SS 2015

Zentralübung zur Vorlesung Theoretische Informatik

Dr. Werner Meixner

Fakultät für Informatik
TU München

http://www14.in.tum.de/lehre/2015SS/theo/uebung/

11. Juni 2015

ZÜ VI

Übersicht:

1. Übungsbetrieb — Fragen, Probleme?
2. Thema — Earley's Algorithmus
   Beispiel
3. Vorbereitung — TA Blatt 6

1. Fragen, Anregungen?

Aktuelle Fragen?

2. Thema: Earley's Algorithmus

Wir betrachten zunächst noch einmal die Definitionen aus der Vorlesung und besprechen anschließend Vorbereitungsaufgabe 2.

2.1 Definitionen und Algorithmus

Sei $x = x_1 \cdots x_n \in \Sigma^+$ gegeben.

Definition

Wir definieren

$$[iAj[\alpha_1 \cdots \alpha_k . \alpha_{k+1} \cdots \alpha_r \,,$$

falls G die Produktion

$$A \to \alpha_1 \cdots \alpha_r$$

enthält und, falls $j > i$, dann $k > 0$ und

$$\alpha_1 \cdots \alpha_k \to^* x_i \cdots x_{j-1} \,.$$

Wir nennen Objekte der soeben definierten Art t-Ableitung.

Bemerkung

t-Ableitungen sind einerseits syntaktische Objekte,
sie bedeuten aber Aussagen.

Der folgende Algorithmus von Earley leitet Aussagen her über die abschnittweise Herleitbarkeit von Eingabewörtern!

Earley's Algorithmus

$S_1 := \{[1S1[.\alpha;\ \alpha$ ist rechte Seite einer S-Produktion$\}$
for $j := 1$ **to** n **do**
führe folgende Schritte so oft wie möglich aus:
if $[iAj[\alpha_1 \cdots \alpha_k.B\alpha_{k+2} \cdots \alpha_r \in S_j$ **then**
füge für jede B-Produktion $B \to \beta$ die t-Ableitung
$[jBj[.\beta$ zu S_j hinzu (falls noch nicht dort)
if $[iAj[\alpha_1 \cdots \alpha_r. \in S_j$ **then**
füge für jede t-Ableitung $[lBi[\beta_1 \cdots \beta_k.A\beta_{k+2} \cdots \beta_r$ die
t-Ableitung $[lBj[\beta_1 \cdots \beta_k A.\beta_{k+2} \cdots \beta_r$ zu S_j hinzu
if $[jAj[.a \in S_j$ **and** $x_j = a$ **then**
füge zu S_{j+1} die t-Ableitung $[jAj+1[a.$ hinzu
if $S_{j+1} = \emptyset$ **then return** $x \notin L$
od
if S_{n+1} enthält t-Ableitung der Form $[1Sn+1[\alpha.$, α rechte Seite einer S-Produktion **then return** $x \in L$

Die drei Schritte in der Laufschleife werden auch als

predictor, completer und scanner

bezeichnet.

Der Predictor zeigt an, dass als nächstes ein neu gefundene Variable vorliegt und bereitet die Startaussage für die entsprechenden Produktionen vor.

Der Completer nimmt gefundene Ersetzungsmöglichkeiten zur Kenntnis.

Der Scanner nimmt das nächste Eingabezeichen zur Kenntnis.

Wir sehen uns nun die einzelnen Schritte des Earley-Algorithmus genauer an.

Earley's Algorithmus

$S_1 := \{[1S1[.\alpha;\ \alpha$ ist rechte Seite einer S-Produktion$\}$

for $j := 1$ **to** n **do**

führe folgende Schritte so oft wie möglich aus:

if $[iAj[\alpha_1 \cdots \alpha_k.B\alpha_{k+2} \cdots \alpha_r \in S_j$ **then**

füge für jede B-Produktion $B \to \beta$ die t-Ableitung $[jBj[.\beta$ zu S_j hinzu (falls noch nicht dort)

if $[iAj[\alpha_1 \cdots \alpha_r. \in S_j$ **then**

füge für jede t-Ableitung $[lBi[\beta_1 \cdots \beta_k.A\beta_{k+2} \cdots \beta_r$ die t-Ableitung $[lBj[\beta_1 \cdots \beta_k A.\beta_{k+2} \cdots \beta_r$ zu S_j hinzu

if $[jAj[.a \in S_j$ **and** $x_j = a$ **then**

füge zu S_{j+1} die t-Ableitung $[jAj+1[a.$ hinzu

if $S_{j+1} = \emptyset$ **then return** $x \notin L$

od

if S_{n+1} enthält t-Ableitung der Form $[1Sn+1[\alpha.$, α rechte Seite einer S-Produktion **then return** $x \in L$

2.2 Beispiel VA 2

Die Produktionen einer kontextfreien Grammatik $G = (V, \Sigma, P, S)$ seien gegeben durch

$$S \to A\,, \qquad A \to E \mid E + A \mid E - (A)\,,$$
$$E \to P \mid P \times E \mid E/P\,, \qquad P \to (A) \mid a\,.$$

Zeigen Sie durch Anwendung von Earley's Algorithmus, dass $a \times a - (a + a) \in L(G)$ gilt.

Zur Orientierung überlegen wir uns vorab eine Linksableitung von
$x = a \times a - (a + a) \in L(G)$
und geben den entsprechenden Syntaxbaum an.

$$
\begin{aligned}
S &\rightarrow A \rightarrow E-(A) \rightarrow P\times E-(A) \\
&\rightarrow a\times E-(A) \rightarrow a\times P-(A) \rightarrow a\times a-(A) \\
&\rightarrow a\times a-(E+A) \rightarrow a\times a-(P+A) \rightarrow a\times a-(a+A) \\
&\rightarrow a\times a-(a+E) \rightarrow a\times a-(a+P) \rightarrow a\times a-(a+a)
\end{aligned}
$$

Mann kann leicht zeigen, dass es keine weitere Linksableitung gibt.

Nun führen wir den Earley-Algorithmus mit der Eingabesequenz $a \times a - (a + a) \in L(G)$ aus.

Wir leiten nur die notwendigen Aussagen ab.
Beachten Sie dabei die unterschiedlichen farbigen Markierungen.
Die Behandlung der Operationszeichen und Klammern wird etwas verkürzt dargestellt, wie auch in der Vorlesung so geschehen.

Eingabesequenz $x = a \times a - (a + a) \in L(G)$;
Initialisierung S_1 : $[1S1[.A$;

S_1 : $[1A1[.E-(A)$ ∎ $[1E1[.P\times E$ ∎ $[1P1[.a$ ∎ $\cdots$

S_2 : $[1P2[a.$ ∎ $[1E2[P.\times E$ ∎ $\cdots$

S_3 : $[1E3[P\times .E$ ∎ $[3E3[.P$ ∎ $[3P3[.a$ ∎ $\cdots$

S_4 : $[3P4[a.$ ∎ $[3E4[P.$ ∎ $[1E4[P\times E.$ ∎ $[1A4[E.-(A)$ ∎ $\cdots$

S_5 : $[1A5[E-.(A)$ ∎ $\cdots$

S_6 : $[1A6[E-(.A)$ ∎ $[6A6[.E+A$ ∎ $[6E6[.P$ ∎ $[6P6[.a$ ∎ $\cdots$

S_7 : $[6P7[a.$ ∎ $[6E7[P.$ ∎ $[6A7[E.+A$ ∎ $\cdots$

S_8 : $[6A8[E+.A$ ∎ $[8A8[.E$ ∎ $[8E8[.P$ ∎ $[8P8[.a$ ∎ $\cdots$

S_9 : $[8P9[a.$ ∎ $[8E9[P.$ ∎ $[8A9[E.$ ∎
$[6A9[E+A.$ ∎ $[1A9[E-(A.)$ ∎ $\cdots$

S_{10} : $[1A10[E-(A).$ ∎ $[1S10[A.$ ∎ $\cdots$ (Rückgabe $x \in L$).

3. Vorbereitung TA Blatt 8

3.1 VA 1

Seien $L_1, L_2 \subseteq \Sigma^*$. Zeigen Sie:

Wenn L_1 kontextfrei ist und L_2 regulär, dann ist $L_1 \cap L_2$ kontextfrei.

Hinweis: Konstruieren sie aus einem PDA und einem DFA/NFA einen neuen PDA.

Seien
$M_1 = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$ ein PDA mit $L_F(M_1) = L_1$, und
$M_2 = (Q', \Sigma, \delta', q_0', F')$ ein DFA mit $L(M_2) = L_2$.

Wir konstruieren einen neuen PDA
$M'' = (Q \times Q', \Sigma, \Gamma, (q_0, q_0'), Z_0, \delta'', F \times F')$,
dessen Übergangsfunktion δ'' wie folgt definiert ist:

$$\delta''((q, q'), a, Z) = \{((p, \delta'(q', a)), \gamma)\,;\, (p, \gamma) \in \delta(q, a, Z)\} \quad \forall a \in \Sigma$$
$$\delta''((q, q'), \epsilon, Z) = \{((p, q'), \gamma)\,;\, (p, \gamma) \in \delta(q, \epsilon, Z)\}$$

Man kann diesen Automaten als das Produkt aus einem DFA und einem Kellerautomaten ansehen. Der neue Automat akzeptiert ein Wort w genau dann, wenn beide Teilautomaten w akzeptieren.

3.2 VA 3

Siehe Thema

3.3 VA 3

Beantworten Sie kurz die folgenden Fragen:

1. Gibt es eine Turingmaschine, die sich nie mehr als vier Schritte vom Startzustand entfernt und eine unendliche Sprache akzeptiert? Begründung!
2. Welche Sprachen lassen sich mit Turingmaschinen, die ihren Kopf immer nur nach rechts bewegen, erkennen?
3. Gibt es für jede Turingmaschine T eine Turingmaschine T' mit nur einem Zustand, die die Sprache von T akzeptiert?

Definition

Eine nichtdeterministische Turingmaschine (kurz TM oder NDTM) wird durch ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ beschrieben, das folgende Bedingungen erfüllt:

1. Q ist eine endliche Menge von Zuständen.
2. Σ ist eine endliche Menge, das Eingabealphabet.
3. Γ ist eine endliche Menge, das Bandalphabet, mit $\Sigma \subseteq \Gamma$
4. $\delta : Q \times \Gamma \to \mathcal{P}_f(Q \times \Gamma \times \{L, R, N\})$ ist die Übergangsfunktion.
5. $q_0 \in Q$ ist der Startzustand.
6. $\square \in \Gamma \setminus \Sigma$ ist das Leerzeichen.
7. $F \subseteq Q$ ist die Menge der (akzeptierenden) Endzustände.

Eine Turingmaschine heißt deterministisch, falls gilt

$$|\delta(q, a)| \leq 1 \qquad \text{für alle } q \in Q, a \in \Gamma.$$

Definition

Eine Konfiguration einer Turingmaschine ist ein Tupel $(\alpha, q, \beta) \in \Gamma^* \times Q \times \Gamma^*$.

Das Wort $w = \alpha\beta$ entspricht dem Inhalt des Bandes, wobei dieses rechts und links von w mit dem Leerzeichen $\square$ gefüllt sei. Der Schreib-/Lesekopf befindet sich auf dem ersten Zeichen von $\beta\square^\infty$.

Die Startkonfiguration der Turingmaschine bei Eingabe $x \in \Sigma^*$ entspricht der Konfiguration

$$(\epsilon, q_0, x)\,,$$

d.h. auf dem Band befindet sich genau die Eingabe $x \in \Sigma^*$, der Schreib-/Lesekopf befindet sich über dem ersten Zeichen der Eingabe und die Maschine startet im Zustand q_0.

Definition

Die von einer Turingmaschine M akzeptierte Sprache ist

$$L(M) = \{x \in \Sigma^*;\ (\epsilon, q_0, x) \rightarrow^* (\alpha, q, \beta) \text{ mit } q \in F, \alpha, \beta \in \Gamma^*\}$$

VA 3

1. Gibt es eine Turingmaschine, die sich nie mehr als vier Schritte vom Startzustand entfernt und eine unendliche Sprache akzeptiert? Begründung!

Lösung

Ja, die Turingmaschine kann einfach unabhängig von der Eingabe direkt in einen akzeptierenden Zustand wechseln.

2. Welche Sprachen lassen sich mit Turingmaschinen, die ihren Kopf immer nur nach rechts bewegen, erkennen?

Lösung

Antwort: Die regulären Sprachen.

Da der Kopf sich nur nach rechts bewegt, liest eine solche Turingmaschine nur einmal die Eingabe, gegebenenfalls gefolgt von (unendlich) vielen Blanks.

Nachdem die Eingabe gelesen wurde, hängt es nur noch von dem aktuellen Zustand der TM ab, ob irgendwann ein Endzustand erreicht werden wird, nicht mehr vom Band.

3. Gibt es für jede Turingmaschine T eine Turingmaschine T' mit nur einem Zustand, die die Sprache von T akzeptiert?

Lösung

Nein! Die Sprache, die eine TM akzeptiert, ist über Endzustände definiert. Sei also T eine TM mit nur einem Zustand q. Dann gibt es zwei Fälle:

$q \notin F$.	Dann ist $L(T) = \emptyset$, da nie ein Endzustand erreicht werden kann.
$q \in F$.	Dann ist $L(T) = \Sigma^*$, da die TM gleich zu Beginn in einem Endzustand ist und es damit eine akzeptierende Konfigurationsfolge gibt.

Also braucht man für jede nicht-triviale Sprache mehrere Zustände.

Bemerkung

Auf ähnliche Weise (aber nicht genauso) kann man zeigen, dass die entsprechende Aussage für Kellerautomaten auch falsch ist, wenn man Akzeptieren mit Endzuständen betrachtet.

3.4 VA 4

Wir konstruieren eine Turingmaschine $T = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$, mit $\Sigma = \{|\}$ wie folgt.

Zu Beginn steht, außer Leerzeichen, nur eine Sequenz von Strichen auf dem Band. Der Schreib-/Lesekopf der Turing-Maschine steht auf dem ersten Strich (von links gesehen).

Die Berechnung erfolgt, indem jeweils der erste Strich (von links gesehen) durch ein Hilfszeichen X ersetzt wird und zusätzlich ein Hilfszeichen X an das linke Ende geschrieben wird.

Zum Schluß werden alle Hilfszeichen von rechts nach links durch Striche ersetzt.

Sei $T = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ mit $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{|\}$, $\Gamma = \{|, X, \square\}$ und $F = \{q_3\}$. Die Übergangsfunktion δ entnehmen wir der folgenden Tabelle:

Übergang		Kommentar
$\delta(q_0, \|)$	$= (q_1, X, L)$	ersetze erstes | durch Hilfszeichen X
$\delta(q_1, X)$	$= (q_1, X, L)$	gehe nach links zum ersten X
$\delta(q_1, \square)$	$= (q_0, X, R)$	füge zusätzliches Hilfszeichen X
	$=$	am Anfang ein
$\delta(q_0, X)$	$= (q_0, X, R)$	gehe nach rechts zum ersten |
$\delta(q_0, \square)$	$= (q_2, \square, L)$	alle | abgearbeitet
$\delta(q_2, X)$	$= (q_2, \|, L)$	ersetze X durch |
$\delta(q_2, \square)$	$= (q_3, \square, R)$	alle X durch | ersetzt, Stopp

Spezifizieren Sie möglichst knapp den Bandinhalt in Abhängigkeit der Eingabe, wenn die Turingmaschine anhält.

Lösung

T verdoppelt die eingegebene Strichzahl.