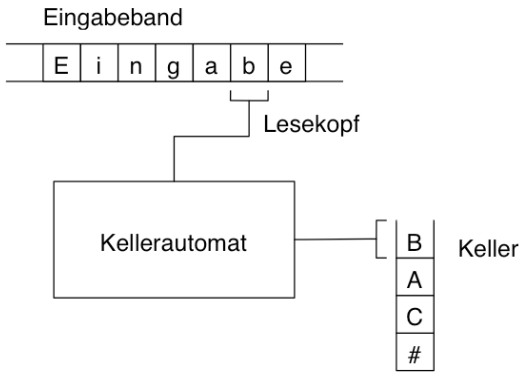
Grenzen endlicher Automaten

Der Kellerautomat

Bei der letzten Übungsaufgabe sind wir bei der Erstellung eines endlichen Automaten für die Sprache $L = \{a^n b^n | n \in \mathbb{N}\}$ an unsere Grenzen gestoßen.
Das wesentliche Hindernis war dabei, dass unsere bekannten Automaten nicht mitzählen können (d.h. für obiges Beispiel: sie können sich nicht merken, wie viele a's im ersten Teil des Wortes enthalten sind).
Eine Grammatik G konnten wir trotz allem entwickeln, auch wenn diese nicht der üblichen Gestalt entsprach – G war gegeben durch die Produktion S ::= aSb| ε.



Demzufolge ist es naheliegend, dass eine andere Art von Maschine existieren muss, die eine derartige Grammatik G nachbilden kann.
Man bräuchte dafür (einfach) ein weiteres unbegrenztes Band für Zwischenspeicherungen. Dazu definieren wir den so genannten

Kellerautomaten (KA = NKA)

Ein (nichtdeterministischer) Kellerautomat (NKA) besteht aus

- einer Zustandsmenge Z mit einem Anfangszustand Z_0 und mindestens einem Endzustand
- einem Eingabealphabet E (wobei das leere Wort Epsilon = ϵ auch erlaubt ist)
- einem Kelleralphabet K mit einem Kellerstartsymbol K_0 – wir einigen uns (mal) auf K_0 = #
- einer Übergangsfunktion f: (Z, E, K) → P(Z, K), die aus den aktuellen Werten von Zustand, Eingabe und oberstes Kellersymbol Folgezustände sowie Folge-Kellersymbole aus der Potenzmenge von (Z, K) bestimmt.

Es gilt also vereinfacht gesagt: **Kellerautomat = Akzeptor + Kellerband**.

Außerdem akzeptiert ein Kellerautomat genau dann eine Eingabe, wenn der Akzeptor nach Abarbeitung dieser in einen Endzustand gelangt - unabhängig von der Belegung des Kellerbands.

Übungen

1. Konstruiere in AtoCC einen Kellerautomaten, der die Sprache $L = \{a^n b^n | n \in \mathbb{N}\}$ akzeptiert.
2. Gib zwei Sprachen mit dem gleichen Eingabealphabet an (inkl. Grammatiken), deren Wörter jeweils durch einen endlichen Automaten akzeptiert werden. Konstruiere beide Automaten und zeige, dass es sowohl für die Vereinigung als auch für den Durchschnitt beider Sprachen wieder einen Akzeptor gibt – indem Du diese ebenfalls entwickelst!
3. Löse Aufgabe 2. für zwei Sprachen mit unterschiedlichem Eingabealphabet (und logischerweise nur für die Vereinigung beider Sprachen)!