Regulated Array Grammars of Finite Index

C. Gruber, J. Reiter

TU Wien

3.März, 2010

Table of contents

n-dimensional array

Ein n-dimensionales Array A über ein Alphabet V (Menge aller non-terminal und terminal Symbole) ist eine Funktion

$$A : Z^n \to V \cup \{\#\} \quad n \in N = \{1, 2, ...\}$$

wobei

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

endlich ist und $\# \notin V$ als *background* oder *blank Symbol* bezeichnet wird. Das Array A kann nun so definiert werden

$$A = \{(v, A(v)) \mid v \in shape(A)\}.$$

n-dimensional array production and grammar

Eine n-dimensionale Array Produktion p über dem Alphabet V ist ein Tripel (W, A_1, A_2) wobei $W \subseteq Z^n$ eine endliche Menge von Koordinaten ist und A_1 und A_2 Abbildungen von W auf $V \cup \{\#\}$ sind.

Eine n-dimensionale Array Grammatik kann nun als Sechstupel

$$G = (n, V_N, V_T, \#, P, \{(v_0, S)\})$$

definiert werden. $\{(v_0, S)\}$ wird als Startarray (Axiom), v_0 als Startvektor und S als das Startsymbol bezeichnet.

matrix grammar

Eine Matrixgrammatik G_M ist ein 4-Tupel

$$G_M = (V_N, V_T, (M, F), S),$$

F kann auch als Fehlermenge bezeichnet werden. Ist $F = \emptyset$, dann kann G_M als Matrixgrammatik ohne appearence checking bezeichnet werden.

graph controlled grammar

Eine graph-controlled Grammatik G_P ist ein 4-Tupel

$$G_M = (V_N, V_T, (R, L_{in}, L_{fin}), S),$$

R ist eine endliche Menge von Regeln r der Form

$$(l(r) : p(l(r)), \sigma(l(r)), \varphi(l(r))).$$

Falls alle Felder $\varphi(l(r))$ leer sind fr alle $r \in R$, dann kann G_P als graph-controlled Grammatik ohne appearence checking bezeichnet werden.

Matrix- und graph-controlled Grammatiken knnen in Arraygrammatiken direkt bergefhrt werden, indem ihre Produktionen durch Arrayproduktionen ersetzt werden.

graph controlled grammar

Eine graph-controlled Grammatik G_P ist ein 4-Tupel

$$G_M = (V_N, V_T, (R, L_{in}, L_{fin}), S),$$

R ist eine endliche Menge von Regeln r der Form

$$(l(r) : p(l(r)), \sigma(l(r)), \varphi(l(r))).$$

Falls alle Felder $\varphi(l(r))$ leer sind fr alle $r \in R$, dann kann G_P als graph-controlled Grammatik ohne appearence checking bezeichnet werden.

Matrix- und graph-controlled Grammatiken knnen in Arraygrammatiken direkt bergefhrt werden, indem ihre Produktionen durch Arrayproduktionen ersetzt werden.

bounded derivations

Index einer Ableitung

Der Index einer Ableitung D eines Terminalobjekts w in einer Grammatik G ist mit der maximalen Anzahl von non-terminal Symbolen, die in einem Zwischenableitungsschritt vorkommen, definiert und wird mit $ind_{G,D}(w)$ bezeichnet.

Weiters bezeichnet $ind_{G,min}(w)$ bzw. $ind_{G,max}(w)$ das Minimum bzw. das Maximum aus der Menge

$$\{ind_{G,D}(w) \mid w \text{ is generated by } G\}$$

Entsprechend gibt es nun die Definition fr die Grammatik

$$ind_Y(G) = sup\{ind_{G,Y}(w) \mid w \text{ is generated by } G\} \quad Y \in \{min, max\}$$

Bei einer endlichen Index Restriktion nur Objekte w, die von einer Grammatik G mit einer Ableitung $ind_{G,Y}(w) \leq k$ fr $Y \in \{min, max\}$.

bounded derivations

Index einer Ableitung

Der Index einer Ableitung D eines Terminalobjekts w in einer Grammatik G ist mit der maximalen Anzahl von non-terminal Symbolen, die in einem Zwischenableitungsschritt vorkommen, definiert und wird mit $ind_{G,D}(w)$ bezeichnet.

Weiters bezeichnet $ind_{G,min}(w)$ bzw. $ind_{G,max}(w)$ das Minimum bzw. das Maximum aus der Menge

$$\{ind_{G,D}(w) \mid w \text{ is generated by } G\}$$

Entsprechend gibt es nun die Definition fr die Grammatik

$$ind_Y(G) = sup\{ind_{G,Y}(w) \mid w \text{ is generated by } G\} \quad Y \in \{min, max\}$$

Bei einer endlichen Index Restriktion nur Objekte w, die von einer Grammatik G mit einer Ableitung $ind_{G,Y}(w) \leq k$ fr $Y \in \{min, max\}$.

grammar with prescribed teams

Eine Grammatik mit prescribed teams G_t ist ein 4-Tupel

$$G_t = (V_N, V_T, (P, R, F), S),$$

$G = (V_N, V_T, P, S)$ ist eine kontextfreie Grammatik, R ist eine endliche Menge von Teams aus P und F ist die Menge von Produktionen, die bei dem appearence checking bersprungen werden knnen. Ist $F = \emptyset$, dann kann G_t als Grammatik mit prescribed teams ohne appearence checking bezeichnet werden.

Beispiel fr eine Grammatik mit prescribed teams (1)

2-dimensionale Array-Grammatik mit prescibed teams:

$$G = (n, \{D, E, L, Q, R, S, U\}, \{a\}, \#, (P, R, F), \{((0,0), S)\})$$

$$P = \left\{ \begin{array}{c} \# \\ S\# \end{array} \rightarrow \begin{array}{c} L \\ aD \end{array}, \begin{array}{c} \# \\ L \end{array} \rightarrow \begin{array}{c} L \\ a \end{array}, D\# \rightarrow aD, \begin{array}{c} \#\# \\ L \end{array} \rightarrow \begin{array}{c} aU \\ a \end{array}, \right.$$

$$D\# \rightarrow aR, \begin{array}{c} \# \\ R \end{array} \rightarrow \begin{array}{c} R \\ a \end{array}, U\# \rightarrow aU,$$

$$\left. U\# \rightarrow aU, U \rightarrow E, \begin{array}{c} \# \\ R \end{array} \rightarrow \begin{array}{c} Q \\ a \end{array}, R \rightarrow a, E \rightarrow a \right\}$$

Beispiel fr eine Grammatik mit prescribed teams (2)

$$R = \left\{ \left\langle \begin{array}{c} \# \\ S\# \end{array} \rightarrow \begin{array}{c} L \\ aD \end{array} \right\rangle, \left\langle \begin{array}{c} \# \\ L \end{array} \rightarrow \begin{array}{c} L \\ a \end{array},\ D\# \rightarrow aD \right\rangle, \right.$$

$$\left\langle \begin{array}{c} \#\# \\ L \end{array} \rightarrow \begin{array}{c} aU \\ a \end{array},\ D\# \rightarrow aR \right\rangle, \left\langle \begin{array}{c} \# \\ R \end{array} \rightarrow \begin{array}{c} R \\ a \end{array},\ U\# \rightarrow aU \right\rangle,$$

$$\left. \left\langle U\# \rightarrow aU, U \rightarrow E, \begin{array}{c} \# \\ R \end{array} \rightarrow \begin{array}{c} Q \\ a \end{array} \right\rangle, \langle R \rightarrow a, E \rightarrow a \rangle \right\}$$

$$F = \left\{ \begin{array}{c} \# \\ R \end{array} \rightarrow \begin{array}{c} Q \\ a \end{array} \right\}$$

Ableitungssequenz:

$$S \Rightarrow_G \begin{array}{ll} L & \\ a & D \end{array} \Rightarrow_G \begin{array}{lll} a & U & \\ a & & \\ a & a & R \end{array} \Rightarrow_G \begin{array}{lll} a & a & U \\ a & & \\ a & a & R \end{array} \Rightarrow_G \begin{array}{lll} a & a & E \\ a & & R \\ a & a & a \end{array} \Rightarrow_G \begin{array}{lll} a & a & a \\ a & & a \\ a & a & a \end{array}$$

Gefolgerte Resultate

Theorem 1

Fr jedes $k \in \{fin\} \cup \{j \mid j \geq 1\}$, gilt

$$PT^{\lceil k \rceil}(cf) = PT_{ac}^{\lceil k \rceil}(cf) = Z^{Y,\lceil k \rceil}(cf) = Z^{min,\cap\lceil k \rceil}(cf)$$

fr alle $Z \in \{M, M_{ac}, P, P_{ut}, P_{ac}\}$ und $Y \in \{min, max\}$.

Theorem 2

Fr $X \in \{n - \# - cf, n - cf \mid n \geq 1\}$, gilt

1. $PT^{\lceil fin \rceil}(X) = Z^{Y,\lceil fin \rceil}(X) = Z^{min,\cap\lceil fin \rceil}(X)$ und auch
2. $PT_{ac}^{\lceil fin \rceil}(X) = Z_{ac}^{Y,\lceil fin \rceil}(X) = Z_{ac}^{min,\cap\lceil fin \rceil}(X)$

fr alle $Z \in \{M, P\}$ und $Y \in \{min, max\}$.

Gefolgerte Resultate

Theorem 1

Fr jedes $k \in \{fin\} \cup \{j \mid j \geq 1\}$, gilt

$$PT^{[k]}(cf) = PT_{ac}^{[k]}(cf) = Z^{Y,[k]}(cf) = Z^{min,\cap[k]}(cf)$$

fr alle $Z \in \{M, M_{ac}, P, P_{ut}, P_{ac}\}$ und $Y \in \{min, max\}$.

Theorem 2

Fr $X \in \{n - \# - cf, n - cf \mid n \geq 1\}$, gilt

1. $PT^{[fin]}(X) = Z^{Y,[fin]}(X) = Z^{min,\cap[fin]}(X)$ und auch
2. $PT_{ac}^{[fin]}(X) = Z_{ac}^{Y,[fin]}(X) = Z_{ac}^{min,\cap[fin]}(X)$

fr alle $Z \in \{M, P\}$ und $Y \in \{min, max\}$.