A Left Corner Parser for Tree Adjoining Grammars

Víctor J. Díaz†, Vicente Carrillo†, and Miguel A. Alonso‡
†*Universidad de Sevilla and* ‡*Universidade da Coruña*

1. Introduction

Tabular parsers can be de£ned as deduction systems where formulas, called items, are sets of complete or incomplete constituents (Sikkel, 1997; Shieber, Schabes and Pereira, 1995). Formally, given an input string $w = a_1 \dots a_n$ with $n \geq 0$ and a grammar G, a parser $\mathbb{P}$ is a tuple $(\mathcal{I}, \mathcal{H}, \mathcal{D})$ where $\mathcal{I}$ is a set of items, $\mathcal{H}$ is a set of hypothesis ($[a_i, i-1, i]$ with $1 \leq i \leq n$) that encodes the input string, and $\mathcal{D}$ is a set of deduction steps that determines how items are combined in order to deduce new items. The deductive approach allows us to establish relations between two parsers in a formal way. One of the most interesting relations between parsers are *£lters* because they can be used to improve the performance of tabular parsers in practical cases. The application of a £lter to a parser yields a new parser which performs less deductions or contracts sequences of deductions to single deduction steps.

One well-known example of a £lter is the relation between Earley and Left Corner (LC) parsers for Context-Free Grammars (CFGs). A LC parser reduces the number of items deduced by Earley's parser using the left corner relation. Given a CFG, the left corner of a non-terminal symbol A is the terminal or non-terminal symbol X if and only if there exists a production $A \rightarrow X\nu$ in the grammar, where ν is a sequence of symbols. In the case of $A \rightarrow \varepsilon$, we consider ε as the left corner of A. The notion of the left corner relation allow us to rule out the prediction performed on X by an Earley's parser.

Most tabular parsers for Tree Adjoining Grammars (TAGs) are extensions of well-known tabular parser for CFGs. For example, we can cite a number of tabular parsers for TAGs de£ned on the basis of the Earley's algorithm (Alonso Pardo *et al.*, 1999; Lang, 1990; Joshi and Schabes, 1997; Nederhof, 1999). Although several approaches have been described to improve the performance of TAGs parsers, most of them based on restrictions in the formalism (Schabes and Waters, 1995) or compilation into £nite-state automata (Evans and Weir, 1998), to the best of our knowledge, no attempt has been made to improve the practical performance of Earley-based parsers for TAGs by introducing the left-corner relation.

2. Notation

Let $\mathcal{G} = (V_N, V_T, S, \boldsymbol{I}, \boldsymbol{A})$ be a TAG, where V_N and V_T are the alphabets of non-terminal and terminal symbols, respectively, $S \in V_N$ is the axiom, and $\boldsymbol{I}$ and $\boldsymbol{A}$ are the set of initial and auxiliary trees, respectively. We refer to the root of an elementary tree γ as $\mathbf{R}^\gamma$ and to the foot of an auxiliary tree β as $\mathbf{F}^\beta$. The set $\mathrm{adj}(M^\gamma)$ includes every auxiliary tree that may be adjoined at node M^γ. We use a dummy symbol $\mathbf{nil} \notin \boldsymbol{A}$ for denoting adjoining constraints. If adjunction is not mandatory at M^γ, then $\mathbf{nil} \in \mathrm{adj}(M^\gamma)$. If adjunction is forbidden at M^γ, then $\mathrm{adj}(M^\gamma) = \{\mathbf{nil}\}$. We say M^γ is an adjunction node if there exists an auxiliary tree β which can be adjoined at that node.

Although TAGs are tree-rewriting systems, we can translate every elementary tree γ into a set of productions $\mathcal{P}(\gamma)$. This notation will be useful when de£ning the set of items for TAGs parsers since dotted productions can be introduced for representing partial parse trees. We de£ne a production $N^\gamma \rightarrow N_1^\gamma \dots N_g^\gamma$ for every node N^γ and its ordered g children $N_1^\gamma \dots N_g^\gamma$ in an elementary tree. We refer to the set of productions related to an elementary tree γ as $\mathcal{P}(\gamma)$. For technical reasons, we consider additional productions $\top \rightarrow \mathbf{R}^\alpha$, $\top \rightarrow \mathbf{R}^\beta$ and $\mathbf{F}^\beta \rightarrow \bot$ for each initial tree α and each auxiliary tree β. No auxiliary tree can be adjoined at the two fresh nodes (top) $\top$ and (bottom) $\bot$.

* Supported in part by Plan Nacional de Investigación Cientí£ca, Desarrollo e Innovación Tecnológica (Grant TIC2000-0370-C02-01), Ministerio de Ciencia y Tecnología (Grant HP2001-0044) and Xunta de Galicia (Grant PGIDT01PXI10506PN).

 Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6), pp. 101–106. Universitá di Venezia.

3. An Earley-based Parser for TAGs

To get a better understanding of our proposal, we £rst overview the Earley-based parser $\mathbb{P}_{\mathrm{E}}$ for TAGs de£ned in (Joshi and Schabes, 1997; Alonso Pardo *et al.*, 1999). This parser does not guarantee the valid pre£x property.

Given the input string $w = a_1 \ldots a_n$ with $n \geq 0$ and a TAG grammar $\mathcal{G}$, the items in the deductive system $\mathbb{P}_{\mathrm{E}}$ will be of the form:

$$[N^\gamma \to \delta \bullet \nu, i, j \mid p, q]$$

where $N^\gamma \to \delta\nu \in \mathcal{P}(\gamma)$ and i, j, p, q are indices related to positions in the input string. The intended meaning of the indices $0 \leq i \leq j$ is that δ spans the substring $a_{i+1} \ldots a_j$. When a node in δ dominates the foot of γ, the values of p and q are known and the substring spanned by the foot is $a_{p+1} \ldots a_q$ with $i \leq p \leq q \leq j$.

As we said, we consider the input string $a_1 \ldots a_n$ is encoded with a set of hypothesis $[a_i, i-1, i]$. Furthermore, being ε the empty word, we assume that $[\varepsilon, i, i]$ trivially holds for each $0 \leq i \leq n$.

We will now introduce the set of deduction steps $\mathcal{D}_{\mathrm{E}}$ for $\mathbb{P}_{\mathrm{E}}$:

$$\mathcal{D}_{\mathrm{E}} = \mathcal{D}_{\mathrm{E}}^{\mathrm{Ini}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{Sc}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{Pred}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{AdjPred}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{FootPred}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{FootComp}} \cup \mathcal{D}_{\mathrm{E}}^{\mathrm{AdjComp}}$$

The recognition process starts by predicting every initial tree:

$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Ini}} = \frac{}{[\top \to \bullet\mathbf{R}^\alpha, 0, 0 \mid -, -]} \quad \alpha \in \boldsymbol{I}$$

Scanner deduction steps can be applied when the recognition reaches a node V^γ whose label is the empty string or a terminal symbol which matches the current symbol in the input string:

$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Sc}} = \frac{\begin{array}{c}[N^\gamma \to \delta \bullet V^\gamma\nu, i, j, p, q],\\ [\mathrm{label}(V^\gamma), j, j + |\mathrm{label}(V^\gamma)|]\end{array}}{[N^\gamma \to \delta V^\gamma \bullet \nu, i, j + |\mathrm{label}(V^\gamma)|, p, q]}$$

where $|w|$ denotes the length of the w word.

The more important deduction steps in the Earley parser for CFGs are those corresponding to predictions and completions. In the case of TAGs, we have three kinds of predictions with their associated completion deduction steps: subtree, adjunction and foot.

- Subtree prediction: This deduction step is similar to predictions in Earley's parser for CFGs. Whenever there is no mandatory adjunction on a node M^γ located in a tree γ, i.e. $\mathbf{nil} \in \mathrm{adj}(M^\gamma)$, we can continue the top-down recognition of the subtree rooted with M^γ:

$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Pred}} = \frac{[N^\gamma \to \delta \bullet M^\gamma\nu, i, j \mid p, q]}{[M^\gamma \to \bullet\upsilon, j, j \mid -, -]} \quad \mathbf{nil} \in \mathrm{adj}(M^\gamma)$$

- Subtree completion: Once the the subtree rooted with M^γ has been completely recognized, we must continue the bottom-up recognition of the elementary tree γ:

$$\mathcal{D}_{\mathrm{E}}^{\mathrm{Comp}} = \frac{\begin{array}{l}[N^\gamma \to \delta \bullet M^\gamma\nu, i, j \mid p, q],\\ [M^\gamma \to \upsilon\bullet, j, k \mid p', q']\end{array}}{[N^\gamma \to \delta M^\gamma \bullet \nu, i, k \mid p \cup p', q \cup q']} \quad \mathbf{nil} \in \mathrm{adj}(M^\gamma)$$

  where $p \cup q = p$ if $q = -$, $p \cup q = q$ if $p = -$, being unde£ned in other case.

- Adjunction prediction: Let β be an auxiliary tree that can be adjoined on a node M^γ, i.e. $\beta \in \mathrm{adj}(M^\gamma)$. When the recognition of γ reaches M^γ, a new instance of the auxiliary tree β must be predicted:

$$\mathcal{D}_{\mathrm{E}}^{\mathrm{AdjPred}} = \frac{[N^\gamma \to \delta \bullet M^\gamma\nu, i, j \mid p, q]}{[\top \to \bullet\mathbf{R}^\beta, j, j \mid -, -]} \quad \beta \in \mathrm{adj}(M^\gamma)$$

- Foot prediction: Considering that $\beta \in \text{adj}(M^\gamma)$, when the recognition of an auxiliary tree β has reached its foot, we must start the recognition of the subtree excised by the adjunction[1]:

$$\mathcal{D}_{\text{E}}^{\text{FootPred}} = \frac{[\mathbf{F}^\beta \to \bullet\bot, k, k \mid -, -]}{[M^\gamma \to \bullet\delta, k, k \mid -, -]} \;\; \beta \in \text{adj}(M^\gamma)$$

- Foot completion: Once the recognition of the excised subtree rooted with M^γ is exhausted we must continue with the recognition of the auxiliary tree β that has been adjoined:

$$\mathcal{D}_{\text{E}}^{\text{FootComp}} = \frac{\begin{array}{c}[M^\gamma \to \delta\bullet, k, l \mid p, q],\\ [\mathbf{F}^\beta \to \bullet\bot, k, k \mid -, -]\end{array}}{[\mathbf{F}^\beta \to \bot\bullet, k, l \mid k, l]} \;\; \beta \in \text{adj}(M^\gamma)$$

- Adjunction completion: Once the recognition of the auxiliary tree β is exhausted, we must continue the recognition of the tree γ where the adjunction was performed:

$$\mathcal{D}_{\text{E}}^{\text{AdjComp}} = \frac{\begin{array}{c}[\top \to \mathbf{R}^\beta\bullet, j, m \mid k, l],\\ [M^\gamma \to \upsilon\bullet, k, l \mid p, q],\\ [N^\gamma \to \delta \bullet M^\gamma\nu, i, j \mid p', q']\end{array}}{[N^\gamma \to \delta M^\gamma \bullet \nu, i, m \mid p \cup p', q \cup q']} \;\; \beta \in \text{adj}(M^\gamma)$$

The input string $a_1 \ldots a_n$ belongs to the language de£ned by the grammar if and only if for some $\alpha \in \boldsymbol{I}$ is obtained a £nal item:

$$[\top \to \mathbf{R}^\alpha\bullet, 0, n \mid -, -]$$

4. A Left Corner Parser for TAGs

In order to extend the left corner parser for CFGs to the case of TAGs, we need to de£ne the left corner relation on elementary trees, taking into account that we can not miss any admissible adjunction during the recognition. Therefore, an item

$$[N^\gamma \to \delta \bullet M^\gamma\nu, i, j \mid p, q]$$

must be deduced if there exists an auxiliary tree that can be attached to M^γ, even when δ is empty.

Given an elementary tree γ, we say that M^γ is a left corner of N^γ, denoted $N^\gamma >_\ell M^\gamma$, if and only if $N^\gamma \to M^\gamma\mu \in P(\gamma)$ and M^γ is a node with a null adjoining constraint. As usual, we will denote with $>_\ell^*$ the re¤exive and transitive closure of the left corner relation.

Informally, left corner relation for TAGs goes down on nodes of elementary trees starting on a node labeled with a non-terminal symbol and ending on an adjunction node, i.e, nodes where an adjunction can be performed. When there not exists such adjunction node, the left corner relation can also end in a $\bot$ node or a node whose label is a terminal symbol or the empty word ε. As it is the case in CFGs parser, the left corner relation for TAGs only depends on the grammar, and it can be computed and stored before applying the parser.

We will go to the de£nition of the left corner parser $\mathbb{P}_{\text{LC}}$ for TAGs. The set of items and hypothesis for $\mathbb{P}_{\text{LC}}$ is the same as $\mathbb{P}_{\text{E}}$. Left corner relation is applied only in the case of predictive deduction steps. Therefore, while $\mathcal{D}_{\text{E}}^{\text{Ini}}$, $\mathcal{D}_{\text{E}}^{\text{Sc}}$, $\mathcal{D}_{\text{E}}^{\text{Comp}}$,$\mathcal{D}_{\text{E}}^{\text{FootComp}}$ and $\mathcal{D}_{\text{E}}^{\text{AdjComp}}$ remain the same in the left corner parser, we must replace the following ones: $\mathcal{D}_{\text{E}}^{\text{Pred}}$, $\mathcal{D}_{\text{E}}^{\text{AdjPred}}$ and $\mathcal{D}_{\text{E}}^{\text{FootPred}}$.

4.1. Filtering Subtree Predictions

We now introduce the following deduction steps ($\mathcal{D}_{\text{LC}}^{\text{PredLC}}$, $\mathcal{D}_{\text{LC}}^{\text{Pred}'}$, $\mathcal{D}_{\text{LC}}^{\text{CompLC}}$) replacing $\mathcal{D}_{\text{E}}^{\text{Pred}}$. Given a subtree rooted with M^γ where no adjunction is mandatory, these new steps £lter subtree predictions applied on nodes that are left corners of M^γ.

1. The valid pre£x property is not ful£lled due to $\mathcal{D}_{\text{E}}^{\text{FootPred}}$ since every subtree rooted with a node M^γ where $\beta \in \text{adj}(M^\gamma)$ is introduced in the recognition.

- In the case that $M^\gamma >_\ell^* O^\gamma$ and the left-most daughter of O^γ is labeled with a terminal symbol or ε, we can go down on the tree directly to that node:

$$\mathcal{D}_{\mathrm{LC}}^{\mathrm{PredLC}} = \frac{\begin{array}{c}[N^\gamma \to \delta \bullet M^\gamma \nu, i, j, p, q],\\ [\text{label}(V^\gamma), j, j + |\text{label}(V^\gamma)|]\end{array}}{[O^\gamma \to V^\gamma \bullet \upsilon, j, j + |\text{label}(V^\gamma)|, -, -]}$$

- In the case that $M^\gamma >_\ell^* O^\gamma$ and O^γ is a node labeled with a non terminal symbol whose left-most daughter P^γ is an adjunction node, we will stop at that node:

$$\mathcal{D}_{\mathrm{LC}}^{\mathrm{Pred}'} = \frac{[N^\gamma \to \delta \bullet M^\gamma \nu, i, j, p, q]}{[O^\gamma \to \bullet P^\gamma \upsilon, j, j, -, -]}$$

- In the bottom-up traversal we should go up on those nodes O^γ and Q^γ that are left corners of M^γ:

$$\mathcal{D}_{\mathrm{LC}}^{\mathrm{CompLC}} = \frac{\begin{array}{c}[N^\gamma \to \delta \bullet M^\gamma \nu, i, j, p, q],\\ [O^\gamma \to \omega \bullet, j, k, p', q']\end{array}}{[Q^\gamma \to O^\gamma \bullet \upsilon, j, k, p', q']}$$

4.2. Filtering Adjunction Predictions

We now explain the set of deduction steps ($\mathcal{D}_{\mathrm{LC}}^{\mathrm{AdjPredLC}}$, $\mathcal{D}_{\mathrm{LC}}^{\mathrm{AdjPred}'}$ and $\mathcal{D}_{\mathrm{LC}}^{\mathrm{AdjCompLC}}$) replacing $\mathcal{D}_{\mathrm{E}}^{\mathrm{AdjPred}}$. Let M^γ be a node in an elementary tree γ where the auxiliary tree β can be adjoined. These new deduction steps £lter predictions on those nodes that are left corners of the top node of β.

- When $\top >_\ell^* O^\beta$ and the left-most daughter of O^β is a node V^β labeled with a terminal symbol or ε, we will apply:

$$\mathcal{D}_{\mathrm{LC}}^{\mathrm{AdjPredLC}} = \frac{\begin{array}{c}[N^\gamma \to \delta \bullet M^\gamma \nu, i, j, p, q],\\ [\text{label}(V^\beta), j, j + |\text{label}(V^\beta)|]\end{array}}{[O^\beta \to V^\beta \bullet \upsilon, j, j + |\text{label}(V^\beta)|, -, -]}$$

- When $\top >_\ell^* O^\beta$ and O^β dominates on the left a node P^β such that either P^β is an adjunction node or P^β is the $\perp$ node of β, we will apply:

$$\mathcal{D}_{\mathrm{LC}}^{\mathrm{AdjPred}'} = \frac{[N^\gamma \to \delta \bullet M^\gamma \nu, i, j, p, q]}{[O^\beta \to \bullet P^\beta \upsilon, j, j, -, -]}$$

- During the bottom-up recognition we must go up on the tree through the nodes O^β that are left corners of the top node of β:

$$\mathcal{D}_{\mathrm{LC}}^{\mathrm{AdjCompLC}} = \frac{\begin{array}{c}[N^\gamma \to \delta \bullet M^\gamma \nu, i, j, p, q],\\ [O^\beta \to \omega \bullet, j, k, p', q']\end{array}}{[Q^\beta \to O^\beta \bullet \upsilon, j, k, p', q']}$$

4.3. Filtering Foot Predictions

We now show the set of deduction steps ($\mathcal{D}_{\mathrm{LC}}^{\mathrm{FootPredLC}}$, $\mathcal{D}_{\mathrm{LC}}^{\mathrm{FootPred}'}$ and $\mathcal{D}_{\mathrm{LC}}^{\mathrm{FootCompLC}}$) replacing $\mathcal{D}_{\mathrm{E}}^{\mathrm{FootPred}}$. Let M^γ be a node in an elementary tree γ such that β can be adjoined. Suppose the recognition has reached a node E^β where it is not mandatory to perform an adjunction and that $E^\beta >_\ell^* \perp$. These new deduction steps £lter predictions on nodes belonging to the auxiliary tree β and to the elementary tree γ where the adjunction is performed:

- When $M^\gamma >_\ell^* O^\gamma$ and the left-most daughter of O^γ is a node V^γ labeled with a terminal symbol or ε we will apply:

$$\mathcal{D}_{\mathrm{LC}}^{\mathrm{FootPredLC}} = \frac{\begin{array}{c}[D^\beta \to \delta \bullet E^\beta \nu, j, k, -, -],\\ [\text{label}(V^\gamma), k, k + |\text{label}(V^\gamma)|]\end{array}}{[O^\gamma \to V^\gamma \bullet \upsilon, k, k + |\text{label}(V^\gamma)|, -, -]}$$

Sentence	Time Reduction	Items Reduction
Srini bought a book	-3%	-44%
Srini bought Beth a book	-5%	-47%
Srini bought a book at the bookstore	-6%	-46%
he put the book on the table	-8%	-44%
$*$ he put the book	-13%	-42%
the sun melted the ice	-11%	-48%
the ice melted	-14%	-46%
Elmo borrowed a book	-7%	-45%
$*$ a book borrowed	+5%	-41%
he hopes Muriel wins	-12%	-49%
he hopes that Muriel wins	-16%	-49%
the man who Muriel likes bought a book	+6%	-42%
the man that Muriel likes bought a book	-2%	-44%
the music should have been being played for the president	-28%	-56%
Clove caught a frisbee	-4%	-45%
who caught a frisbee	-7%	-45%
what did Clove catch	-13%	-49%
the aardvark smells terrible	-4%	-46%
the emu thinks that the aardvark smells terrible	-12%	-48%
who does the emu think smells terrible	-14%	-49%
who did the elephant think the panda heard the emu said smells terrible	-14%	-49%
Herbert is angry	-24%	-53%
Herbert is angry and furious	-21%	-54%
Herbert is more livid than angry	-25%	-51%
Herbert is more livid and furious than angry	-18%	-50%

Table 1: Results of the experiment based on XTAG

- In the case $M^\gamma >_\ell^* O^\gamma$ and the left-most daughter of O^γ is a node P^γ such that either P^γ is an adjunction node or P^γ is the $\perp$ node of γ, prediction is stopped at P^γ:

$$\mathcal{D}_{\mathrm{LC}}^{\mathrm{FootPred}'} = \frac{[D^\beta \to \delta \bullet E^\beta \nu, j, k, -, -]}{[O^\gamma \to \bullet P^\gamma \upsilon, k, k, -, -]}$$

- During the bottom-up recognition we must go up on the tree through the nodes O^γ that are left corners of M^γ, the node where the adjunction was performed, guaranteeing we do not go up beyond M^γ itself, i.e. $M^\gamma \neq O^\gamma$:

$$\mathcal{D}_{\mathrm{LC}}^{\mathrm{FootCompLC}} = \frac{\begin{array}{l}[D^\beta \to \delta \bullet E^\beta \nu, j, k, -, -],\\ [O^\gamma \to \omega \bullet, k, l, p, q]\end{array}}{[Q^\gamma \to O^\gamma \bullet \upsilon, k, l, p, q]}$$

5. Experimental results

The time complexity of the algorithm with respect to the length n of the input string is $O(n^6)$ for both parsers. The improvement in the performance of Left Corner parsers comes from the reduction in the size of the chart (the set of deduced items). It is clear that this reduction depends on the grammar and the input string considered. We have made a preliminary study where we have tested and compared the behavior of the LC parser and the Earley-based parser explained before.

We have incorporated both parsers into a naive implementation in Prolog of the deductive parsing machine presented in (Shieber, Schabes and Pereira, 1995). We have taken a subset of the XTAG grammar (XTAG Research Group, 2001), consisting of 27 elementary trees that cover a variety of English constructions: relative clauses, auxiliary verbs, unbounded dependencies, extraction, etc. In order to eliminate the time spent by uni£cation, we have not considered the feature structures of elementary trees. Instead, we have simulated the features using local constraints. Every sentence has been parsed without previous £ltering of elementary trees. Table 1 includes the reduction ratio with respect to the parsing time in seconds and with respect to the chart size. Brie¤y, we can remark that LC parser shows on average a time reduction of 11% and a chart size reduction of 50%.

6. Conclusion

We have de£ned a new parser for TAG that is an extension of the Left Corner parser for Context Free Grammars. The new parser can be view as a £lter on an Earley-based parser for TAGs where the number of predictions is reduced due to the generalized left corner relation that we have established on the nodes of elementary trees. The worst-case complexity with respect to space and time is the standard one for TAG parsing, but preliminary experiments have shown a better performance than classical Earley-based parsers for TAG. Finally, as further work, we are investigating the conditions the parser should satisfy in order to guarantee the valid pre£x property.

References

Alonso Pardo, Miguel A., David Cabrero Souto, Eric de la Clergerie and Manuel Vilares Ferro. 1999. Tabular Algorithms for TAG Parsing. In *Proc. of EACL'99, Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 150–157, Bergen, Norway, June. ACL.

Evans, Roger and David Weir. 1998. A structure-sharing parser for lexicalized grammars. In *COLING-ACL'98, 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Proceedings of the Conference*, volume I, pages 372–378, Montreal, Quebec, Canada, August. ACL.

Joshi, Aravind K. and Yves Schabes. 1997. Tree-Adjoining Grammars. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages. Vol 3: Beyond Words*. Springer-Verlag, Berlin/Heidelberg/New York, chapter 2, pages 69–123.

Lang, Bernard. 1990. The systematic construction of Earley parsers: Application to the production of $\mathcal{O}(n^6)$ Earley parsers for tree adjoining grammars. In *Proc. of the 1st International Workshop on Tree Adjoining Grammars*, August.

Nederhof, Mark-Jan. 1999. The Computational Complexity of the Correct-Pre£x Property for TAGs. *Computational Linguistics*, 25(3):345–360.

Schabes, Yves and Richard C. Waters. 1995. Tree Insertion Grammar: A Cubic-Time Parsable Formalism That Lexicalizes Context-Free Grammar Without Changing the Trees Produced. *Computational Linguistics*, 21(4):479–513, December. Also as Technical Report TR-94-13, June 1994, Mitsubishi Electric Research Laboratories, Cambridge, MA, USA.

Shieber, Stuart M., Yves Schabes and Fernando C. N. Pereira. 1995. Principles and Implementation of Deductive Parsing. *Journal of Logic Programming*, 24(1–2):3–36, July-August.

Sikkel, Klaas. 1997. *Parsing Schemata — A Framework for Speci£cation and Analysis of Parsing Algorithms*. Texts in Theoretical Computer Science — An EATCS Series. Berlin/Heidelberg/New York: Springer-Verlag.

XTAG Research Group. 2001. *A Lexicalized Tree Adjoining Grammar for English*. Technical Report IRCS-01-03, University of Pennsylvania, USA.