Formale Grundlagen der Informatik 1
Kapitel 4
Über reguläre Sprachen hinaus
Kellerautomaten und Pumping Lemma

Frank Heitmann
heitmann@informatik.uni-hamburg.de

14. April 2014

Grenzen regulärer Sprachen?

Wir haben mittlerweile einiges kennengelernt, um reguläre Sprachen zu erzeugen:

- direkte Konstruktion von (verschiedenen) Automaten
- direkte Konstruktion von rationalen Ausdrücken
- Benutzung von Abschlusseigenschaften

Als Anwendungsfälle hatten wir

- Worterkennung in Texten
- speziell im Compilerbau

Grenzen regulärer Sprachen?

Und obwohl nun viele Sprachen regulär sind, gibt es (gerade auch in obigem Kontext) welche, bei denen es uns schwerfällt einen Automaten zu konstruieren.

Beispiel

Die Sprache aller korrekt geklammerten Ausdrücke ?

Z.B. wollen wir ()(()) akzeptieren, (() nicht.

Grenzen regulärer Sprachen?

Mit unseren bisherigen Techniken scheitern wir hier...
... geht das überhaupt?!

Eine einfachere Sprache, die aber einen wichtigen Aspekt erfasst:

$$\{a^n b^n \mid n \in \mathbb{N}\}$$

Ist diese regulär ?

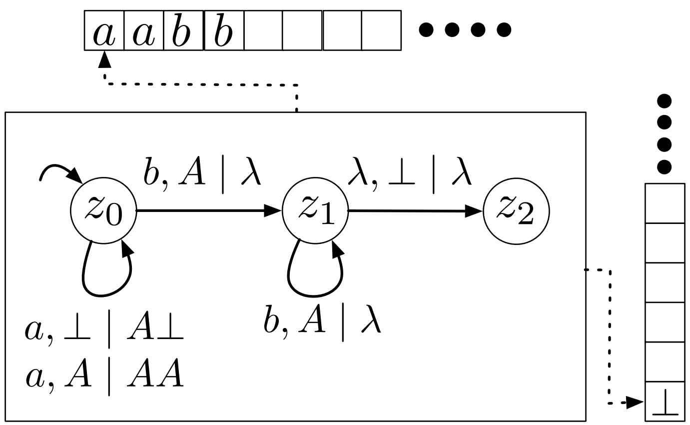
Wir scheitern zunächst auch hier ...

Kellerautomaten

Wir erweitern den DFA um einen Keller, um uns Dinge zu merken.

Den so entstehenden Automaten nennen wir **Kellerautomat**.

Der Kellerautomat



Der Kellerautomat - Formal

Definition (Kellerautomat (PDA = Push Down Automata))

Ein **nichtdeterministischer Kellerautomat** (kurz PDA) ist ein 7-Tupel $A = (Z, \Sigma, \Gamma, \delta, Z_{start}, Z_{end}, \perp)$ mit

- Der endlichen Menge von *Zuständen* Z.
- Dem endlichen Alphabet Σ von *Eingabesymbolen*.
- Dem endlichen Alphabet Γ von *Kellersymbolen*.
- Der *Überführungsfunktion* $\delta : Z \times (\Sigma \cup \{\lambda\}) \times \Gamma \to 2^{Z \times \Gamma^*}$.
- Der Menge von *Startzuständen* $Z_{start} \subseteq Z$.
- Der Menge der *Endzustände* $Z_{end} \subseteq Z$.
- Dem *Kellerbodensymbol* $\perp \in \Gamma$.

Anmerkung

Dieses Automatenmodell ist *nichtdeterministisch*

Der Kellerautomat - Formal

Wie beim NFA kann statt der Überführungsfunktion

$$\delta : Z \times (\Sigma \cup \{\lambda\}) \times \Gamma \to 2^{Z \times \Gamma^*}$$

auch mit einer Relation

$$K \subseteq Z \times (\Sigma \cup \{\lambda\}) \times \Gamma \times \Gamma^* \times Z$$

gearbeitet werden.

Konfiguration und Rechnung

Definition (Konfiguration und Rechnung)

Eine **Konfiguration** eines PDA A ist ein Element

$$(z, w, v) \in Z \times \Sigma^* \times \Gamma^*,$$

mit der Bedeutung,

- dass A im Zustand z ist,
- das w noch auf dem Eingabeband zu lesen ist und
- der aktuelle Kellerinhalt v ist.

Die **Überführungsrelation** $\vdash \subseteq (Z \times \Sigma^* \times \Gamma^*) \times (Z \times \Sigma^* \times \Gamma^*)$ ist definiert durch

$$(z, xu, yw) \vdash (z', u, y'w) \text{ gdw. } (z', y') \in \delta(z, x, y).$$

Eine **Rechnung** ist wieder eine Folge solcher Übergänge.

Akzeptierte Sprache

Definition (Akzeptierte Sprache)

Die von einem PDA *mit leerem Keller akzeptierte Sprache* ist die Menge

$$L_\lambda(A) := \{w \in \Sigma^* \mid (z_0, w, \perp) \vdash^* (z, \lambda, \lambda)\}$$

und die von einem PDA *mit Endzustand akzeptierte Sprache* ist die Menge

$$L_Z(A) := \{w \in \Sigma^* \mid (z_0, w, \perp) \vdash^* (z_e, \lambda, v), z_e \in Z_{end}, v \in \Gamma^*\}.$$

Anmerkung

Anmerkungen auf der nächsten Folie!

Akzeptierte Sprache

$$\begin{aligned} L_\lambda(A) &:= \{w \in \Sigma^* \mid (z_0, w, \perp) \vdash^* (z, \lambda, \lambda)\} \\ L_Z(A) &:= \{w \in \Sigma^* \mid (z_0, w, \perp) \vdash^* (z_e, \lambda, v), z_e \in Z_{end}, v \in \Gamma^*\} \end{aligned}$$

Anmerkung

- Bei $L_\lambda(A)$ werden die Endzustände nicht benutzt und
- bei $L_Z(A)$ ist der Kellerinhalt v am Ende egal.
- Statt $L_\lambda(A)$ schreiben wir auch (insb. im Skript) $N(A)$ und statt $L_Z(A)$ auch $L(A)$.
- Da das Modell nichtdeterministisch ist, genügt es wie beim NFA, wenn es bei einem Eingabewort **eine Rechnung gibt, die den Keller leert bzw. den Automaten in einen Endzustand bringt**.

Zur Akzeptanzbedingung

Bemerkung

Man kann zeigen, dass beide Akzeptanzbedingungen äquivalent sind, dass also

- Zu einem PDA A und einer Sprache M mit $M = L_Z(A)(= L(A))$ auch ein PDA A' konstruiert werden kann mit $M = L_\lambda(A')(= N(A'))$.
- Zu einem PDA A' und einer Sprache M mit $M = L_\lambda(A')(= N(A'))$ auch ein PDA A konstruiert werden kann mit $M = L_Z(A)(= L(A))$.

Wir wollen dies nicht im Detail beweisen und konzentrieren und meist auf die Akzeptanz mit leerem Keller.

Literaturhinweis

Interessierte finden die Konstruktion(en) im Skript und in [HMU]. Die Kernidee ist bei beiden Richtungen mit einem "zweiten" Kellerbodenzeichen zu arbeiten und dies unter das erste zu schieben. Man kann dann genau feststellen, wann der Keller in einem Automaten leer ist, kann aber trotzdem noch weiter arbeiten.

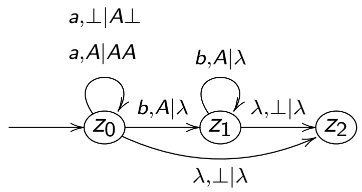
Kellerautomat - Funktionsweise

Die (informale) Funktionsweise eines PDA:

- Ein PDA beginnt im Startzustand z_0 und mit $\perp$ im Keller.
- Ist der Automat
  - im Zustand z,
  - liest vom Eingabeband das a
  - und vom Keller das X (dies ist ganz oben auf dem Keller),
- so kann ein Element $(z', w) \in \delta(z, a, X)$ gewählt werden.
- Es wird dann
  - in den Zustand z' gewechselt,
  - der Lesekopf auf dem Eingabeband ein Feld nach rechts bewegt,
  - das X vom Keller gelöscht
  - und dieses X durch w ersetzt (wobei das erste Symbol von w nun ganz oben auf dem Keller ist).

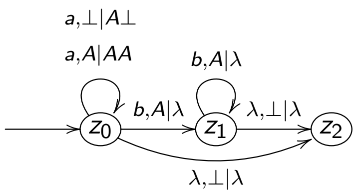
Beispiel $a^n b^n$

$$L = \{a^n b^n \mid n \in \mathbb{N}\}$$



Beispiel $a^n b^n$

$$L = \{a^n b^n \mid n \in \mathbb{N}\}$$

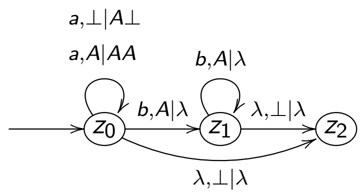


$$\begin{aligned}(z_0, aabb, \perp) \;\vdash\; & (z_0, abb, A\perp) \vdash (z_0, bb, AA\perp) \vdash \\ & (z_1, b, A\perp) \vdash (z_1, \lambda, \perp) \vdash (z_2, \lambda, \lambda)\end{aligned}$$

Akzeptieren! Wort zu Ende gelesen und Keller leer!

Beispiel $a^n b^n$

$$L = \{a^n b^n \mid n \in \mathbb{N}\}$$

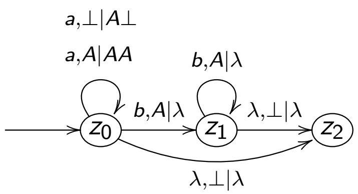


$$(z_0, abb, \perp) \vdash (z_0, bb, A\perp) \vdash (z_1, b, \perp) \vdash (z_1, b, \lambda)$$

Nicht akzeptieren, da Wort nicht zu Ende gelesen!

Beispiel $a^n b^n$

$$L = \{a^n b^n \mid n \in \mathbb{N}\}$$



$$(z_0, aa, \perp) \vdash (z_0, a, A\perp) \vdash (z_0, \lambda, AA\perp)$$

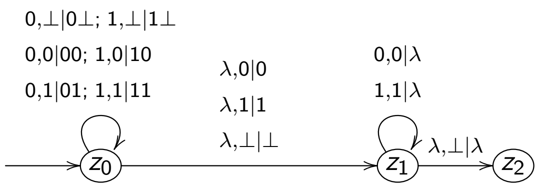
Nicht akzeptieren, da Keller nicht leer!

DFA vs. PDA

Wichtige Anmerkung

Alles was ein DFA kann, kann auch ein PDA! Jede Kante (z, a, z') im DFA wird im PDA zur Kante $(z, a, \perp, \perp, z')$. Will man mit leerem Keller akzeptieren, fügt man an jeden Endzustand (des DFAs) z_e noch eine Kante $(z_e, \lambda, \perp, \lambda, z_e)$ hinzu, mit der das $\perp$ gelöscht wird. Der PDA ist also *mindestens so mächtig* wie der DFA. Die Frage ist später noch, ob er tatsächlich mehr kann...

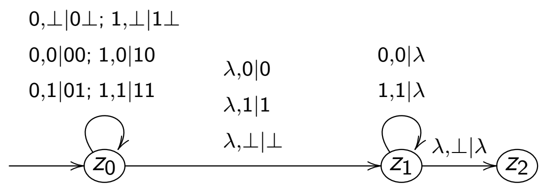
Fragen ...



In welchen Zuständen kann man nach Lesen von 0011 sein?

1. Nur z_0
2. z_0 und z_1
3. z_0, z_1 und z_2
4. Nur z_2

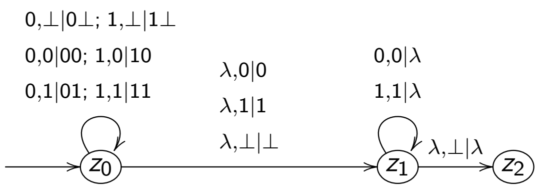
Fragen ...



Unabhängig vom Eingabewort: In welchen Zuständen kann man sein, so dass der Keller leer ist?

1. Nur z_0
2. Nur z_1
3. Nur z_2
4. In z_1 und z_2 kann der Keller leer sein

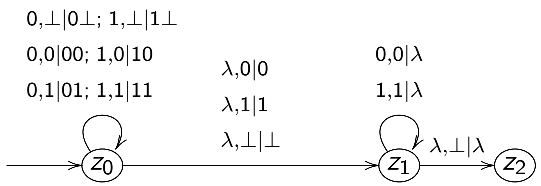
Fragen ...



In welchen Zuständen kann man nach Lesen von 0110 sein?

1. Nur z_0
2. z_0 und z_1
3. z_0, z_1 und z_2
4. Nur z_2

Fragen ...



Bei diesem Automaten kann man mit leeren Keller nicht mehr lesen, weil es keine Kante aus z_2 heraus gibt. Ist es im Allgemeinen möglich bei leerem Keller weitere Aktionen auszuführen? Also z.B. hier eine Kante $(z_2, \lambda, \lambda, \lambda, z_0)$ einzufügen?

1. Ja!
2. Nein!

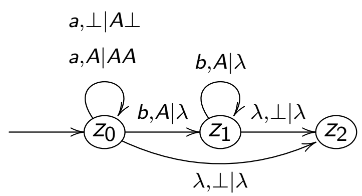
Zur Nachbereitung

Zur Nachbereitung

Richtige Antworten sind:

1. 2
2. 3
3. 3
4. 2

Beispiel $a^n b^n$

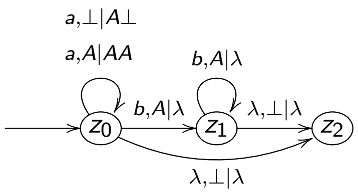


Behauptung

$$N(A) = \{a^n b^n \mid n \in \mathbb{N}\} =: M$$

Vorüberlegung: Es gilt $\lambda \in M$ (mit $n = 0$) und mit $(z_0, \lambda, \bot) \vdash (z_2, \lambda, \lambda)$ auch $\lambda \in N(A)$. Wir können daher nachfolgend davon ausgehen, dass $|w| > 0$ gilt.

Korrektheitsbeweis

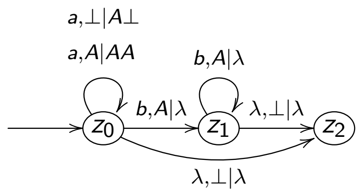


$M := \{a^n b^n \mid n \in \mathbb{N}\} \subseteq N(A)$. Sei $w = a^n b^n \in M$ für ein $n \geq 1$. In A gilt nun

$$\begin{aligned} (z_0, a^n b^n, \perp) &\vdash (z_0, a^{n-1} b^n, A\perp) \vdash \ldots \vdash (z_0, b^n, A^n \perp) \\ &\vdash (z_1, b^{n-1}, A^{n-1}\perp) \vdash \ldots \vdash (z_1, \lambda, \perp) \\ &\vdash (z_2, \lambda, \lambda) \end{aligned}$$

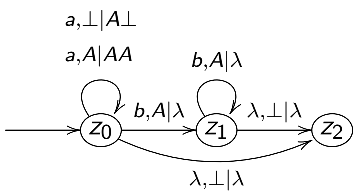
und folglich auch $w \in N(A)$.

Korrektheitsbeweis



$N(A) \subseteq \{a^n b^n \mid n \in \mathbb{N}\} = M$. Sei $w \in N(A)$ mit $|w| \geq 1$. Der Kantenübergang von z_0 nach z_2 ist nur ganz zu Anfang möglich, da sonst in z_0 nie vom Keller gelöscht wird und das Kellerbodensymbol dann nicht mehr erreichbar ist. D.h. diese Kante kann nur genutzt werden, um das leere Wort zu akzeptieren und ist hier nicht von Bedeutung.

Korrektheitsbeweis

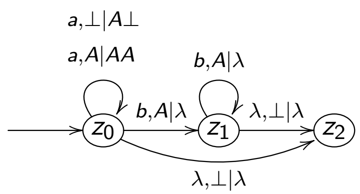


An den Kantenübergängen erkennt man nun, dass

- ausschließlich in z_0 *a*s gelesen werden können.
- Für jedes gelesene a wird zudem ein A auf den Keller geschrieben.
- Zudem muss w mit mindestens einem a beginnen, da sonst kein Übergang möglich ist.

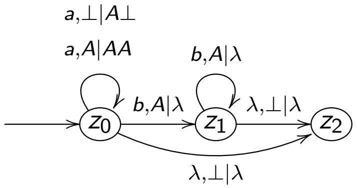
Insgesamt heißt dies, dass w mit a^i für ein $i \geq 1$ beginnen muss.

Korrektheitsbeweis



Sei also $w = a^i v$, wobei v das Restwort ist und mit b beginnen muss (da sonst z_0 nicht verlassen werden kann). Nach Lesen von a^i ist A dann in der Konfiguration $(z_0, v, A^i\perp)$. An den Kantenübergängen erkennt man nun, dass genau i mal das b gelesen werden muss, um an das $\perp$ heranzukommen. Dies geschieht beim Übergang nach z_1 und dann in z_1. Bei jedem Lesen eines b wird ein A gelöscht. Da wir an $\perp$ herankommen müssen, ist es also weder möglich weniger noch mehr als i *b*s zu lesen.

Korrektheitsbeweis



Es muss also gerade $v = b^i$ sein. Erst nun ist das $\perp$ im Keller erreichbar und wir können es beim Übergang nach z_2 löschen. Damit hat w die Form $a^i b^i$ für ein $i \geq 1$ und damit gilt auch $w \in M$.

Kellerautomaten Motivation
Pumping Lemma Formales

Die Technik

Zur Zähltechnik

Diese Technik des Zählens kann häufig benutzt werden. Varianten:

- Um z.B. $a^n b^{2n}$ zu akzeptieren, achtet man beim Löschen der A im Keller darauf, dass hierfür immer zwei b gelesen werden müssen (oder alternativ: schreibt stets zwei A beim Lesen eines a).
- Für $a^n b^m$ mit $n > m$ muss der Keller beim letzten zu lesenden Eingabesymbol nicht leer werden. Er muss dann aber noch in einem Zustand (in dem man davon ausgeht, dass das Wort zu Ende gelesen wurde) noch geleert werden.
- Für $a^n b^m$ mit $n < m$ liest man, wenn man $\perp$ erreicht hat, noch beliebig viele weitere b.

Kellerautomaten Motivation
Pumping Lemma Formales

Die Technik

Zur Zähltechnik

Weitere Varianten:

- Will man Worte akzeptieren, bei denen jedes Präfix mehr 0en als 1en hat, kann man auch mit einem Zähler arbeiten, muss diesen aber dynamisch erhöhen und verringern und ggf. abbrechen, wenn man versucht ihn bei $\perp$ noch weiter zu verringern. ($\Rightarrow$ Präsenzaufgabe)
- Will man Worte akzeptieren, die gleich viele 0en wie 1en haben, kann man ähnlich verfahren, aber der Zähler "flippt" sozusagen von einem 0-Zähler zu einem 1-Zähler (oder andersherum) und am Ende des Wortes muss der Zähler leer sein. ($\Rightarrow$ Übungsaufgabe)

Anmerkung

Definition des Präfix auf der nächsten Folie.

Kellerautomaten Motivation
Pumping Lemma Formales

Zwei Definitionen

Definition (Präfix)

Unter einem **Präfix** eines Wortes $w \in \Sigma^*$ versteht man ein Wort $v \in \Sigma^*$ so dass ein $u \in \Sigma^*$ existiert mit $w = vu$. Informal: ein beliebiges Anfangsstück von w ist ein Präfix. Ist bspw. $w = 1001$, so ist 1 ein Präfix, ebenso wie 10 und 100. Außerdem sind auch λ und w selbst Präfixe von w.

Definition (w^{rev})

Das **Spiegelwort** w^{rev} zu einem Wort $w \in \Sigma^*$ ist definiert durch

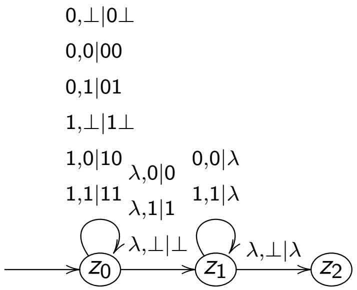
1. $\lambda^{rev} = \lambda$ und
2. $(ux)^{rev} = x \cdot u^{rev}$ mit $x \in \Sigma$ und $u \in \Sigma^*$

Bspw. ist für $w = 1011$ dann $w^{rev} = 1101$.
Für Mengen notieren wir $L^{rev} = \{w^{rev} \mid w \in L\}$.

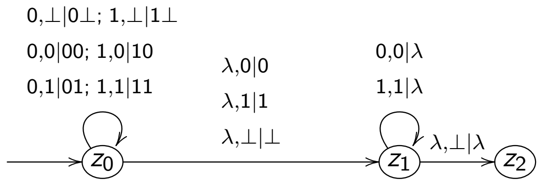
Ein zweites Beispiel: ww^{rev}

Wir wollen nun einen PDA konstruieren für

$$L := \{ww^{rev} \mid w \in \{0,1\}^*\}$$



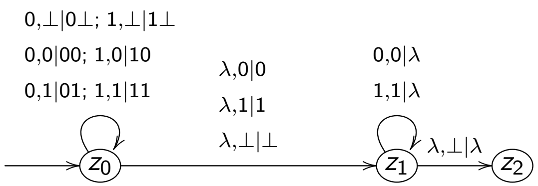
Korrektheitsbeweis



$L := \{ww^{rev} \mid w \in \{0,1\}^*\} \subseteq N(A)$. Sei $v = ww^{rev} \in L$. Nach Konstruktion

- wird in z_0 für jede vom Eingabeband gelesene 0, eine 0 auf den Keller geschrieben; ebenso für die 1;
- in z_1 kann eine 0 nur vom Eingabeband gelesen werden, wenn sie auch auf dem Keller steht (und dann gelöscht wird); ebenso für die 1;

Korrektheitsbeweis



Damit folgt also nach Konstruktion:

$$\begin{aligned}(z_0, ww^{rev}, \perp) &\vdash \ldots \vdash (z_0, w^{rev}, w^{rev}\perp)\\ &\vdash (z_1, w^{rev}, w^{rev}, \perp) \vdash \ldots \vdash (z_1, \lambda, \perp)\\ &\vdash (z_2, \lambda, \lambda)\end{aligned}$$

Also $v = ww^{rev} \in N(A)$.

Korrektheitsbeweis

Die Rückrichtung ist schwieriger. Wir beobachten zunächst:

- z_2 kann nur von z_1 aus erreicht werden und nur, wenn auf dem Keller nur noch $\perp$ ist. Dies ist die einzige Möglichkeit den Keller zu leeren.
- In z_2 kann nichts mehr gelesen werden. Das Eingabewort muss also schon in z_1 zu Ende gelesen worden sein.
- Jede akzeptierende Rechnung startet in z_0, gelangt von dort nach z_1 und kehrt von dort nie nach z_0 zurück.

Es genügt also Bedingungen für

$$(z_0, w, \perp) \vdash^* (z_1, \lambda, \perp)$$

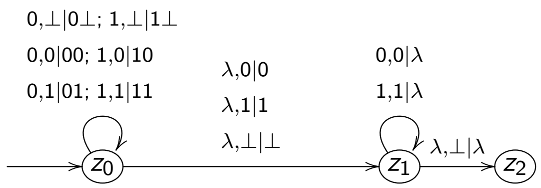
zu finden. Dies sind dann genau die Wörter, die akzeptiert werden!

Die Behauptung

Wir zeigen eine etwas allgemeinere Behauptung **mittels Induktion über die Wortlänge**:

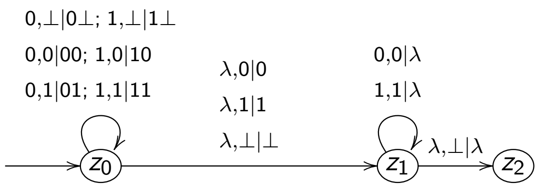
Wenn $(z_0, x, V) \vdash^* (z_1, \lambda, V)$ (für beliebiges V), dann ist $x = ww^{rev}$ für ein $w \in \{0,1\}^*$

Korrektheitsbeweis



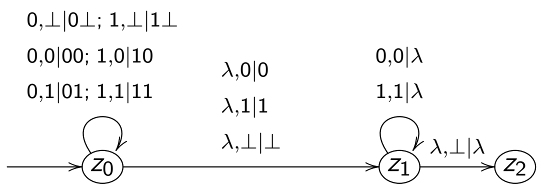
(Z.Z. Wenn $(z_0, x, V) \vdash^* (z_1, \lambda, V)$, dann $x = ww^{rev}$.)
Induktionsanfang: Sei $x = \lambda$. Dann ist x von der gewünschten Form und wir sind fertig (unabhängig davon, ob der "wenn"-Teil gilt oder nicht).

Korrektheitsbeweis



Induktionsannahme: Gelte die Behauptung für Worte der Länge bis zu $n-1$, d.h. für jedes Wort x der Länge bis zu $n-1$ gilt, wenn $(z_0, x, V) \vdash^* (z_1, \lambda, V)$ für ein V, dann ist $x = ww^{rev}$ für ein $w \in \{0,1\}^*$

Korrektheitsbeweis



Induktionsschritt: Sei $x = a_1 \dots a_n$ ein Wort der Länge n mit $(z_0, x, V) \vdash^* (z_1, \lambda, V)$ für ein V. Wir wollen $x = ww^{rev}$ zeigen. Zunächst kann der erste Schritt nicht $(z_0, x, V) \vdash (z_1, x, V)$ sein, da in z_1 nur Buchstaben vom Keller gelöscht werden können und wegen $|x| \geq 1$ auf jeden Fall mindestens ein Buchstabe gelöscht werden würde. Dann wäre aber nicht mehr V auf dem Keller!

Korrektheitsbeweis

Der erste Schritt ist also $(z_0, a_1 a_2 \dots a_n, V) \vdash (z_0, a_2 \dots a_n, a_1 V)$. Da wir ferner das a_1 wieder vom Keller löschen müssen und dies nur in z_1 geschehen kann, muss der letzte Schritt $(z_1, a_n, a_1 V) \vdash (z_1, \lambda, V)$ sein. Hieraus folgt zunächst $a_n = a_1$ (wegen der Übergänge an z_1). Außerdem wissen wir, da ja nach Voraussetzung $(z_0, x, V) \vdash^* (z_1, \lambda, V)$ gilt, dass nun insgesamt

$$(z_0, a_1 a_2 \dots a_n, V) \vdash (z_0, a_2 \dots a_n, a_1 V) \vdash^* (z_1, a_n, a_1 V) \vdash (z_1, \lambda, V)$$

also insb.

$$(z_0, a_2 \dots a_n, a_1 V) \vdash^* (z_1, a_n, a_1 V)$$

Da das a_n von der Eingabe nicht benutzt wird ist dies gleichbedeutend mit

$$(z_0, a_2 \dots a_{n-1}, a_1 V) \vdash^* (z_1, \lambda, a_1 V)$$

Korrektheitsbeweis

$$(z_0, a_2 \dots a_{n-1}, a_1 V) \vdash^* (z_1, \lambda, a_1 V)$$

Nun ist aber $a_2 \dots a_{n-1}$ ein Wort der Länge $n - 2 < n$, so dass die Induktionsannahme anwendbar ist! Es gibt also ein v derart, dass $a_2 \dots a_{n-1} = vv^{rev}$. Da ferner $a_1 = a_n$ gilt, haben wir mit $w = a_1 v$ $a_1 \dots a_n = a_1 vv^{rev} a_n = a_1 vv^{rev} a_1 = ww^{rev}$ und sind fertig.

Die Technik

Zur rev-Technik

Es gilt ganz allgemein, dass wenn man ein Wort w von der Eingabe liest und dieses Buchstabe für Buchstabe auf den Keller schreibt. Auf dem Keller genau w^{rev} liegt. Dies kann man ggf. auch in anderen Kontexten benutzen...

Zusammenfassung

Wir haben

- PDAs / Kellerautomaten kennengelernt
- Wieder die Begriffe wie Konfiguration, Rechnung, akzeptierte Sprache usw. eingeführt

Dabei

- können PDAs mit leerem Keller oder mit Endzustand akzeptieren.
- Beide Akzeptanzbedingungen sind äquivalent (ohne Beweis).

Wir haben

- einen PDA für $L = \{a^n b^n \mid n \in \mathbb{N}\}$ gebaut
  - und dabei den Keller als Zähler benutzt
- einen PDA für $L = \{ww^{rev} \mid w \in \{0, 1\}^*\}$ gebaut
  - und dabei den Keller als Speicher benutzt

Zum Pumping Lemma hin...

Wir haben jetzt einen PDA für $L := \{a^n b^n \mid n \in \mathbb{N}\}$, *aber* bisher wissen wir nicht mit Sicherheit, dass es hierfür keinen DFA gibt! Können wir vielleicht eine Eigenschaft X finden, die alle regulären Sprachen haben müssen? Dann könnten wir vielleicht nachweisen, dass L oben die Eigenschaft X nicht hat. Dann kann L nicht regulär sein!

Zum Pumping Lemma hin...

Wir wissen von regulären Sprachen, dass

- sie von endlichen Automaten akzeptiert werden.

Von diesen wissen wir, dass

- sie endlich viele Zustände haben.

Zum Pumping Lemma hin...

- Wörter, die akzeptiert werden und
- die mehr Buchstaben haben als der Automat Zustände,
- müssen dann Schleifen im Automaten durchlaufen!
- Diese Schleifen könnte man öfter entlang gehen!
- Und damit neue Wörter erzeugen, die auch in der Sprache sein müssen!

Bei $a^n b^n$ wird das aber nicht gehen! Man überlege sich:

- Wenn es einen Automat gibt mit k Zuständen für L
- betrachtet man $a^k b^k$
- Es müsste eine Schleife geben
- Aber welche sollte das sein, so dass man sie zweimal entlang gehen kann und trotzdem ein Wort der Form $a^j b^j$ herauskommt?

Das Pumping Lemma

Wir werden jetzt die oben skizzierte Eigenschaft der regulären Sprachen genauer fassen und so das **Pumping Lemma** formulieren.

Dies werden wir dann nutzen, um tatsächlich zu zeigen, dass $L = \{a^n b^n\}$ nicht regulär ist!

Das Pumping Lemma - Das Lemma

Lemma

Sei $L \in REG$ eine reguläre Sprache. Dann gibt es ein $n \in \mathbb{N}$, so dass jedes Wort $z \in L$ mit $|z| \geq n$ in die Form $z = uvw$ zerlegt werden kann, wobei

1. $|uv| \leq n$
2. $|v| \geq 1$
3. $uv^iw \in L$ *für jedes* $i \in \mathbb{N}$ *(inkl. der 0)*

(Äquivalent zur letzten Aussage ist die Aussage $\{u\}\{v\}^*\{w\} \subseteq L$.)

Wichtige Anmerkung

Das v kann (mit $i = 0$) auch ganz weggelassen werden, d.h. die dritte Bedingung fordert, dass nicht nur $uvw, uvvw, uvvvw$ usw. in L sein müssen, sondern auch $uv^0w = u\lambda w = uw$. Dies ist manchmal die einzige Möglichkeit einen Widerspruch herzustellen!

Das Pumping Lemma - Der Beweis

Beweis.

Sei L regulär. Dann gibt es einen vollständigen DFA A mit $L(A) = L$. Sei n die Anzahl der Zustände von A. Wir betrachten nun ein Wort $z \in L$ mit $|z| \geq n$ (gibt es kein solches z ist nichts zu zeigen).

- Wir legen A das Wort z vor.
- A beginnt im Startzustand z_0
- Da A vollständig ist und z akzeptiert wird, kann z gelesen werden.
- Da z aus mindestens n Buchstaben besteht, finden mindestens n Zustandsübergänge statt.
- D.h. es werden mindestens $n + 1$ Zustände besucht.
- Da es nur n Zustände gibt, muss mindestens ein Zustand doppelt auftreten!

Das Pumping Lemma - Der Beweis

Beweis.

Sei die Zustandsfolge vom Startzustand z_0 zu einem Endzustand z_e, die beim Lesen von z besucht wird durch $z_0, z_1, z_2, \dots, z_n, \dots, z_e$ gegeben.

- Seien z_i und z_j ($i \neq j$ und $i < j$) in dieser Folge gleich und außerdem z_j der erste Zustand, der ein zweites Mal auftritt!
- Nun ist $j \leq n$ (denn von z_0 bis z_n sind es bereits $n + 1$ Zusände; einer davon muss doppelt auftreten!)
- und $j - i \geq 1$
- Sei u das Wort das von z_0 bis z_i gelesen wird
- v das Wort das von z_i bis z_j gelesen wird
- und w das Wort von z_j bis z_e

Das Pumping Lemma - Der Beweis

Beweis.

- Sei u das Wort das von z_0 bis z_i gelesen wird
- v das Wort das von z_i bis z_j gelesen wird
- und w das Wort von z_j bis z_e

Es ist nun $|uv| \leq n$ (wegen $j \leq n$ oben) und $|v| \geq 1$ (wegen $j - i \geq 1$ oben). Ferner kann die Schleife von z_i nach z_j (man beachte: $i \neq j$, aber $z_i = z_j$!) beliebig oft durchlaufen werden und trotzdem kann dann von z_j aus mit w der Endzustand z_n erreicht werden. Daraus folgt, dass auch uv^iw für alle $i \geq 0$ akzeptiert werden. Und wir sind fertig! $\square$

Das Pumping Lemma - Das Lemma

Lemma

Sei $L \in REG$ eine reguläre Sprache. Dann gibt es ein $n \in \mathbb{N}$, so dass jedes Wort $z \in L$ mit $|z| \geq n$ in die Form $z = uvw$ zerlegt werden kann, wobei

1. $|uv| \leq n$
2. $|v| \geq 1$
3. $uv^iw \in L$ *für jedes* $i \in \mathbb{N}$ *(inkl. der 0)*

(Äquivalent zur letzten Aussage ist die Aussage $\{u\}\{v\}^*\{w\} \subseteq L$.)

Wichtige Anmerkung

Das v kann (mit $i = 0$) auch ganz weggelassen werden, d.h. die dritte Bedingung fordert, dass nicht nur $uvw, uvvw, uvvvw$ usw. in L sein müssen, sondern auch $uv^0w = u\lambda w = uw$. Dies ist manchmal die einzige Möglichkeit einen Widerspruch herzustellen!

Das Pumping Lemma - Der Ablauf

Der Ablauf beim Pumping Lemma:

1. Annehmen L wäre regulär
2. Die Zahl aus dem Pumping Lemma benennen (z.B. k).
3. **Ein Wort** z finden mit $|z| \geq k$ und $z \in L$.
   - Dieses Wort muss gut gewählt sein, damit der nachfolgende Widerspruch gelingt! Hier muss man also u.U. experimentieren!
4. **Für alle Zerlegungen** von z in uvw zeigen, dass sie im Widerspruch zur ersten, zweiten oder dritten Bedingung sind.
   - Üblicherweise betrachtet man alle Zerlegungen, die die erste und zweite Bedingung erfüllen und zeigt, dass man dann einen Widerspruch zur dritten Bedingung erhält.
5. Also: Alle Zerlegungen $z = uvw$ mit $|uv| \leq k$ und $|v| \geq 1$ betrachten.
6. Zeigen, dass man **ein** i findet mit $uv^iw \notin L$. (Dies kann auch $i = 0$ sein!)

Das Pumping Lemma - Beispiel 1

Behauptung

$L := \{a^nb^n \mid n \in \mathbb{N}\}$ ist nicht regulär.

Beweis.

Angenommen L wäre regulär. Dann gilt das Pumping Lemma. Sei k die Zahl aus dem Pumping Lemma. Wir betrachten das Wort $z = a^kb^k$. Es gilt $z \in L$ und $|z| \geq k$. Damit muss es eine Zerlegung $z = uvw$ geben, die die drei Eigenschaften erfüllt. Wir betrachten nun nur jene Zerlegungen, die die erste und zweite Bedingung erfüllen und zeigen, dass diese im Widerspruch zur dritten stehen. Wir haben dann gezeigt, dass jede Zerlegung von z in uvw im Widerspruch zu einer der drei Bedingungen steht und sind dann fertig. ...

Das Pumping Lemma - Beispiel 1

Behauptung

$L := \{a^nb^n \mid n \in \mathbb{N}\}$ ist nicht regulär.

Beweis.

Sei also $z = a^kb^k = uvw$ eine Zerlegung mit $i)|uv| \leq k$ und $ii)|v| \geq 1$. Es muss dann $v \in \{a\}^+$ gelten, da uv ja die ersten Buchstaben von z ausmacht, aber davon nur maximal k einnehmen kann und da ferner v mindestens die Länge 1 hat. Es gibt sogar ein j mit $1 \leq j \leq k$ derart, dass $v = a^j$ gilt. Wir betrachten nun das Wort uv^2w, dass nach der dritten Bedingung in L sein müsste. Es ist aber $uv^2w = a^{k+j}b^k$ und somit $uv^2w \notin L$. Die ursprüngliche Annahme muss also falsch sein und daher ist L nicht regulär. □

Wichtige Folgerung

Wichtige Anmerkung

Damit ist bewiesen,
dass der PDA tatsächlich mehr kann als der DFA!
Die Sprachfamilie der von einem PDA akzeptierten Sprachen ist also echt größer als die Sprachfamilie REG (der von DFAs akzeptierten Sprachen).

Fragen...

Wovon muss das gewählte Wort z abhängen?

1. Von der Anzahl der Zustände $|Z|$ des DFA.
2. Von der Zahl aus dem Pumping Lemma k, die noch genauer spezifiziert werden muss.
3. Von der Zahl aus dem Pumping Lemma k, die nicht genauer spezifiziert werden muss.
4. Das Wort muss leidiglich in der Sprache sein.

Fragen...

Wieviele Zerlegungen von z müssen betrachtet werden?

1. Eine muss zum Widerspruch geführt werden!
2. Alle Zerlegungen von z müssen zum Widerspruch gebracht werden!
3. Zerlegungen mit bestimmten Eigenschaften müssen zum Widerspruch gebracht werden!
4. Man muss eine Zerlegung mit bestimmten Eigenschaften finden und diese zum Widerspruch führen!

Fragen...

Kann mit dem Pumping Lemma gezeigt werden, dass eine Sprache regulär ist?

1. Ja!
2. Nein!

Zur Nachbereitung

Zur Nachbereitung

Richtige Antworten sind:

1. 3
2. 2 und 3 ist beides ok
3. 2

Das Pumping Lemma - Beispiel 2

Behauptung

$L := \{ww^{rev} \mid w \in \{0,1\}^*\}$ ist nicht regulär.

Beweis.

Angenommen L wäre regular. Dann gilt das Pumping Lemma. Sei k die Zahl aus dem Pumping Lemma. Wir betrachten $z = 0^k1^k1^k0^k$. Es ist $z \in L$ und $|z| \geq k$. Sei $z = uvw$ eine Zerlegung von z mit $|uv| \leq k$ und $|v| \geq 1$. Hieraus folgt $v = 0^j$ für ein j mit $1 \leq j \leq k$. Nun ist aber $uv^0w = 0^{k-j}1^k1^k0^k$ nicht mehr in L im Widerspruch zum Pumping Lemma. Die Annahme ist daher falsch und L somit nicht regulär. □

Das Pumping Lemma - Nochmal der Ablauf

Der Ablauf beim Pumping Lemma:

1. Annehmen L wäre regulär
2. Die Zahl aus dem Pumping Lemma benennen (z.B. k).
3. **Ein Wort** z finden mit $|z| \geq k$ und $z \in L$.
   - Dieses Wort muss gut gewählt sein, damit der nachfolgende Widerspruch gelingt! Hier muss man also u.U. experimentieren!
4. **Für alle Zerlegungen** von z in uvw zeigen, dass sie im Widerspruch zur ersten, zweiten oder dritten Bedingung sind.
   - Üblicherweise betrachtet man alle Zerlegungen, die die erste und zweite Bedingung erfüllen und zeigt, dass man dann einen Widerspruch zur dritten Bedingung erhält.
5. Also: Alle Zerlegungen $z = uvw$ mit $|uv| \leq k$ und $|v| \geq 1$ betrachten.
6. Zeigen, dass man **ein** i findet mit $uv^iw \notin L$. (Dies kann auch $i = 0$ sein!)

Das Pumping Lemma - Anmerkung zum Ablauf

Anmerkung

Bei der Zerlegung (vorletzter Schritt eben) gelingt oft eine Parametrisierung. Bspw. konnte in den Beispielen oben $v = 0^j$ bzw. $v = a^j$ mit einem j mit $1 \leq j \leq k$ gesetzt werden. Das j wandert quasi den Bereich von 1 bis k ab. Für jedes j ist dies eine mögliche Zerlegung, die die ersten beiden Bedingungen erfüllt – und man führt im Grunde genommen *alle* diese Zerlegungen zum Widerspruch!

Wiederholung

Wir haben heute:

- Kellerautomaten kennengelernt
- Zwei Konstruktionsmethoden kennengelernt
  - Etwas zählen
  - Eine Zeichenkette merken, die dann andersherum vom Stack gelesen werden kann
- Das Pumping Lemma kennengelernt
- Damit bewiesen, dass Kellerautomaten mächtiger sind als endliche Automaten.