Einführung in die Theoretische Informatik

Johannes Köbler

Institut für Informatik
Humboldt-Universität zu Berlin

WS 2011/12

Deterministische Kellerautomaten

Von besonderem Interesse sind kontextfreie Sprachen, die von einem deterministischen Kellerautomaten erkannt werden.

Definition

- Ein Kellerautomat heißt deterministisch, falls $\vdash$ eine rechtseindeutige Relation ist:

$$K \vdash K_1 \wedge K \vdash K_2 \Rightarrow K_1 = K_2.$$

- Äquivalent hierzu ist, dass die Überführungsfunktion δ für alle $(q, a, A) \in Z \times \Sigma \times \Gamma$ folgende Bedingung erfüllt (siehe Übungen):

$$\|\delta(q, a, A)\| + \|\delta(q, \varepsilon, A)\| \leq 1.$$

Deterministische Kellerautomaten

Beispiel

- Betrachte den PDA $M = (\{q_0, q_1, q_2\}, \{a, b, c\}, \{A, B, \#\}, \delta, q_0, \#)$ mit

$$\delta\colon \quad \begin{array}{llll} q_0 a\# \to q_0 A\# & q_0 b\# \to q_0 B\# & q_0 aA \to q_0 AA & q_0 bA \to q_0 BA \\ q_0 aB \to q_0 AB & q_0 bB \to q_0 BB & q_0 cA \to q_1 A & q_0 cB \to q_1 B \\ q_1 aA \to q_1 & q_1 bB \to q_1 & q_1 \varepsilon\# \to q_2 & \end{array}$$

  Darstellung von δ in Tabellenform

δ	$q_0, \#$	q_0, A	q_0, B	$q_1, \#$	q_1, A	q_1, B	$q_2, \#$	q_2, A	q_2, B
ε	–	–	–	q_2	–	–	–	–	–
a	$q_0 A\#$	$q_0 AA$	$q_0 AB$	–	q_1	–	–	–	–
b	$q_0 B\#$	$q_0 BA$	$q_0 BB$	–	–	q_1	–	–	–
c	–	$q_1 A$	$q_1 B$	–	–	–	–	–	–

- Man beachte, dass jedes Tabellenfeld höchstens eine Anweisung enthält und jede Spalte mit einem ε-Übergang keine weiteren Anweisungen enthält.
- Daher ist die Bedingung $\|\delta(q, a, A)\| + \|\delta(q, \varepsilon, A)\| \leq 1$ für alle $q \in Z$, $a \in \Sigma$ und $A \in \Gamma$ erfüllt. $\triangleleft$

Deterministische Kellerautomaten

Frage

- Können deterministische Kellerautomaten zumindest alle regulären Sprachen durch Leeren des Kellers akzeptieren?
- Kann z.B. die Sprache $L = \{a, aa\}$ von einem deterministischen Kellerautomaten M durch Leeren des Kellers akzeptiert werden?

Antwort: Nein

- Um $x = a$ zu akzeptieren, muss M den Keller nach Lesen von a leeren und kann somit keine anderen Wörter mit dem Präfix a akzeptieren.
- Deterministische Kellerautomaten können also durch Leeren des Kellers nur präfixfreie Sprachen L akzeptieren (d.h. kein Wort $x \in L$ ist Präfix eines anderen Wortes in L).

Lösung des Problems

Wir vereinbaren, dass deterministische Kellerautomaten ihre Eingabe durch Erreichen eines Endzustands akzeptieren dürfen.

Deterministische Kellerautomaten

Definition

- Ein Kellerautomat mit Endzuständen wird durch ein 7-Tupel $M = (Z, \Sigma, \Gamma, \delta, q_0, \#, E)$ beschrieben.
- Dabei sind die Komponenten $Z, \Sigma, \Gamma, \delta, q_0, \#$ wie bei einem PDA.
- Zusätzlich ist $E \subseteq Z$ eine Menge von Endzuständen.
- Die von M akzeptierte oder erkannte Sprache ist

$$L(M) = \{x \in \Sigma^* \mid \exists\, p \in E\, \exists\, \alpha \in \Gamma^* \colon (q_0, x, \#) \vdash^* (p, \varepsilon, \alpha)\}.$$

- M ist ein det. Kellerautomat mit Endzuständen (kurz: DPDA), falls M für alle $(q, a, A) \in Z \times \Sigma \times \Gamma$ zusätzlich folgende Bedingung erfüllt:

$$\|\delta(q, a, A)\| + \|\delta(q, \varepsilon, A)\| \leq 1.$$

- Weiter sei

$$\mathrm{DCFL} = \{L(M) \mid M \text{ ist ein DPDA}\}$$

(*deterministic context free languages*).

Charakterisierung von DCFL mittels Grammatiken

- Die Klasse DCFL lässt sich auch mit Hilfe von speziellen kontextfreien Grammatiken charakterisieren, den so genannten $LR(k)$-Grammatiken.
- Der erste Buchstabe L steht für die Leserichtung bei der Syntaxanalyse, d.h. das Eingabewort x wird von links (nach rechts) gelesen.
- Der zweite Buchstabe R bedeutet, dass bei der Syntaxanalyse eine Rechtsableitung entsteht.
- Schließlich gibt der Parameter k an, wieviele Zeichen man in der Eingabe vorauslesen muss, damit die nächste anzuwendende Regel eindeutig feststeht (k wird auch als *Lookahead* bezeichnet).
- Durch $LR(0)$-Grammatiken lassen sich nur die präfixfreien Sprachen in DCFL erzeugen.
- Dagegen erzeugen die $LR(k)$-Grammatiken für jedes $k \geq 1$ genau die Sprachen in DCFL.
- Daneben gibt es noch $LL(k)$-Grammatiken, die für wachsendes k immer mehr deterministisch kontextfreie Sprachen erzeugen.

Abschlusseigenschaften von DCFL

Frage

Ist DCFL unter Komplementbildung abgeschlossen?

Antwort

Ja. Allerdings ergeben sich beim Versuch, einfach die End- und Nichtendzustände eines DPDA M zu vertauschen, um einen DPDA $\overline{M}$ für $\overline{L(M)}$ zu erhalten, folgende Schwierigkeiten:

1. Falls M eine Eingabe x nicht zu Ende liest, wird x weder von M noch von $\overline{M}$ akzeptiert.
2. Falls M nach dem Lesen von x noch ε-Übergänge ausführt und dabei End- und Nichtendzustände besucht, wird x von M und von $\overline{M}$ akzeptiert.

DPDAs, die ihre Eingabe zu Ende lesen

Der nächste Satz zeigt, wie sich Problem 1 beheben lässt.

Satz

Jede Sprache $L \in \mathrm{DCFL}$ wird von einem DPDA M' erkannt, der alle Eingaben zu Ende liest.

Beweis.

Sei $M = (Z, \Sigma, \Gamma, \delta, q_0, \#, E)$ ein DPDA mit $L(M) = L$.
Falls M eine Eingabe $x = x_1 \cdots x_n$ nicht zu Ende liest, muss einer der folgenden drei Gründe vorliegen:

1. M gerät in eine Konfiguration $(q, x_i \cdots x_n, \varepsilon)$, $i \leq n$, mit leerem Keller.
2. M gerät in eine Konfiguration $(q, x_i \cdots x_n, A\gamma)$, $i \leq n$, in der wegen $\delta(q, x_i, A) = \delta(q, \varepsilon, A) = \emptyset$ keine Anweisung ausführbar ist.
3. M gerät in eine Konfiguration $(q, x_i \cdots x_n, A\gamma)$, $i \leq n$, so dass M ausgehend von (q, ε, A) eine unendliche Folge von ε-Anweisungen ausführt.

DPDAs, die ihre Eingabe zu Ende lesen

Beweis (Fortsetzung)

- Die erste Ursache schließen wir aus, indem wir ein neues Zeichen $\square$ auf dem Kellerboden platzieren:

  (a) $s\varepsilon\# \rightarrow q_0\#\square$ (dabei sei s ein neuer Startzustand).

- Die zweite Ursache schließen wir durch Hinzunahme eines Fehlerzustands f sowie folgender Anweisungen aus:

  (b) $qaA \rightarrow fA$, für alle $(q, a, A) \in Z \times \Sigma \times \Gamma'$ mit $A = \square$ oder $\delta(q, a, A) = \delta(q, \varepsilon, A) = \emptyset$ (hierbei ist $\Gamma' = \Gamma \cup \{\square\}$),

  (c) $faA \rightarrow fA$, für alle $a \in \Sigma$ und $A \in \Gamma'$.

DPDAs, die ihre Eingabe zu Ende lesen

Beweis (Fortsetzung)

- Als nächstes verhindern wir die Ausführung einer unendlichen Folge von ε-Übergängen.

  Dabei unterscheiden wir die beiden Fälle, ob M hierbei auch Endzustände besucht oder nicht.

  (d) $q\varepsilon A \rightarrow fA$, für alle $q \in Z$ und $A \in \Gamma$, so dass M ausgehend von der Konfiguration (q, ε, A) unendlich viele ε-Übergänge ausführt ohne dabei einen Endzustand zu besuchen.

  Falls ja, sehen wir einen Umweg über den neuen Endzustand e vor.

  (e) $q\varepsilon A \rightarrow eA$, $e\varepsilon A \rightarrow fA$, für alle $q \in Z$ und $A \in \Gamma$, so dass M ausgehend von der Konfiguration (q, ε, A) unendlich viele ε-Übergänge ausführt und dabei auch Endzustände besucht.

- Schließlich übernehmen wir von M noch

  (f) alle Anweisungen aus δ, soweit sie nicht durch Anweisungen vom Typ (d) oder (e) überschrieben wurden.

DPDAs, die ihre Eingabe zu Ende lesen

Beweis (Schluss)

Zusammenfassend transformieren wir $M = (Z, \Sigma, \Gamma, \delta, q_0, \#, E)$ in den DPDA

$$M' = (Z \cup \{s, e, f\}, \Sigma, \Gamma', \delta', s, \#, E \cup \{e\}) \text{ mit } \Gamma' = \Gamma \cup \{\square\},$$

wobei δ' folgende Anweisungen enthält:

(a) $s\varepsilon\# \to q_0\#\square$,

(b) $qaA \to fA$, für alle $(q, a, A) \in Z \times \Sigma \times \Gamma'$ mit $A = \square$ oder $\delta(q, a, A) = \delta(q, \varepsilon, A) = \emptyset$,

(c) $faA \to fA$, für alle $a \in \Sigma$ und $A \in \Gamma'$,

(d) $q\varepsilon A \to fA$, für alle $q \in Z$ und $A \in \Gamma$, so dass ausgehend von der Konfiguration (q, ε, A) unendlich viele ε-Übergänge ausgeführt werden, ohne dass dabei ein Endzustand besucht wird.

(e) $q\varepsilon A \to eA$
$e\varepsilon A \to fA$, für alle $q \in Z$ und $A \in \Gamma$, so dass ausgehend von der Konfiguration (q, ε, A) unendlich viele ε-Übergänge ausgeführt und dabei auch Endzustände besucht werden,

(f) alle Anweisungen aus δ, soweit sie nicht durch Anweisungen vom Typ (c) oder (e) überschrieben wurden.

$\square$

DPDAs, die ihre Eingabe zu Ende lesen

Beispiel

Wenden wir diese Konstruktion auf den DPDA

$$M = (\{q_0, q_1, q_2\}, \{a, b, c\}, \{A, B, \#\}, \delta, q_0, \#, \{q_2\})$$

mit der Überführungsfunktion

δ	$q_0, \#$	q_0, A	q_0, B	$q_1, \#$	q_1, A	q_1, B	$q_2, \#$	q_2, A	q_2, B
ε	–	–	–	q_2	–	–	$q_2\#$	–	–
a	$q_0A\#$	q_0AA	q_0AB	–	q_1	–	–	–	–
b	$q_0B\#$	q_0BA	q_0BB	–	–	q_1	–	–	–
c	–	q_1A	q_1B	–	–	–	–	–	–

an, so erhalten wir den DPDA

$$M' = (\{q_0, q_1, q_2, s, e, f\}, \{a, b, c\}, \{A, B, \#, \square\}, \delta', s, \#, \{q_2, e\})$$

mit folgender Überführungsfunktion δ':

DPDAs, die ihre Eingabe zu Ende lesen

Beispiel (Schluss)

δ'	$q_0,\#$	q_0,A	q_0,B	$q_0,\square$	$q_1,\#$	q_1,A	q_1,B	$q_1,\square$	$q_2,\#$	q_2,A	q_2,B	$q_2,\square$
ε	–	–	–	–	q_2	–	–	–	$e\#$	–	–	–
a	$q_0A\#$	q_0AA	q_0AB	$f\square$	–	q_1	fB	$f\square$	–	fA	fB	$f\square$
b	$q_0B\#$	q_0BA	q_0BB	$f\square$	–	fA	q_1	$f\square$	–	fA	fB	$f\square$
c	$f\#$	q_1A	q_1B	$f\square$	–	fA	fB	$f\square$	–	fA	fB	$f\square$
Typ	(f,b)	(f)	(f)	(b)	(f)	(f,b)	(f,b)	(b)	(e)	(b)	(b)	(b)

	$s,\#$	s,A	s,B	$s,\square$	$f,\#$	f,A	f,B	$f,\square$	$e,\#$	e,A	e,B	$e,\square$
ε	$q_0\#\square$	–	–	–	–	–	–	–	$f\#$	–	–	–
a	–	–	–	–	$f\#$	fA	fB	$f\square$	–	–	–	–
b	–	–	–	–	$f\#$	fA	fB	$f\square$	–	–	–	–
c	–	–	–	–	$f\#$	fA	fB	$f\square$	–	–	–	–
Typ	(a)				(c)	(c)	(c)	(c)	(e)			

Komplementabschluss von DCFL

Satz

Die Klasse DCFL ist unter Komplement abgeschlossen.

Beweis

- Sei $M = (Z, \Sigma, \Gamma, \delta, q_0, \#, E)$ ein DPDA, der alle Eingaben zu Ende liest, und sei $L(M) = L$.
- Wir konstruieren einen DPDA $\overline{M}$ für $\overline{L}$, der M simuliert.
- Dabei merkt sich $\overline{M}$ in seinem Zustand (q, i) neben dem aktuellen Zustand q von M in der Komponente i, ob M nach Lesen des letzten Zeichens (bzw. seit Rechenbeginn) einen Endzustand besucht hat $(i = 2)$ oder nicht $(i = 1)$.
- Möchte M das nächste Zeichen lesen und befindet sich $\overline{M}$ im Zustand $(q, 1)$, so macht $\overline{M}$ noch einen Umweg über den Endzustand $(q, 3)$.

Komplementabschluss von DCFL

Beweis (Schluss)

- Konkret sei $\overline{M} = (Z \times \{1,2,3\}, \Sigma, \Gamma, \delta', s, \#, Z \times \{3\})$ mit

$$s = \begin{cases} (q_0, 1), & q_0 \notin E, \\ (q_0, 2), & \text{sonst,} \end{cases}$$

wobei δ' für jede Anweisung $q\varepsilon A \rightarrow_M p\gamma$ die Anweisungen

$$\begin{aligned} &(q,1)\varepsilon A \rightarrow (p,1)\gamma, \quad \text{falls } p \notin E, \\ &(q,1)\varepsilon A \rightarrow (p,2)\gamma, \quad \text{falls } p \in E \text{ und} \\ &(q,2)\varepsilon A \rightarrow (p,2)\gamma, \end{aligned}$$

sowie für jede Anweisung $qaA \rightarrow_M p\gamma$ folgende Anweisungen enthält:

$$\begin{aligned} &(q,1)\varepsilon A \rightarrow (q,3)A, \\ &(q,2)aA \rightarrow (p,1)\gamma, \quad \text{falls } p \notin E, \\ &(q,2)aA \rightarrow (p,2)\gamma, \quad \text{falls } p \in E, \\ &(q,3)aA \rightarrow (p,1)\gamma, \quad \text{falls } p \notin E \text{ und} \\ &(q,3)aA \rightarrow (p,2)\gamma, \quad \text{falls } p \in E. \end{aligned}$$

Man beachte, dass $\overline{M}$ in einem Endzustand keine ε-Übergänge macht. $\square$

Komplementabschluss von DCFL

Beispiel

- Angenommen, ein DPDA $M = (Z, \Sigma, \Gamma, \delta, q_0, \#, E)$ führt bei Eingabe $x = a$ folgende Rechnung aus:

$$(q_0, a, \#) \vdash (q_1, \varepsilon, \gamma_1) \vdash (q_2, \varepsilon, \gamma_2).$$

- Dann würde $\overline{M}$ im Fall $E = \{q_0, q_2\}$ (d.h. $x \in L(M)$) die Rechnung

$$((q_0, 2), a, \#) \vdash ((q_1, 1), \varepsilon, \gamma_1) \vdash ((q_2, 2), \varepsilon, \gamma_2)$$

  ausführen und das Wort a verwerfen, da $(q_1, 1), (q_2, 2) \notin Z \times \{3\}$ sind.

- Im Fall $E = \{q_0\}$ (d.h. $x \notin L(M)$) würde $\overline{M}$ dagegen die Rechnung

$$((q_0, 2), a, \#) \vdash ((q_1, 1), \varepsilon, \gamma_1) \vdash ((q_2, 1), \varepsilon, \gamma_2) \vdash ((q_2, 3), \varepsilon, \gamma_2)$$

  ausführen und das Wort a akzeptieren, da $(q_2, 3) \in Z \times \{3\}$ ist. ◁

Komplementabschluss von DCFL

Definition

Für eine Sprachklasse $\mathcal{C}$ bezeichne co-$\mathcal{C}$ die Klasse $\{\bar{L} \mid L \in \mathcal{C}\}$ aller Komplemente von Sprachen in $\mathcal{C}$.

Korollar

- REG $=$ co-REG,
- DCFL $=$ co-DCFL,
- CFL $\neq$ co-CFL.

Weitere Abschlusseigenschaften von DCFL

Satz

Die Klasse DCFL ist nicht abgeschlossen unter Durchschnitt, Vereinigung, Produkt und Sternhülle.

$A, B \in \mathrm{DCFL} \not\Rightarrow A \cap B \in \mathrm{DCFL}$

- Die beiden Sprachen

  $$L_1 = \{a^n b^m c^m \mid n, m \geq 0\} \quad \text{und} \quad L_2 = \{a^n b^n c^m \mid n, m \geq 0\}$$

  sind sogar deterministisch kontextfrei (siehe Übungen).
- Da $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$ nicht kontextfrei ist, liegt der Schnitt dieser Sprachen natürlich auch nicht in DCFL.

$A, B \in \mathrm{DCFL} \not\Rightarrow A \cup B \in \mathrm{DCFL}$

- Da DCFL unter Komplementbildung abgeschlossen ist, kann DCFL wegen de Morgan dann auch nicht unter Vereinigung abgeschlossen sein.
- Beispielsweise sind folgende Sprachen deterministisch kontextfrei:

  $$L_3 = \{a^i b^j c^k \mid i \neq j\} \text{ und } L_4 = \{a^i b^j c^k \mid j \neq k\}.$$

- Ihre Vereinigung gehört aber nicht zu DCFL, d.h.

  $$L_3 \cup L_4 = \{a^i b^j c^k \mid i \neq j \text{ oder } j \neq k\} \in \mathrm{CFL} \setminus \mathrm{DCFL}.$$

- DCFL ist nämlich unter Schnitt mit regulären Sprachen abgeschlossen (siehe Übungen).
- Daher wäre mit $L_3 \cup L_4$ auch die Sprache

  $$(\overline{L_3 \cup L_4}) \cap L(a^* b^* c^*) = \{a^n b^n c^n \mid n \geq 0\}$$

  (deterministisch) kontextfrei.

$A, B \in \text{DCFL} \not\Rightarrow AB \in \text{DCFL}$

- Betrachte die DCFL Sprachen
$$L_0 = \{0\},\ L_3 = \{a^i b^j c^k \mid i \neq j\} \text{ und } L_4 = \{a^i b^j c^k \mid j \neq k\}.$$
- Wir wissen bereits, dass $L_3 \cup L_4 \notin \text{DCFL}$ ist.
- Dann ist aber auch die Sprache
$$L_5 = L_0(L_3 \cup L_4) = \{0a^i b^j c^k \mid i \neq j \vee j \neq k\} \notin \text{DCFL},$$
da sich ein DPDA $M = (Z, \Sigma, \Gamma, \delta, q_0, \#, E)$ für L_5 leicht zu einem DPDA M' für $L_3 \cup L_4$ umbauen ließe:
  - Sei (p, ε, γ) die Konfiguration, die M nach Lesen der Eingabe 0 erreicht.
  - Dann erkennt der DPDA $M' = (Z \cup \{s\}, \Sigma, \Gamma, \delta', s, \#, E)$ die Sprache $L_3 \cup L_4$, wobei δ' wie folgt definiert ist:
$$\delta'(q, u, A) = \begin{cases} (p, \gamma), & (q, u, A) = (s, \varepsilon, \#), \\ \delta(q, u, A), & (q, u, A) \in Z \times (\Sigma \cup \{\varepsilon\}) \times \Gamma. \end{cases}$$

$A, B \in \mathrm{DCFL} \not\Rightarrow AB \in \mathrm{DCFL}$

- Betrachte die DCFL Sprachen

$$L_0 = \{0\},\ L_3 = \{a^i b^j c^k \mid i \neq j\} \text{ und } L_4 = \{a^i b^j c^k \mid j \neq k\}.$$

- Es ist leicht zu sehen, dass auch die beiden Sprachen L_0^* und $L = L_0 L_3 \cup L_4$ in DCFL sind (siehe Übungen).
- Ihr Produkt $L_0^* L$ gehört aber nicht zu DCFL.
- Da DCFL unter Schnitt mit regulären Sprachen abgeschlossen ist, wäre andernfalls auch die Sprache

$$L_0^* L \cap L_0 L(a^* b^* c^*) = \{0a^i b^j c^k \mid i \neq j \vee j \neq k\} = L_0(L_3 \cup L_4) = L_5$$

in DCFL, was wir bereits ausgeschlossen haben.

Bemerkung

Dass DCFL auch nicht unter Sternhüllenbildung abgeschlossen ist, lässt sich ganz ähnlich zeigen (siehe Übungen).

Abschlusseigenschaften

Abschlusseigenschaften der Klassen REG, DCFL und CFL

	Vereinigung	Schnitt	Komplement	Produkt	Sternhülle
REG	*ja*	*ja*	*ja*	*ja*	*ja*
DCFL	*nein*	*nein*	*ja*	*nein*	*nein*
CFL	*ja*	*nein*	*nein*	*ja*	*ja*