Regulierte Array-Grammatiken

Henning Fernau
Universität Trier
fernau@informatik.uni-trier.de

Aus Ergebnissen einer Zusammenarbeit mit:

Rudolf Freund

Institut für Computersprachen
Technische Universität Wien
26. Juli 2007

Überblick

- Einführung
- Definitionen
- Ergebnisse
- Zusammenfassung

Einführung zu mehrdimensionalen Grammatiken

Was soll z. B. eine kontextfreie Regel

$$A \longrightarrow B \cdot C$$

bedeuten?

- Modifizierte Konkatenation als neue (partielle) Grundoperation

- "Fenster"- oder "Lupen"-Interpretation

Bezeichnungen

$\mathcal{A} : Z^n \to V \cup \{\#\}$ ist *n-dimensionales Array* über V, falls

$$shape(\mathcal{A}) = \{v \in Z^n \mid \mathcal{A}(v) \neq \#\}$$

endlich ist; $\# \notin V$ ist das *Hintergrundsymbol*.
Schreibweise: $\mathcal{A} = \{(v, \mathcal{A}(v)) \mid v \in shape(\mathcal{A})\}$.

V^{*n} : Menge der n-dimensionalen Arrays über V.
Λ_n: das *leere Array*. $V^{+n} = V^{*n} \setminus \{\Lambda_n\}$.
n-dimensionale Arraysprache $L \subseteq V^{*n}$.

Die *Translation* $\tau_v(\mathcal{A})$ ist definiert durch

$$(\tau_v(\mathcal{A}))(w) = \mathcal{A}(w - v) \text{ für alle } w \in Z^n.$$

Arrays, die durch Translation ineinander überführt werden können, heißen *äquivalent*. Menge der Äquivalenzklassen $[V^{*n}]$.

$\emptyset \neq W \subseteq Z^n$ heißt k-*zusammenhängend*, falls

$$\forall w \in W \exists v \in W |w - v|_{max} \leq k\,.$$

$\mathcal{A} \neq \Lambda_n$ heißt k-*zusammenhängend*, falls $shape(\mathcal{A})$ k-zusammenhängend ist. $\|\mathcal{A}\|$ sei die *Norm* von $\mathcal{A} \neq \Lambda_n$, d. i. das kleinste $k \geq 0$, so daß $\mathcal{A}$ k-zusammenhängend ist.

Arrayregeln

$p = (W, \mathcal{A}_1, \mathcal{A}_2)$ ist eine *n-dimensionale generierende Arrayregel*, wobei $W \subseteq Z^n$ endlich, $\mathcal{A}_1, \mathcal{A}_2 : W \to V \cup \{\#\}$;
p heißt *Λ-frei*, falls $\mathrm{shape}(\mathcal{A}_2) \neq \emptyset$.
$\|p\| = \max\{|v|_{max} \mid v \in W\}$ ist die *Norm* von p.

Ableitungsschritt $\mathcal{C}_1 \vdash_p \mathcal{C}_2$ (mit $\mathcal{C}_i \in V^{*n}$):
Es gibt $v \in Z^n$, so daß für $w \in Z^n \setminus \tau_v(W)$ gilt: $\mathcal{C}_1(w) = \mathcal{C}_2(w)$, während für $w \in \tau_v(W)$ gilt: $\mathcal{C}_i(w) = \mathcal{A}_i(\tau_{-v}(w))$.

O. E.: $\Omega_n := (0, \ldots, 0) \in W$, $\mathcal{A}_1(\Omega_n) \neq \#$.

Speziell: ($V = V_N \cup V_T$)

- *monoton*, falls $\mathrm{shape}\,(\mathcal{A}_1) \subseteq \mathrm{shape}\,(\mathcal{A}_2)$;

- *streng monoton*, falls $\mathrm{shape}\,(\mathcal{A}_2) = W$ und $\|p\| = 1$;

- *#-kontextfrei*, falls $\mathrm{card}\,(\mathrm{shape}\,(\mathcal{A}_1)) = 1$ und $\mathcal{A}_1\,(\Omega_n) \in V_N$;

- *kontextfrei*, falls p monoton und #-kontextfrei ist. Übliche Darstellung:

  $A \to \{(v, \mathcal{A}_2\,(v)) \mid v \in W\}$ statt
  $(W, \{(\Omega_n, A)\} \cup \{(v, \#) \mid v \in W \setminus \{\Omega_n\}\},$
  $\{(v, \mathcal{A}_2\,(v)) \mid v \in W\})$.

- *streng kontextfrei*, falls p streng monoton und kontextfrei ist.

- *regulär*, falls entweder

  1. $W = \{\Omega_n, v\}$, v ist Einheitsvektor, $\mathcal{A}_1 = \{(\Omega_n, B), (v, \#)\}$, $\mathcal{A}_2 = \{(\Omega_n, a), (v, C)\}$ wobei $B, C \in V_N$ und $a \in V_T$, (kurz: $Bv\# \to avC$), *oder*

  2. $W = \{\Omega_n\}$, $\mathcal{A}_1 = \{(\Omega_n, B)\}$, $\mathcal{A}_2 = \{(\Omega_n, a)\}$ mit $B \in V_N$ und $a \in V_T$ (kurz: $B \to a$).

Generierende Array-Grammatik

$G = (n, V_N, V_T, \#, P, \{(v_0, S)\})$,
V_N *Nichtterminalalphabet*,
V_T *Terminalalphabet*, wobei
$V_N \cap V_T = \emptyset$, $\# \notin V = V_N \cup V_T$;
P ist eine endliche Menge n-dimensionaler Arrayregeln über V
$\{(v_0, S)\}$ ist das *Start-Array (Axiom)*.

$\mathcal{B}_2 \in V^{*n}$ ist *direkt ableitbar* aus $\mathcal{B}_1 \in V^{*n}$ vermöge G, i. Z. $\mathcal{B}_1 \vdash_G \mathcal{B}_2$, falls es ein $p \in P$ gibt mit $\mathcal{B}_1 \vdash_p \mathcal{B}_2$. Bezeichne $\vdash_G^*$ die reflexive transitive Hülle von $\vdash_G$.

Die von G erzeugte *(n-dimensionale) Array-Sprache* ist
$L_{gen}(G) = \left\{\mathcal{A} \mid \mathcal{A} \in V_T^{*n},\ \{(v_0, S)\} \vdash_G^* \mathcal{A}\right\}$.

Chomsky-Hierarchie

Für alle $n \geq 1$ gilt:
$L(n,(gen,reg)) \subset L(n,(gen,scf)) \subset$
$L(n,(gen,mon)) \subset L(n,(gen,enum))$.

Ferner gelten die Inklusionen:

$L(n,(gen,scf)) \subset L(n,(gen,smon))$ sowie
$L(n,(gen,cf)) \subset L(n,(gen,mon))$ und
$L(n,(gen,cf)) \subset L(n,(gen,\#-cf)) \subset$
$L(n,(gen,enum))$.

$L(n,(gen,cf))$ und $L(n,(gen,smon))$ sowie
$L(n,(gen,\#-cf))$ und $L(n,(gen,mon))$
sind unvergleichlich.

Bezug zum Wortfall

1-dimensionale Arrays entsprechen Wörtern.
Die vermittelnde Abbildung str löscht alle #-Symbole.

Es gilt:

$$str(L(1,(gen,cf))) = str(L(1,(gen,scf))) =$$
$$str(L(1,(gen,reg))) = L(reg),$$

während aber

$$L(cf) \subset str(L(1,(gen,\# - cf))).$$

Bezug zum Wortfall

$L(cf) \subset str(L(1,(gen, \# - cf)))$.

Betrachte Grammatik G mit

$N = \{S, A, B, C, C'\}, T = \{a, b, c\},$

$P = \{S\#\#\#\# \rightarrow AB\#bc, \#\#\#B \rightarrow ABCb, B \rightarrow \#, C\,(2)\,\# \rightarrow \#C'\}$
$\cup \{\#A \rightarrow A\#, C(1)\# \rightarrow \#C, C'\# \rightarrow \#C', A \rightarrow a, C' \rightarrow c\}.$

sowie Start mit $(0, S)$.
Es ist $str(L(G)) = \{a^n b^n c^n \mid n \geq 1\}$.

Idee: Schaffe passende Lücken; in den Regeln mit (3) angedeutet.Beispiel:
$S \Rightarrow AB\#bc \Rightarrow^3 A\#\#\#B\#bc \Rightarrow AABCb\#bc \Rightarrow AAB\#bCbc \Rightarrow AAB\#b\#bcC' \Rightarrow^7 AA\#\#\#B\#b\#bc\#C' \Rightarrow AAABCb\#b\#bc\#C' \Rightarrow AAAB\#bCb\#bc\#C' \Rightarrow AAAB\#b\#b\#bcC'C' \Rightarrow^* aaa\#\#b\#b\#bccc$

Geordnete Grammatiken

$G = (V_N, V_T, P, S, <)$; es sei $V_G = V_N \cup V_T$.

$<$ sei Halbordnung auf P.

$\alpha \to \beta$ ist anwendbar, wenn keine größere Regel anwendbar ist.

Bezeichnung: O.

Programmierte Grammatiken

$G = (V_N, V_T, P, L_i, L_f, S)$; es sei $V_G = V_N \cup V_T$.
$L_i, L_f \subseteq \Lambda(G)$ Markenmengen (Start/Ziel-Mengen)
P Produktionen der Form $(r:\ \alpha \to \beta,\ \sigma(r),\ \phi(r))$
$r:\ \alpha \to \beta$: mit r markierte Kernregel $\alpha \to \beta$
$\sigma(r), \phi(r) \subseteq \Lambda(G)$ Markenmengen (Erfolgs- und Misserfolgsfelder).
Für (x, r_1) und (y, r_2) aus $V_G^* \times \mathrm{Lab}(P)$:
$(x, r_1) \vdash (y, r_2)$ gdw. entweder
$x = z_1\alpha z_2$, $y = z_1\beta z_2$, falls
$(r_1:\ \alpha \to \beta,\ \sigma(r_1),\ \phi(r_1)) \in P$ und $r_2 \in \sigma(r_1)$
oder $x = y$, falls $(r_1:\ \alpha \to \beta,\ \sigma(r_1),\ \phi(r_1)) \in P$ nicht anwendbar auf x und $r_2 \in \phi(r_1)$.
Bezeichnungen: P_{ac}, P, P_{ut}.

Ergebnisse im Wortfall

- $\mathcal{L}(X) = \mathcal{Y}(X)$ for $X \in \{\mathrm{reg}, \mathrm{mon}, \mathrm{enum}\}$ and $Y \in \{O, P_{ac}, P_{ut}, P\}$;
- $\mathcal{L}(\mathrm{cf} - \lambda) \subset \mathcal{P}(\mathrm{cf} - \lambda) \subset \mathcal{P}_{ac}(\mathrm{cf} - \lambda) \subset \mathcal{L}(\mathrm{mon})$;
- $\mathcal{L}(\mathrm{cf} - \lambda) \subset \mathcal{P}_{ut}(\mathrm{cf} - \lambda) \subseteq \mathcal{P}_{ac}(\mathrm{cf} - \lambda) \subset \mathcal{L}(\mathrm{mon})$;
- $\mathcal{P}(\mathrm{cf} - \lambda) \subseteq \mathcal{P}(\mathrm{cf}) \subset \mathcal{P}_{ac}(\mathrm{cf}) = \mathcal{L}(\mathrm{enum})$;
- $\mathcal{O}(\mathrm{cf} - \lambda) \subset \mathcal{P}_{ut}(\mathrm{cf} - \lambda) \subset \mathcal{P}_{ut}(\mathrm{cf}) \subseteq \mathcal{P}_{ac}(\mathrm{cf}) = \mathcal{L}(\mathrm{enum})$;
- $\mathcal{L}(\mathrm{cf} - \lambda) = \mathcal{L}(\mathrm{cf}) \subset \mathcal{O}(\mathrm{cf} - \lambda) \subseteq \mathcal{O}(\mathrm{cf}) \subset \mathcal{P}_{ut}(\mathrm{cf}) \subseteq \mathcal{L}(\mathrm{enum})$.
- $\mathrm{reg}, \mathrm{mon}, \mathrm{enum}$ “uninteressant”.

Rechteckränder Dies geht nicht mit einer regulären Array-Grammatik (Yamamoto et al. 89).
Lemma: $R_H \in L(2, (gen, reg), O)$.

$$
\begin{aligned}
G &= (2, \{S, A, B, C, D, E, F, Q\}, \{a\}, \#, \\
&\quad (P, <), \{((0,1), S)\}), \\
P &= \left\{ \begin{matrix} \# \\ S \end{matrix} \rightarrow \begin{matrix} A \\ a \end{matrix}, \begin{matrix} \# \\ A \end{matrix} \rightarrow \begin{matrix} A \\ a \end{matrix}, A\# \rightarrow aB, \right. \\
&\quad B\# \rightarrow aB,\ B\# \rightarrow aC, \begin{matrix} C \\ \# \end{matrix} \rightarrow \begin{matrix} a \\ D \end{matrix}, \\
&\quad \begin{matrix} D \\ \# \end{matrix} \rightarrow \begin{matrix} a \\ D \end{matrix}, \begin{matrix} D \\ \# \end{matrix} \rightarrow \begin{matrix} a \\ E \end{matrix}, \#E \rightarrow Ea, \\
&\quad \left. \#E \rightarrow Fa, \begin{matrix} \# \\ F \end{matrix} \rightarrow \begin{matrix} Q \\ a \end{matrix}, F \rightarrow a \right\}.
\end{aligned}
$$

Die Ordnung ist gegeben durch $F \rightarrow a < \begin{matrix} \# \\ F \end{matrix} \rightarrow \begin{matrix} Q \\ a \end{matrix}$, so daß $F \rightarrow a$ nur anwendbar ist, wenn kein Hintergrundsymbol über F steht, da sonst $\begin{matrix} \# \\ F \end{matrix} \rightarrow \begin{matrix} Q \\ a \end{matrix}$ das Fehlersymbol Q einführt.

Resultate im Array-Fall

Satz: Es seien $n \geq 1$ und $X \in \{enum, \#{-}cf, mon, smon, cf, scf, reg\}$ beliebig. Dann gilt $L(n, (gen, X), O) \subseteq L(n, (gen, X), P_{ut})$.

Beweis: Für jede (o.E. markierte) Regel $r : (W, \mathcal{A}_1, \mathcal{A}_2)$ der geordneten Grammatik ($r \in \Lambda_O$), wobei $F(r) = \{(r, i) \mid 1 \leq i \leq k(r)\}$ die Marken der Regeln $p_{r,i}$ sei, die größer als r sind, nehmen wir auf:

$$\left((r, i) : p^{\ominus}_{r,i}, \{(r, i+1)\}, \{(r, i+1)\}\right)$$

für $1 \leq i \leq k(r)$ und

$$((r, k(r)+1) : (W, \mathcal{A}_1, \mathcal{A}_2), \Lambda_O \times \{1\}, \Lambda_O \times \{1\}).$$

Hierbei sei $p^{\ominus} = (W, \mathcal{A}_1, F_W)$ mit $F_W = \{(v, F) \mid v \in W\}$.
Im regulären Fall Obacht!

Satz: (s. auch Freund 94) Es seien $n \geq 1$ und $X \in \{enum, \# - cf\}$, $Y \in \{O, P, P_{ac}, P_{ut}\}$. So ist: $L(n, (gen, X), Y) = L(n, (gen, enum))$.

Satz (Freund & Păun 93)
Es sei $Y \in \{O, P, P_{ac}, P_{ut}\}$. So ist:
$L(lin) \subset str(L(1, (gen, cf), Y)) \subset Y(cf)$.

Folgerung Es seien $n \geq 1$ und $Y \in \{O, P, P_{ac}, P_{ut}\}$. $L(n, (gen, cf)) \subset L(n, (gen, cf), Y) \subset L(n, (gen, mon))$.
Bei "streng" vieles noch offen!

Akzeptierende (Array-) Grammatiken

Frage: Wie ist ein akzeptierender Ableitungsschritt zu definieren?

Ansatz: Nimm Definition für den generierenden Fall und lies sie reduzierend.

Rechteckränder

$$
\begin{aligned}
G &= (2, \{S, A, B, C, D, E, F, Q\}, \{a\}, \#, \\
&\quad (P, <), \{((0,1), S)\}), \quad \text{kontextfrei} \\
P &= \left\{ \begin{matrix} A \\ a \end{matrix} \to \begin{matrix} \# \\ S \end{matrix}, \begin{matrix} A \\ a \end{matrix} \to \begin{matrix} \# \\ A \end{matrix}, aB \to A\#, \right. \\
&\quad aB \to B\#,\ aC \to B\#, \begin{matrix} a \\ D \end{matrix} \to \begin{matrix} C \\ \# \end{matrix}, \\
&\quad \begin{matrix} a \\ D \end{matrix} \to \begin{matrix} D \\ \# \end{matrix}, \begin{matrix} a \\ E \end{matrix} \to \begin{matrix} D \\ \# \end{matrix}, Ea \to \#E, \\
&\quad \left. Fa \to \#E, \begin{matrix} \# \\ F \end{matrix} \to \begin{matrix} \# \\ Q \end{matrix}, a \to F \right\}.
\end{aligned}
$$

$Fa \to \#E < \begin{matrix} \# \\ F \end{matrix} \to \begin{matrix} \# \\ Q \end{matrix}$ definiert die Ordnung. $a \to F$ kann als erstes nur einmal angewandt werden. Sodann ist $Fa \to \#E$, überwacht durch $\begin{matrix} \# \\ F \end{matrix} \to \begin{matrix} \# \\ Q \end{matrix}$, anzuwenden.

Obacht: Fehlerregel im Vergleich zum gen. Fall (nicht streng monoton)!

Dualitätssatz: Es seien $n \geq 1$ und $X \in \{\mathrm{enum}, \# - \mathrm{cf}, \mathrm{mon}, \mathrm{smon}, \mathrm{cf}, \mathrm{scf}, \mathrm{reg}\}$, $Y \in \{P, \lambda\}$. Dann gilt:
$L(n, (\mathrm{gen}, X), Y) = L(n, (\mathrm{acc}, X), Y)$.

Satz: Es seien $n \geq 1$ und $X \in \{\mathrm{enum}, \#-\mathrm{cf}, \mathrm{mon}, \mathrm{smon}, \mathrm{cf}, \mathrm{scf}, \mathrm{reg}\}$ beliebig. Dann gilt $L(n, (\mathrm{acc}, X), O) \subseteq L(n, (\mathrm{acc}, X), P_{ut})$.

Beweis wie im generierenden Fall; setze

$p^{\ominus} = \left(W, \mathcal{A}_1, \mathcal{A}_2^F\right)$, wobei
$\mathcal{A}_2^F = \{(\Omega_n, F)\} \cup \{(v, \mathcal{A}_2(v)) \mid v \in (W \setminus \{\Omega_n\})\}$.

Satz: Es seien $n \geq 1$, $X \in \{enum, \# - cf, mon, cf\}$. Dann ist:
$L(n, (gen, X), P_{ac}) \subseteq L(n, (acc, X), P_{ac})$.

Beweis. Es sei $G = (n, V_N, V_T, \#, (R, L_i, L_f), \{(v_0, S)\})$zu simulieren. Definiere

$$\begin{aligned}
G^d &= \left(n, V_N \cup \{F\}, V_T, \#, \left(R^d, L_i^d, L_f^d\right), \{(v_0, S)\}\right), \\
R^d &= R' \cup R'', \quad L_f^d = \{f', f''\}, \\
L_i^d &= \{s', s'' \mid r \in \sigma(s) \text{ für ein } r \in L_f\}.
\end{aligned}$$

Für $(r : (W, \mathcal{A}_1, \mathcal{A}_2), \sigma(r), \varphi(r))$ in R nimm

$$\begin{aligned}
&\left(r' : (W, \mathcal{A}_2, \mathcal{A}_1), \left(\sigma^{-1}(r)\right)' \cup \left(\varphi^{-1}(r)\right)'', \emptyset\right) \text{ in } R', \\
&\left(r'' : (W, \mathcal{A}_1, \mathcal{A}_1), \emptyset, \left(\sigma^{-1}(r)\right)' \cup \left(\varphi^{-1}(r)\right)''\right) \text{ in } R'';
\end{aligned}$$

Darüberhinaus: nimm $(f' : F \to F, \emptyset, \emptyset)$ in R' und $(f'' : F \to F, \emptyset, \emptyset)$ in R''. Für $x \in \{\sigma, \varphi\}$ sind die Mengen $x^{-1}(r)$ gegeben durch

$$\begin{aligned}
x^{-1}(r) &= \{s \mid r \in x(s)\} \text{ für } r \notin L_i \text{ sowohl} \\
x^{-1}(r) &= \{s \mid r \in x(s)\} \cup \{f\} \text{ für } r \in L_i.
\end{aligned}$$

Satz: Für jedes $n \geq 2$ und $Y \in \{P_{ut}, P_{ac}, O\}$ sind $L(n, (gen, reg), Y)$ und $L(n, (acc, reg), Y)$ unvergleichbar.

Idee: (a) Linien der Dicke 1 mit Startpunkt im Ursprung und einem "freien Ende" liegen in $L(n, (acc, reg), O)$, aber nicht in $L(n, (gen, reg), P_{ac})$.

(b) $R_H \in L(n, (gen, reg), O)$ liegt nicht in $L(n, (acc, reg), P_{ac})$.

Bemerkung: Dagegen gilt für $Y \in \{P, P_{ut}, P_{ac}, O\}$:
$L(1, reg) = L(1, (acc, reg), Y) = L(1, (gen, reg), Y)$.

Satz: Es seien $n \geq 1$ und $Y \in \{P_{ut}, P_{ac}, O\}$.

- $L\left(n, \left(gen, cf\right), Y\right) \subseteq$
  $L\left(n, \left(acc, cf\right), Y\right) =$
  $L\left(n, \left(acc, mon\right), Y\right) =$
  $L\left(n, \left(gen, mon\right), Y\right) =$
  $L(n, mon)$.

- $L\left(n, \left(acc, \# - cf\right), Y\right) =$
  $L\left(n, \left(gen, \# - cf\right), Y\right) =$
  $L\left(n, \left(acc, enum\right), Y\right) =$
  $L(n, enum)$.

Beweis teilweise mühsam.