PARSING MILDLY CONTEXT-SENSITIVE RMS

Tilman Becker **Dominik Heckmann**
German Research Center for Artificial Intelligence (DFKI GmbH)
{becker,heckmann}@dfki.de

Abstract
We introduce Recursive Matrix Systems (RMS) which encompass mildly context-sensitive formalisms and present efficient parsing algorithms for linear and context-free variants of RMS. The time-complexities are $O(n^{2h+1})$, and $O(n^{3h})$ respectively, where h is the height of the matrix. It is possible to represent Tree Adjoining Grammars (TAG [1], MC-TAG [2], and R-TAG [3]) as RMS uniformly.

1 Recursive Matrix Systems (RMS)

$RMS = (G, I)$ is a two-step formalism. In a first step, a grammar $G = (N, T, S, P)$ generates a set of recursive matrices. In a second step, a yield function maps, according to the interpretation I, the recursive matrices to a set of strings $L(G, I)$. The recursive matrix generating grammars are ordinary phrase structure grammars, where the terminal symbols are replaced by vertical vectors. These vertical vectors are filled with terminal symbols, nonterminal symbols, or they are left empty. The two-step formalism: $S \in N \overset{derive}{\Longrightarrow_G^*} m \in \mathcal{M} \overset{yield}{\longmapsto_I} w \in T^*$.

Example: $G_1 = (N, T, S, P) = (\{S\}, \{a, b, c\}, S, \{S \to \begin{bmatrix} a \\ b \\ c \end{bmatrix} S\ ,\ S \to \begin{bmatrix} a \\ b \\ c \end{bmatrix}\})$. Every time the first rule is applied, a new column with terminals is added to the matrix. The last rule terminates the process. A possible derivation: $S \Longrightarrow_{G_1}^1 \begin{bmatrix} a \\ b \\ c \end{bmatrix} S \Longrightarrow_{G_1}^1 \begin{bmatrix} a & a \\ b & b \\ c & c \end{bmatrix} S \Longrightarrow_{G_1}^1 \begin{bmatrix} a & a & a \\ b & b & b \\ c & c & c \end{bmatrix}$.

The product of the derivation process is a matrix with terminals as elements. Finally, these terminal symbols are combined into one string. There are many possible interpretation functions. However, it seems reasonable to read the terminal symbols within the matrix row by row from top to bottom, and each row alternating from left to right and from right to left. This interpretation condition for height 3 could be visualized by $\begin{smallmatrix} \to \\ \leftarrow \\ \to \end{smallmatrix}$. The grammar G_1 together with this interpretation generates strings of the form *aaa...bbb...ccc...*. Thus, the generated language is $L(G_1, \begin{smallmatrix} \to \\ \leftarrow \\ \to \end{smallmatrix}) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$.

Definition: h is a positive integer. A *recursive matrix grammar* is a four-tuple $G = (N, T, S, P)$, where N and T are disjoint alphabets, $S \in N$, P is a finite set of ordered pairs $(u, v) \in N \times (N \cup \mathcal{C}_h)^*$, where G is *context-free*, if $P \subset N \times (N \cup \mathcal{C}_h)^*$, *linear*, if $P \subset N \times \mathcal{C}_h N^\epsilon \mathcal{C}_h^\epsilon$, *regular*, if $P \subset N \times \mathcal{C}_h N^\epsilon$.

$\mathcal{C}_h = \{ \begin{bmatrix} v_1 \\ \vdots \\ v_h \end{bmatrix} \mid \forall_{1 \leq x \leq h} : v_x \in V^\epsilon \}$. $\mathcal{C}_h$ is the set of *columns* of height h over $V^\epsilon = N \cup T \cup \{\epsilon\}$.

An *interpretation* $I \in \mathcal{I}_h$ is a vertical vector with left and right arrows as elements, where

$\mathcal{I}_h = \{ \begin{matrix} d_1 \\ \vdots \\ d_h \end{matrix} \mid \forall_{1 \leq x \leq h} : d_x \in \{\to, \leftarrow\} \}$. $\mathcal{I}_h$ is the *set of Interpretations* of height h.

2 Parsing Schemata for Recursive Matrix Systems (RMS)

The notation for the parsing schemata is close to [4]. An Earley parser for RMS can be found in [2].

RMS offer brief vector notations: Let $\vec{i} = \begin{matrix} i_1 \\ \vdots \\ i_h \end{matrix}$, $\vec{i'} = \begin{matrix} i_2 \\ \vdots \\ i_{h+1} \end{matrix}$, $\vec{j} = \begin{matrix} j_1 \\ \vdots \\ j_h \end{matrix}$, $\vec{k} = \begin{matrix} k_1 \\ \vdots \\ k_h \end{matrix}$, $\vec{v} = \boxed{\begin{matrix} v_1 \\ \vdots \\ v_h \end{matrix}}$.

The set of hypothesis: $\mathcal{H} = \{ [\, a \mid i-1 \mid i \,] \mid a = a_i \,,\ 1 \leq i \leq n \} \cup \{ [\, \epsilon \mid i \mid i \,] \mid 0 \leq i \leq n \}$.

The set of items $\mathcal{J}$ depends on the height h: $\mathcal{J}(h) = \{ [\, A \mid i \mid j \,] , [\, A \mid \vec{i} \mid \vec{j} \,] \}$.

The parsing schema $\mathbb{P}(CYK)(h\mathcal{RMS}(CFG, \rightrightarrows) \equiv MC-TAG)$ is defined by the deduction rules:

$$D^{init} = \{ [\, v_1 \mid i_1 \mid k_1 \,], \ldots, [\, v_h \mid i_h \mid k_h \,] \vdash [\, A \mid \vec{i} \mid \vec{k} \,] \mid (A \rightarrow \vec{v}) \in P \}$$

$$D^{combine} = \{ [\, B \mid \vec{i} \mid \vec{j} \,], [\, C \mid \vec{j} \mid \vec{k} \,] \vdash [\, A \mid \vec{i} \mid \vec{k} \,] \mid (A \rightarrow BC) \in P \}$$

$$D^{linearize} = \{ [\, A \mid \vec{i} \mid \vec{i'} \,] \vdash [\, A \mid i_1 \mid i_{h+1} \,] \}$$

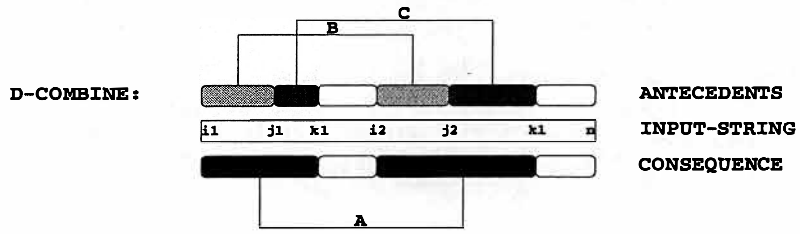
The parsing schema $\mathbb{P}(CYK)(2\mathcal{RMS}(LIN, \rightrightarrows) \equiv R-TAG)$ is defined by the deduction rules:

$$D^{init} = \left\{ \begin{bmatrix} v_1 & i_1 & j_2 \\ v_3 & i_3 & j_4 \end{bmatrix} \vdash \begin{bmatrix} A & \begin{matrix} i_1 & j_2 \\ i_3 & j_4 \end{matrix} \end{bmatrix} \mid (A \rightarrow \boxed{\begin{matrix} v_1 \\ v_3 \end{matrix}}) \in P \right\}$$

$$D^{adjoin} = \left\{ \begin{bmatrix} v_1 & i_1 & j_1 \\ v_3 & i_3 & j_3 \end{bmatrix} \begin{bmatrix} B & \begin{matrix} j_1 & i_2 \\ j_3 & i_4 \end{matrix} \end{bmatrix} \begin{bmatrix} v_2 & i_2 & j_2 \\ v_4 & i_4 & j_4 \end{bmatrix} \vdash \begin{bmatrix} A & \begin{matrix} i_1 & j_2 \\ i_3 & j_4 \end{matrix} \end{bmatrix} \mid (A \rightarrow \boxed{\begin{matrix} v_1 \\ v_3 \end{matrix}} B \boxed{\begin{matrix} v_2 \\ v_4 \end{matrix}}) \in P \right\}$$

$$D^{linearize} = \left\{ \begin{bmatrix} A & \begin{matrix} i_1 & i_2 \\ i_2 & i_3 \end{matrix} \end{bmatrix} \vdash [\, A \mid i_1 \mid i_3 \,] \right\}$$

The time–complexity of the algorithms can be derived from the number of relevant indices, where each vector $\vec{i}$ represents h indices. Step $D^{combine}$ in algorithm $\mathbb{P}(CYK)(h\mathcal{RMS}(CFG, \rightrightarrows))$ has three relevant vectors $\vec{i}, \vec{j}, \vec{k}$, which each represent h indices, resulting in a time–complexity of $O(n^{3h})$. Step $D^{linearize}$ has a $O(n^{h+1})$ time-complexity. The correctness is inherited from the CFG cases. The following figure explains the variables of the deduction rule $D^{Combine}$ for $\mathbb{P}(CYK)(2\mathcal{RMS}(CFG, \rightrightarrows)$:



References

[1] Tilman Becker and Dominik Heckmann. 1998. Recursive Matrix Systems (RMS) and TAG. *Proceedings of the Fourth International Workshop on TAG (TAG+4),Philadelphia, PA.*

[2] Dominik Heckmann. 1999. Recursive Matrix Systems. *Diploma Thesis, University of Saarland.*

[3] Giorgio Satta and William Schuler. 1998. Restrictions on Tree Adjoining Languages. *Proceedings of COLING-ACL'98*, pages 1176–1182.

[4] Klaas Sikkel and Anton Nijhold. 1997. Parsing of Context–Free Languages. *Handbook of Formal Languages, Vol. 2, Springer, Berlin.*