Deterministische Kellerautomaten

Die zuletzt definierten Kellerautomaten (PDA) waren ein nichtdeterministisches Maschinenmodell. Es liegt nahe zu fragen, inwiefern sich die Beschreibungskraft des Modells ändert, wenn wir den Nichtdeterminismus vermeiden. Wir definieren daher:

Ein DPDA (deterministischer Kellerautomat) ist ein PDA mit folgenden Abweichungen vom ursprünglichen Modell:

1) In jeder Situation darf maximal ein Übergang möglich sein, d.h. $\forall a \in \Sigma, z \in Z, A \in \Gamma$:
$$|\delta(z, a, A)| + |\delta(z, \varepsilon, A)| \leq 1$$
2) Akzeptierung ist durch Endzustand definiert.

DPDA für $\{a^n b^n \mid n \geq 1\}$

Wie bei vielen anderen DPDAs für wichtige Beispielmengen genügt auch hier ein DPDA, der keine ε-Übergänge verwendet:

$$M = (\{z_0, z_1, z_2, z_a\}, \{a, b\}, \{a, \#\}, \delta, z_0, \#, \{z_a\})$$

Die letzte Komponente ist hier die Menge der akzeptierenden Endzustände, und δ wird folgendermaßen festgelegt:

$$\delta(z_0, a, \#) = \{(z_1, \#)\}$$
$$\delta(z_1, a, \#) = \{(z_1, a\#)\}$$
$$\delta(z_1, a, a) = \{(z_1, aa)\}$$

Wenn n mal ein a gelesen wird, hat der Keller schließlich den Inhalt $a^{n-1}\#$.

$$\delta(z_1, b, \#) = \{(z_a, \varepsilon)\}$$
$$\delta(z_1, b, a) = \{(z_2, \varepsilon)\}$$

Bei $n = 1$ führt ein b in den Endzustand.

$$\delta(z_2, b, a) = \{(z_2, \varepsilon)\}$$
$$\delta(z_2, b, \#) = \{(z_a, \varepsilon)\}$$

Bei $n > 1$ muss zuerst für jedes a ein b vom Eingabeband gelesen werden.

Akzeptierte Sprache eines DPDA

Im Beispiel konnte man sehen, dass es wichtig ist festzulegen, wann wir eine Eingabe als akzeptiert ansehen. Wir verlangen, dass *nach Lesen des gesamten Eingabeworts* ein Endzustand angenommen wird. Sei M ein DPDA mit Endzustandsmenge F.

$$N(M) := \{x \in \Sigma^* \mid \exists z \in F, V \in \Gamma^* : (z_0, x, \#) \vdash^* (z, \varepsilon, V)\}$$

Mit dieser Definition überprüfen wir nun, ob der Beispiel-DPDA korrekt arbeitet. Wir betrachten Inputs ab, $aabb$, $aaab$ und aba:

$$(z_0, ab, \#) \;\vdash\; (z_1, b, \#) \;\vdash\; (z_a, \varepsilon, \varepsilon)$$

$$(z_0, aabb, \#) \;\vdash\; (z_1, abb, \#) \;\vdash\; (z_1, bb, a\#) \;\vdash\; (z_2, b, \#) \;\vdash\; (z_a, \varepsilon, \varepsilon)$$

$$(z_0, aaab, \#) \;\vdash\; (z_1, aab, \#) \;\vdash\; (z_1, ab, a\#) \;\vdash\; (z_1, b, aa\#) \;\vdash\; (z_2, \varepsilon, a\#)$$

$$(z_0, aba, \#) \;\vdash\; (z_1, ba, \#) \;\vdash\; (z_a, a, \varepsilon) \quad - \quad \text{Ende der Rechnung}$$

Die Klasse DCFL

Die Klasse DCFL enthält alle Sprachen $L \subseteq \Sigma^*$, für die es einen DPDA M gibt mit $N(M) = L$.

Ein DEA ist ein DPDA, der seinen Keller nicht nutzt.

Daher gilt $REG \subseteq DCFL$.

Andererseits ist offenbar $DCFL$ eine Teilmenge von CFL, der Klasse aller Typ-2 Sprachen.
Zudem haben wir schon gesehen, dass $\{a^n b^n \mid n \geq 1\}$ in $DCFL$ liegt, aber wir hatten gezeigt, dass diese Sprache nicht regulär ist.
Somit ergibt sich:

$$REG \subsetneq DCFL \subseteq CFL$$

DCFL nicht abgeschlossen unter Schnitt

Wir hatten die Nichtabgeschlossenheit von CFL unter Durchschnitt mit Hilfe der beiden folgenden Sprachen gezeigt:

$$\{a^n b^n c^m \mid n, m \geq 1\} \quad \text{und} \quad \{a^m b^n c^n \mid n, m \geq 1\}$$

Aber diese beiden Sprachen gehören sogar zu DCFL!

Da ihr Durchschnitt nicht einmal Typ-2 ist, gehört er insbesondere nicht zu DCFL, und daher erhalten wir:

Die Klasse DCFL ist nicht abgeschlossen gegen Durchschnitt.

Gibt es Typ-2 Sprachen, die nicht zu DCFL gehören?

Die Antwort ist JA, z.B. die Menge der Palindrome ist in CFL, aber nicht in DCFL.

Markierte Palindrome

Während also die Menge $\{w \in \Sigma^* \mid w = w^R\}$ nicht in DCFL (wohl aber in CFL) liegt, können wir die Menge der markierten Palindrome, d.h.

$$L_1 = \{w\$w^R \mid w \in \Sigma^*\}$$

mit einem DPDA erkennen.
(Hierbei ist $\$ \notin \Sigma$ ein sonst nicht benutztes Eingabesymbol.)

Entwerfen Sie für $\Sigma = \{a, b, c\}$ einen DPDA, der die obige Sprache L_1 erkennt.