

Das Pumping Lemma

- Kann man einen DEA konstruieren, der die Sprache $L = \{a^k b^k \mid k \in \mathbb{N}_0\}$ erkennt?
 - Alle Wörter der Sprache bestehen aus einer bestimmten Anzahl an a's, gefolgt von der gleichen Anzahl an b's
- Nein, kann man nicht
 - Aber wie zeigen wir das?
- Es gilt bekanntlich: Sprache regulär \Leftrightarrow DEA existiert
 - Können wir zeigen, dass $L = \{a^k b^k \mid k \in \mathbb{N}_0\}$ nicht regulär ist?

Das Pumping Lemma

- Sei L eine beliebige reguläre Sprache
- Dann existiert eine natürliche Zahl $n \in \mathbb{N}$, sodass jedes Wort x der Sprache L , dessen Länge mindestens n beträgt ($|x| \geq n$) so in drei Teile u , v und w zerlegt werden kann (also $x = uvw$), dass gilt:
 - $|v| > 0$ (Das Teilwort v ist nicht leer)
 - $|uv| \leq n$ (Die Teilwörter u und v sind zusammen höchstens so lang wie n)
 - uv^*w gehört ebenfalls zu L (Das Teilwort v kann also beliebig oft wiederholt werden)

- Da L regulär ist, existiert ein DEA, der L akzeptiert
- Falls L nicht endlich ist, so **muss** der DEA mindestens eine Schleife enthalten, weil sonst mit endlich vielen Zuständen auch nur endlich viele Wörter akzeptiert werden könnten
 - Jeder Durchlauf dieser Schleife fügt ein weiteres Mal v hinzu

Was bringt und das jetzt?

- Bisher wissen wir:
 - Wenn L regulär, dann gilt das Pumping Lemma
- Umgekehrt heißt das:
 - Wenn das Pumping Lemma **nicht** gilt, dann ist L **nicht** regulär
- Vorsicht:
 - Aus “Pumping Lemma gilt” folgt **nicht** zwingend “Sprache regulär”!

Anwendung (1)

- Wir betrachten wieder die Sprache $L = \{a^k b^k \mid k \in \mathbb{N}_0\}$ und führen einen **Widerspruchsbeweis** durch
- Annahme: L ist regulär
- Dann existiert gemäß Pumping Lemma ein $n \in \mathbb{N}$ mit den oben genannten Eigenschaften
- Wir betrachten das Wort $a^n b^n \in L$
 - Gemäß Pumping Lemma existiert eine Zerlegung in drei Teile u, v, w mit $|uv| \leq n$ und $|v| > 0$
- Wegen $|uv| \leq n$ kann uv nur aus a's bestehen (denn das Wort beginnt ja mit n a's)
 - Insbesondere enthält auch v kein b
- Wegen $|v| > 0$ besteht daher auch v aus mindestens einem a

Anwendung (2)

- Durch “Aufpumpen” des Wortes $a^n b^n$ zu $uvvw$, $uvvvw$ usw. kommen also weitere a’s hinzu, jedoch keine b’s
 - Solche Wörter gehören **nicht** zu $L = \{a^k b^k \mid k \in \mathbb{N}_0\}$
- Das Pumping Lemma gilt also **nicht** \Rightarrow Die Sprache L ist nicht regulär
 - Was ist L dann?

Kontextfreie Grammatiken

- Eine Grammatik heißt **kontextfrei**, wenn alle Produktionsregeln den folgenden Anforderungen genügen:
 - auf der linken Seite steht nur genau ein Nichtterminalsymbol
 - auf der rechten Seite steht eine beliebige Folge aus Terminal- und Nichtterminalsymbolen
- Eine von einer kontextfreien Grammatik erzeugte Sprache heißt **kontextfreie Sprache**

- Regulär (und somit auch kontextfrei):

- $A \rightarrow aB|a; B \rightarrow bA|b$

- Kontextfrei, aber nicht regulär:

- $A \rightarrow aAb|B; B \rightarrow ab$

- Nicht kontextfrei:

- $aAb \rightarrow aab|aBb; B \rightarrow aAa|b$

Das Wortproblem bei kontextfreien Sprachen

- Für reguläre Sprachen kann ein DEA entscheiden, ob ein Wort zur Sprache gehört oder nicht
- Für kontextfreie Sprachen ist dies nicht mehr möglich
 - Beispiel $L = \{a^k b^k \mid k \in \mathbb{N}_0\}$: Der Automat bräuchte ein “Gedächtnis”, um die a’s und b’s zählen zu können
- Daher: Wir brauchen einen Automaten **mit** Speichermöglichkeit

- Ein Kellerautomat verfügt neben einer endlichen Zahl von Zuständen auch über einen (unendlichen) Speicher, der nach dem LIFO-Prinzip arbeitet (Stack/Keller)
- Ein Kellerautomat besteht aus:
 - endlicher Zustandsmenge
 - Eingabealphabet
 - Kelleralphabet
 - Startzustand
 - Menge von Endzuständen
 - Anfangssymbol auf dem Keller (bezeichnet mit Z oder $\#$)
 - Zustandsübergangsfunktion

Zustandsübergänge bei Kellerautomaten

- Welcher Zustandsübergang gewählt wird, hängt ab vom
 - aktuellen Zustand
 - Eingabezeichen
 - obersten Kellerzeichen
- $\delta(q_0, a, b) = (q_1, bb)$ bedeutet:
 - Falls der Automat im Zustand q_0 das Eingabezeichen a und das Kellerzeichen b liest, so entfernt er das b vom Keller, wechselt in den Zustand q_1 und schreibt bb auf den Keller
- Um Zeichen vom Keller zu löschen, kann das leere Wort ε darauf geschrieben werden

- Man kann Kellerautomaten auch ohne Endzustände definieren
 - Dann wird eine Eingabe akzeptiert, wenn nach Abarbeitung der Keller leer ist
- In ihrer ursprünglichen Definition sind Kellerautomaten nichtdeterministisch

- Starte das Program JFlap und wähle “Pushdown Automaton”
- Wähle “Multiple Character Input” ¹
- Zustände sind beschaffen wie bei DEAs
- Zustandsübergänge erwarten drei Werte:
 - Eingabezeichen
 - Kellerzeichen
 - auf den Keller zu legendes Zeichen
- Hinweis: JFlap verwendet standardmäßig λ statt ε für das leere Wort

¹Nur so ist es möglich, mehr als ein Zeichen auf den Stack zu legen und ihn dadurch wachsen zu lassen. Der Automat soll aber dennoch die Eingabe in Einzelzeichen verarbeiten.

Aufgabe 1

- Erstelle einen Kellerautomaten, der genau die Sprache $L = \{a^k b^k \mid k \in \mathbb{N} \setminus \{0\}\}$ akzeptiert

Aufgabe 2

- Erstelle einen Kellerautomaten, der die Sprache korrekter Klammerterme erkennt, also z. B. $(())$ oder $((()) ())$
 - Es muss so viele öffnende wie schließende Klammern geben
 - Die “Verschachtelung” der Klammern muss korrekt sein: $() ($ darf bspw. nicht akzeptiert werden

Aufgabe 3 - schwer!

- Erstelle einen Kellerautomaten, der prüft, ob die Eingabe ein Palindrom ist, das aus den Zeichen 0 und 1 besteht
- Hinweis und Hilfestellung:
 - Hier ist zwingend ein nichtdeterministischer Kellerautomat erforderlich
 - Zur Vereinfachung kann man zunächst einen deterministischen Kellerautomaten für Palindrome einer bestimmten Länge erstellen