

Übungsaufgaben Informatik 4 Stünder KW14 Di, 31 März

Aufgabe 1: Ein PDA ist ein Septupel:

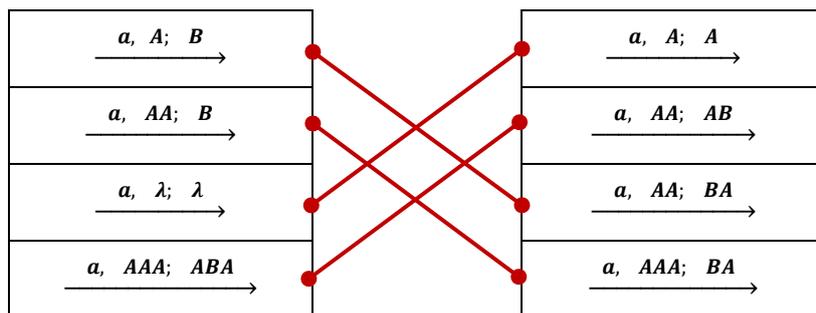
$$PDA = (Q, \Sigma, \Gamma, \delta, q_0, F, Z)$$

Wiederhole die Definition und erläutere die sieben Bestandteile der Definition

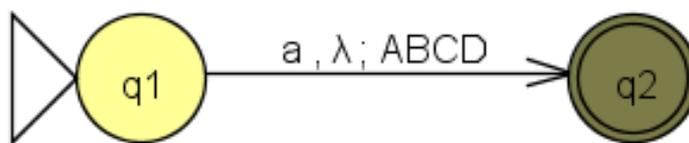
1. **Q:** Menge der Zustände: $q_0, q_1, q_2...$
2. **Σ :** Eingabealphabet
3. **Γ :** Kelleralphabet, zugehörig ist auch das Symbol λ
4. **δ :** Zustandsübergangsfunktion
5. **q_0 :** Startzustand
6. **F:** Menge der Endzustände
7. **Z:** Zeichen für leeren Keller

Aufgabe 2: Die möglichen Übergänge

a) Welche Übergänge auf der linken Seite entsprechen denen auf der rechten? Verbinde die sich entsprechenden Übergänge mit Pfeilen. In allen Fällen soll der aktuelle Stack so aussehen: *AAA*



Erläuterung:



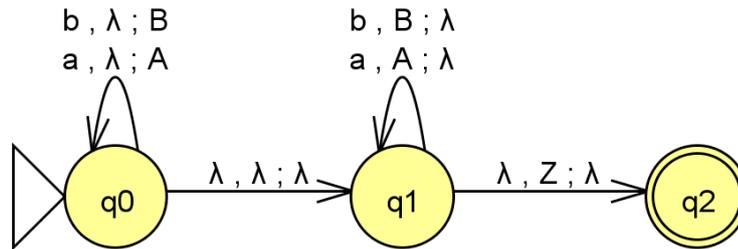
Wie man an dieser JFLAP-Simulation erkennt, wird beim Push auf den Stack zuerst das hinterste Symbol gelegt, hier also das „D“. Dann, „C“, danach „B“ und zuletzt „A“.

b) Um in JFLAP die Entsprechungen überprüfen zu können, müsste man einen Automaten bauen. Wie muss der Übergang von q_0 nach q_2 aussehen, damit der aktuelle Stack tatsächlich „AAA“ enthält?

λ, λ, AAA

Aufgabe 3: Sprachen mit dem PDA

a) Was macht dieser Automat? Hinweis: Versucht man ihn, in JFLAP zu simulieren, probiert JFLAP viele unterschiedliche Verzweigungen aus – von denen die meisten in einen roten FailedState führen. Warum ist das so (übrigens kann man den einzigen grünen State mit ‚Trace‘ sich genauer anschauen ...)



Zitat Florian:

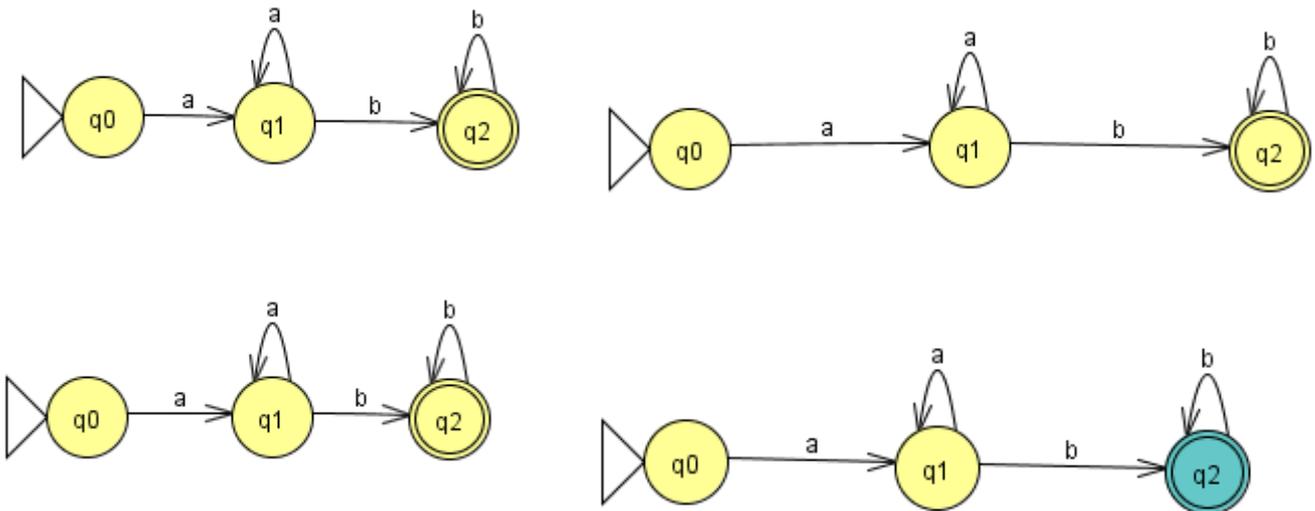
„Der Automat prüft Wörter aus den Buchstaben „a“ und „b“ darauf, ob sie ein Palindrom sind. Sehr viele Verzweigungen führen in einen FailedState, da der $\lambda, \lambda, \lambda$ - Übergang von q_0 nach q_1 zum falschen Zeitpunkt genutzt werden kann, und somit ein Fehler im Stack auftritt“

Zitat Sarah:

„Der Automat überprüft Palindrome mit gerader Anzahl von Buchstaben. Da der Automat ein NDA werden alle Möglichkeiten versucht, deshalb erschienen so viele Verzweigungen, welche in einen FailedState führen.“

b) Konstruiere einen Automaten für die Sprache $L_1 = \{a^n b^m \mid m, n > 0\}$
(braucht man einen PDA dafür? Begründe)

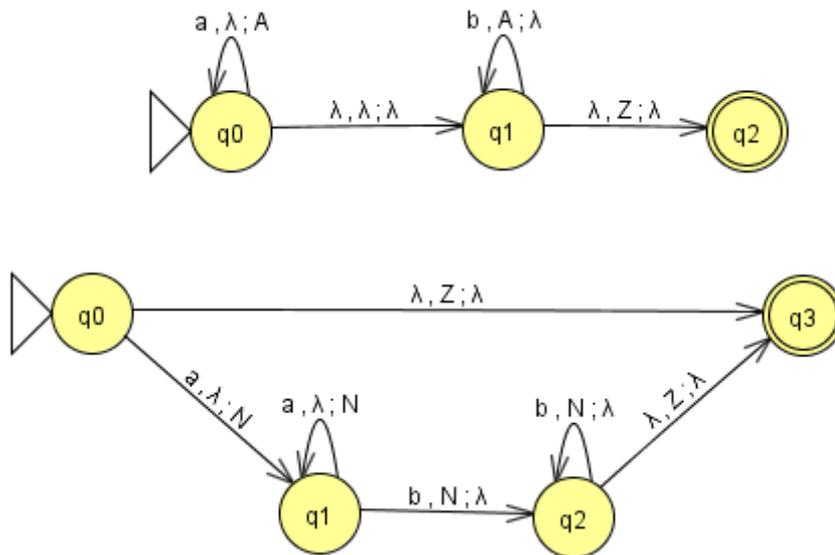
Hier die Beiträge von Michael, Sarah, Max Künzl und Florian. Erstaunlich unterschiedliche Lösungsansätze:



Es wird kein PDA benötigt, da die Anzahl der Buchstaben „a“ und „b“ voneinander unabhängig sind, und somit kein Stack zur Überprüfung notwendig ist.

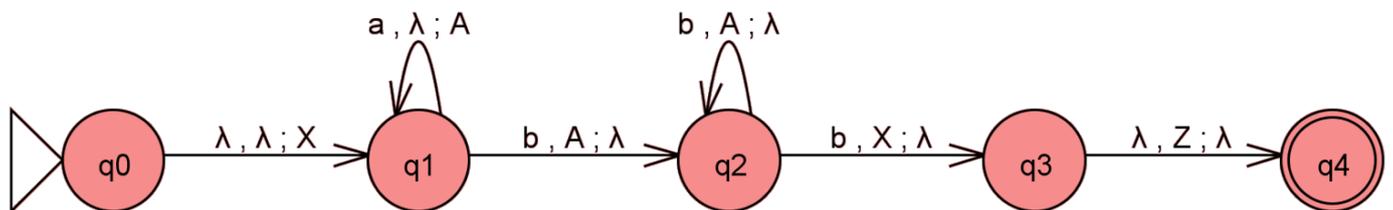
c) Konstruiere einen Automaten für die Sprache $L_2 = \{a^n b^n \mid n \geq 0\}$
 (Beachte die „größer gleich Null“-Bedingung)

Hier die Lösungen von Florian und Sarah:



d) Konstruiere einen Automaten für die Sprache $L_3 = \{a^n b^m \mid m > n \geq 0\}$
 (Beachte, dass diesmal „m“ größer als „n“ sein muss)

Step with Closure – Accept by Empty Stack



Der „Trick“ ist hier: der PDA akzeptiert keine Eingabe, bei der zuletzt noch das „X“ im Keller liegen bleibt, weshalb die Anzahl an „b“ immer größer sein muss als die Anzahl „a“.

zu guter letzt:

<https://www.cs.wcupa.edu/rkline/fcs/pdas.html>

<https://www.geeksforgeeks.org/npda-for-accepting-the-language-l-an-bn-n1/>

<https://www.cs.odu.edu/~zeil/cs390/latest/Public/pda-jflap/index.html>

