## Konfigurationen

Wir wollen jetzt den Begriff der *Konfiguration* definieren:

**Definition:** Sei $M = (Z, \Sigma, \Gamma, \delta, z_0, \#)$ ein PDA. Als Konfiguration von $M$ bezeichnen wir jedes Element $k$ aus der Menge $Z \times \Sigma^* \times \Gamma^*$.

Die Konfiguration $(z, w, V)$ für $z \in Z$, $w \in \Sigma^*$, $V \in \Gamma^*$ wollen wir so interpretieren, dass der Kellerautomat sich im Zustand $z$ befindet, dass auf dem Eingabeband noch das Wort $w$ zu lesen ist (d.h. wenn $w = a_1 a_2 \dots a_n$ gilt, dann steht der Lesekopf auf $a_1$), und dass der Kellerinhalt $V$ ist, wobei für $V = A_1 A_2 \dots A_m$ das unterste Kellersymbol $A_m$ ist, während wir ganz oben $A_1$ sehen.

# Konfigurationsübergänge

Wie ändert sich die Konfiguration des PDA, wenn einer der in dieser Konfiguration möglichen Übergänge angewendet wird?

Die aktuelle Konfiguration sei $(z, a_1 \dots a_n, A_1 \dots A_m)$.

Möglich sind die Übergänge aus $\delta(z, a_1, A_1)$ und aus $\delta(z, \varepsilon, A_1)$.

Wir setzen die intuitive Idee des Übergangs formal um und erhalten für $(z', B_1 \dots B_k) \in \delta(z, a_1, A_1)$ als neue Konfiguration:

$$(z', a_2 \dots a_n, B_1 \dots B_k A_2 \dots A_m)$$

Wir notieren diesen Übergang auf Konfigurationen so:

$$(z, a_1 \dots a_n, A_1 \dots A_m) \quad \vdash \quad (z', a_2 \dots a_n, B_1 \dots B_k A_2 \dots A_m)$$

Entsprechend erhalten wir im Fall $(z', B_1 \dots B_k) \in \delta(z, \varepsilon, A_1)$:

$$(z, a_1 \dots a_n, A_1 \dots A_m) \quad \vdash \quad (z', a_1 \dots a_n, B_1 \dots B_k A_2 \dots A_m)$$

# Akzeptierung durch leeren Keller

Mit Hilfe der Konfigurationsbeschreibungen können wir nun festlegen, welches die durch den PDA $M$ akzeptierte Sprache ist. Die hier verwendete Art der Akzeptierung nennt man sinnvoller Weise

Akzeptierung durch leeren Keller:

$$N(M) = \{w \in \Sigma^* \mid \exists z \in Z: \ (z_0, w, \#) \vdash^* (z, \varepsilon, \varepsilon)\}$$

Beachte: Bei leerem Keller kann $M$ nicht mehr weiter arbeiten. Aber die ursprüngliche Eingabe gilt nur dann als akzeptiert, wenn sie zu diesem Zeitpunkt dann auch komplett gelesen worden ist.

Alternativ zur Akzeptierung durch leeren Keller kann man auch bei Kellerautomaten die Akzeptierung durch Endzustand definieren – wir werden das in den Übungen untersuchen.

# Beispiel $\{a^n b^n \mid n \geq 1\}$

Wir wollen einen Kellerautomat $M$ entwerfen, für den gilt:

$$N(M) = \{a^n b^n \mid n \geq 1\}$$

Hierfür werden wir weder Nichtdeterminismus noch $\varepsilon$-Übergänge benötigen.

$$\begin{aligned}
\delta(z_0, a, \#) &= \{(z_0, a)\} \\
\delta(z_0, a, a) &= \{(z_0, aa)\} \\
\delta(z_0, b, a) &= \{(z_1, \varepsilon)\} \\
\delta(z_1, b, a) &= \{(z_1, \varepsilon)\}
\end{aligned}$$

Für alle übrigen Tripel ist der Wert von $\delta$ die leere Menge.

Die Intuition sollte klar sein: Solange $a$'s in der Eingabe erscheinen, schieben wir sie einfach hinüber in den Keller. (Das sind die ersten zwei Übergänge der Liste oben.) Beim ersten $b$ in der Eingabe wechseln wir in den Zustand $z_1$ und dürfen von da an nur noch $b$'s lesen und gegen $a$'s *matchen*.

## Die Arbeit von $M$

Wir geben die Konfigurationsabfolge für die Rechnung des gerade definierten PDA auf dem Input *aaabbb* an:

$$(z_0, aaabbb, \#) \vdash (z_0, aabbb, a) \vdash (z_0, abbb, aa) \vdash (z_0, bbb, aaa) \\ \vdash (z_1, bb, aa) \vdash (z_1, b, a) \vdash (z_1, \varepsilon, \varepsilon)$$

Damit ist dieser Input akzeptiert.

Und was macht der PDA auf dem Input *aaabb*?

$$(z_0, aaabb, \#) \vdash (z_0, aabb, a) \vdash (z_0, abb, aa) \vdash (z_0, bb, aaa) \\ \vdash (z_1, b, aa) \vdash (z_1, \varepsilon, a)$$

Hier endet die Rechnung – der Input ist nicht akzeptiert!

Was passiert bei leerem Input?

$$(z_0, \varepsilon, \#)$$

Es gibt keinen 1. Schritt – der Input ist nicht akzeptiert!

## Noch ein Beispiel

Als letztes wollen wir die Arbeit unseres PDA auf dem Input *aaabbaa* notieren:

$$(z_0, aaabbaa, \#) \vdash (z_0, aabbaa, a) \vdash (z_0, abbaa, aa) \vdash (z_0, bbaa, aaa) \vdash (z_1, baa, aa) \vdash (z_1, aa, a)$$

Die Rechnung endet, obwohl der Keller noch nicht leer ist, denn es gilt $\delta(z_1, a, a) = \emptyset$. Die Eingabe ist abgelehnt.

Jetzt wollen wir beweisen, dass die PDAs genau die Typ-2 Sprachen beschreiben. Also:

Für jede Sprache $L \subseteq \Sigma^*$ gilt:

Es gibt einen PDA $M$ mit $N(M) = L$ genau dann, wenn es eine Typ-2 Grammatik $G$ mit $L(G) = L$ gibt.