \{0, 1, 2\}$. Es fehlen genau drei Übergänge. Ergänze die fehlenden Übergänge. Hinweis: w^R ist das umgedrehte Wort w, z.B. $011\,2\,110 \in L$.

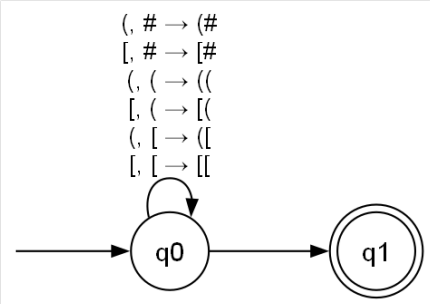


Lösung:

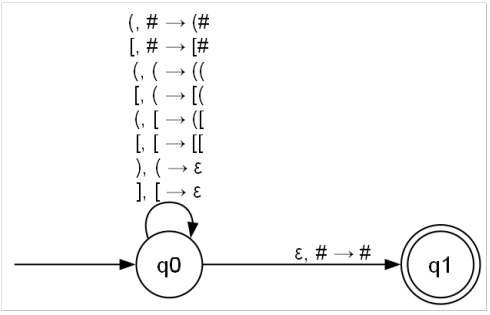


Ergänzt: $0, 0 \to \varepsilon$ bei q_1, $1, 1 \to \varepsilon$ bei q_1, $\varepsilon, \# \to \#$ von q_1 nach q_2.

6. (4 Punkte) Die Abbildung zeigt einen Kellerautomaten für die korrekt geklammerten Wörter über $\Sigma = \{(,), [,]\}$. Beispiele in der Sprache: (), ([]), [()[]]. Nicht in der Sprache: (], ([)], ((). Es fehlen genau drei Übergänge. Ergänze die fehlenden Übergänge.



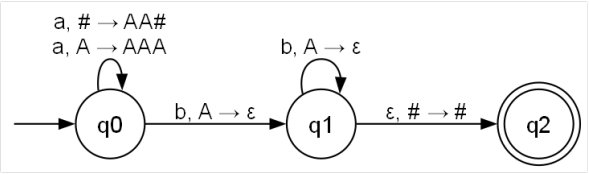
Lösung:



Ergänzt: $), (\to \varepsilon$ bei q_0, $], [\to \varepsilon$ bei q_0, $\varepsilon, \# \to \#$ von q_0 nach q_1.

7. (3 Punkte) Entwirf einen Kellerautomaten für die Sprache $L = \{a^n b^{2n} \mid n \geq 1\}$.

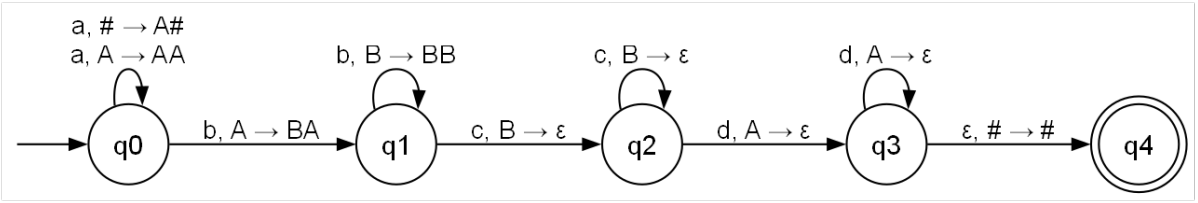
Lösung: Für jedes a werden zwei A abgelegt, jedes b nimmt ein A weg.



8. (5 Punkte) Gegeben ist die Sprache $L = \{a^n b^m c^m d^n \mid n, m \geq 1\}$.
   (a) Entwirf einen Kellerautomaten für L.
   (b) Gib eine kontextfreie Grammatik für L an.

Lösung:

(a) Für jedes a ein A ablegen, darüber für jedes b ein B. Jedes c nimmt ein B weg, danach jedes d ein A.

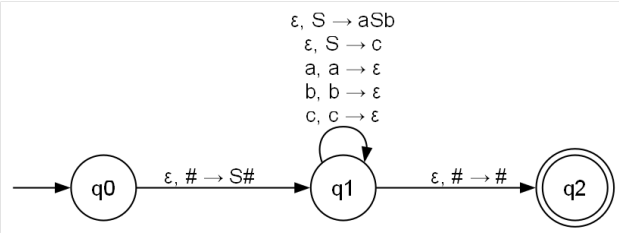


(b) Variablen $\{S, B\}$, Startsymbol S:
$S \to aSd \mid aBd$
$B \to bBc \mid bc$

9. (4 Punkte) Gegeben ist die Grammatik mit Startsymbol S und den Regeln $S \to aSb \mid c$.
   (a) Konstruiere wie im Notebook einen Kellerautomaten, der die Linksableitungen im Keller nachspielt.
   (b) Gib die Abarbeitung des Wortes acb an.

Lösung:

(a)



(b)

Zustand	Restwort	Keller
q_0	acb	`#`
q_1	acb	`S#`
q_1	acb	`aSb#`
q_1	cb	`Sb#`
q_1	cb	`cb#`
q_1	b	`b#`
q_1	ε	`#`
q_2	ε	`#`

10. (4 Punkte) (a) Wann heißt ein Kellerautomat deterministisch?

    (b) Für $\{w2w^R \mid w \in \{0,1\}^*\}$ gibt es einen deterministischen Kellerautomaten, für $\{ww^R \mid w \in \{0,1\}^*\}$ nicht. Erkläre den Unterschied anschaulich.

    (c) Bei endlichen Automaten können NEAs nicht mehr Sprachen erkennen als DEAs. Gilt das auch für Kellerautomaten?

Lösung:

(a) Wenn es in jeder Situation (Zustand, gelesenes Zeichen, oberstes Kellerzeichen) höchstens einen möglichen Übergang gibt. Gibt es für einen Zustand und ein oberstes Kellerzeichen einen ε-Übergang, darf es dafür keinen Übergang geben, der ein Zeichen liest.

(b) Der Automat legt die erste Hälfte auf den Keller und vergleicht danach die zweite Hälfte mit dem Keller. Bei $w2w^R$ zeigt die 2 an, wann er umschalten muss. Bei ww^R sieht er der Eingabe nicht an, wo die Mitte ist, er muss raten, und das kann nur ein nichtdeterministischer Automat.

(c) Nein. Nichtdeterministische Kellerautomaten erkennen genau die kontextfreien Sprachen, deterministische nur einen Teil davon, z.B. nicht $\{ww^R\}$.

11. (5 Punkte) (a) Erkläre anschaulich, warum kein Kellerautomat die Sprache $\{a^n b^n c^n \mid n \geq 1\}$ erkennt.

    (b) Gib kontextfreie Grammatiken für $L_1 = \{a^n b^n c^m \mid n, m \geq 0\}$ und $L_2 = \{a^m b^n c^n \mid n, m \geq 0\}$ an. Welche Sprache ist $L_1 \cap L_2$?

    (c) Was folgt aus b) für den Durchschnitt kontextfreier Sprachen?

Lösung:

(a) Der Automat kann die a im Keller zählen. Um zu prüfen, ob gleich viele b kommen, muss er sie beim Lesen der b wieder abbauen. Danach weiß er nicht mehr, wie viele c kommen müssen. Mit einem Keller kann man eine Anzahl nur mit einer anderen vergleichen.

(b) L_1: $S \to AC,\ A \to aAb \mid \varepsilon,\ C \to cC \mid \varepsilon$
L_2: $S \to AB,\ A \to aA \mid \varepsilon,\ B \to bBc \mid \varepsilon$
$L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$: gleich viele a wie b und gleich viele b wie c.

(c) Der Durchschnitt zweier kontextfreier Sprachen ist nicht immer kontextfrei. (Nach a) ist $\{a^n b^n c^n\}$ nicht kontextfrei, das leere Wort ändert daran nichts.)

12. (4 Punkte) Schreibe eine Funktion `klammern_ok(s)`, die mit einer Liste als Keller prüft, ob in einem String die Klammern `()`, `[]` und `{}` korrekt gesetzt sind. Andere Zeichen werden überlesen. Beispiele: `print([x*(y+1) for x in {1,2}])` ist korrekt, `f(a[1)]` und `(()` nicht.

Lösung:

```
def klammern_ok(s):
    passend = {')': '(', ']': '[', '}': '{'}
    keller = []
    for z in s:
        if z in '([{':
            keller.append(z)
        elif z in ')]}':
            if keller == [] or keller.pop() != passend[z]:
                return False
    return keller == []

print(klammern_ok('print([x*(y+1) for x in {1,2}])'))
print(klammern_ok('f(a[1)]'))
print(klammern_ok('(()'))
```

Ausgabe: `True`, `False`, `False`. Der Keller übernimmt die Rolle des Kellers im Kellerautomaten: Öffnende Klammern werden abgelegt, eine schließende muss zur obersten passen.