Parsing as Deduction

Joseph Kühner

Department of Computer Science,
Saarland University, D-66041 Saarbrücken, Germany
`jrkuehner@t-online.de`
`http://www.ps.uni-sb.de`

Abstract. Parsing algorithms for various types of languages are represented in a formal logic framework as deduction systems, where items (formulas) describe the grammatical status of strings, and inference rules produce new items from already generated items. On this more abstract level, Parsing Deduction Systems reflect the structure of parsers in a clear and concise manner and provide unified tools for the proof of correctness, completeness and complexity analysis.

Key words: Parsing, Deduction, Axioms, Inference, Correctness, Completeness, Complexity

1 Introduction

These notes are an elaborate version of the authors's talk about Parsing as Deduction at the Formal Grammars Seminar of the Programing Systems Lab, Informatics, Saarland University.

The aim of this talk was to present the principles of deductive parsing and to describe parsing reduction systems for various types of formalisms, including context free grammars and tree adjoining grammars. Proofs of correctness and completeness for these parsing deduction systems are detailed. A single deduction algorithm is introduced which can be applied to implement all these various deduction systems. It is proved to be correct and complete.

1.1 Basic Notation

We present parsing algorithms as deductive processes in which rules of inference are used to derive statements about the grammatical status of strings from other such statements. Statements are represented by formulas in a suitable formal language.

A deduction system is defined by a set of items, subsets of distiguishes items called axioms resp. goals and a set of inference rules given by appropriate formula schemata.

The general form of a rule of inference is

$$\frac{A_1 \ldots A_k}{B} \quad < \text{side conditions on } A_1, \ldots, A_k, B > .$$

The antecedents $A_1, \ldots A_k$ and the consequent B of the rule are items. Axioms can be represented as inference rules without antecedents.

Given a deduction system, a derivation of an item B from assumptions $A_1, \ldots, A_m$ is a sequence of items $S_1, \ldots S_n$ such that $S_n = B$, and each S_i is either an axiom or there is a rule of inference R and items $S_{i_1}, \ldots, S_{i_k}$ with $i_1, \ldots, i_k < i$ such that $S_{i_1}, \ldots, S_{i_k}$ match the antecedents of R and S_i matches the consequent and the side conditions are satisfied. We write $A_1, \ldots, A_m \vdash B$ and say that B is a consequence of $A_1, \ldots, A_m$ if such a derivation exists. If B is the consequence of the empty set of assumptions, it is said to be derivable and we write $\vdash B$.

2 Deductive Parsing of Context-Free Grammars

In this section various parsing algorithms for context-free grammars are presented as parsing deduction system.

Let $\mathcal{G} = (N, \Sigma, P, S)$ be a context free grammar (CFG). We use standard notation for meta-variables ranging over the objects under discussion: uppercase Latin letters A, B, C … for nonterminals, lowercase Latin letters a, b, c … for terminals, lowercase Greek letters α, β, γ … for strings of nonterminals or terminals, ϵ for the empty string. We write $\rightarrow$ for direct derivation and $\overset{*}{\rightarrow}$ for its reflexive, transitive closure.

2.1 The CYK Parsing Algorithm

In this section, we recall the CYK parser. Let $\mathcal{G} = (N, \Sigma, P, S)$ be a context free grammar (CFG) in Chomsky-Normal-Form (CNF). This means that there are only two types of production rules in $\mathcal{G}$:

$$A \rightarrow a \quad \text{and} \quad A \rightarrow BC$$

where a is a terminal and A, B, C are nonterminals in $\mathcal{G}$. Let $w = w_1 \ldots w_n \in \Sigma^*$ be a string to be parsed. The algorithm constructs sets T_{ij}, $(1 \leq i \leq j \leq n)$, of nonterminals in the following manner:

For $1 \leq i \leq n$ a nonterminal A belongs to the set T_{ii} if and only if $A \rightarrow w_i$ is a production.

For $1 \leq i < j \leq n$ a nonterminal A belongs to the set T_{ij} if and only if there is an index k, $i \leq k \leq j - 1$ and a production $A \rightarrow BC$ such that $B \in T_{ik}$ and $C \in T_{k+1,j}$.

This could be implemented using nested loops. Correctness and completeness can be showed by verification of suitable loop invariants. The complexity is $\mathcal{O}(|P|n^3)$.

2.2 CYK Parsing as Deduction

Let $\mathcal{G} = (N, \Sigma, P, S)$ be a CFG in CNF, $w = w_1 \ldots w_n$ a string in Σ^* to be parsed. Consider items (formulas) $[A, i, j]$, $A \in N$, $1 \leq i \leq j \leq n$, which are meant to state that $A \xrightarrow{*} w_i \ldots w_j$.

For each terminal w_i in w and each production $A \to w_i$, it is clear that the item $[A, i, i]$ makes a true claim, so that such items can be taken as axiomatic.

Let $[B, i, k]$ and $[C, k+1, j]$ be items and $A \to BC$ a production. Since these items assert $B \xrightarrow{*} w_i \ldots w_k$ and $C \xrightarrow{*} w_{k+1} \ldots w_j$, it is sound to conclude $A \xrightarrow{*} w_i \ldots w_j$. This argument can be coded as rule of inference.

$$\frac{[B, i, k]\ [C, k+1, j]}{[A, i, j]} \quad < A \to BC >$$

If, starting with axioms and using inference rules, an item $[S, 1, n]$ is derivable, we can conclude that the string $w = w_1 \ldots w_n$ is admitted by the grammar, since it asserts $A \xrightarrow{*} w_i \ldots w_j$. We call this item goal item.

In summary, the CYK deduction system can be specified with four components: a class of items, a set of axioms, a set of inference rules and a subclass of items, the goal items. These are given in summary form in table 1. We give a formal proof of the correctness an completeness of this deductive parser.

Table 1. CYK deduction System

Item Form:	$[A, i, j]$	
Axioms:	$\frac{}{[A, i, i]}$	$< A \to w_i >$
Inference Rules:	$\frac{[B, i, k]\ [C, k+1, j]}{[A, i, j]}$	$< A \to BC >$
Goals:	$[S, 1, n]$	

Lemma 1. *Let $[A, i, j]$ be a derivable item in the parsing deduction system specified in table 1. Then in grammar $\mathcal{G}$ we have $A \xrightarrow{*} w_i \ldots w_j$.*

Proof. We proceed by induction on the number of steps which are needed to infer item $[A, i, j]$ from the axioms.

If the item is an axiom $[A, i, i]$ then, by the side condition, there is a production rule $A \to w_i$ in the grammar.

If the item is not an axiom, then there is an index $i \leq k \leq j-1$, and a production $A \to BC$ and items $[B, i, k]$ and $[C, k+1, j]$ which derive $[A, i, j]$ using the inference rule

$$\frac{[B, i, k]\ [C, k+1, j]}{[A, i, j]}.$$

The number of steps needed to derive items $[B, i, k]$ and $[C, k+1, j]$ is smaller than the number of derivation steps for $[A, i, j]$, so we can apply the induction hypothesis and conclude that $B \xrightarrow{*} w_i \dots w_k$ and $C \xrightarrow{*} w_{k+1} \dots w_j$. But this implies the claim: $A \xrightarrow{*} w_i \dots w_j$.

Correctness follows from application of lemma 1 to the goal item $[S, 1, n]$. Indeed if the goal item can be derived, we have $S \xrightarrow{*} w_1 \dots w_n$.

Corollary 1. *If the goal item $[S, 1, n]$ can be derived in the parsing deduction system, then the string $w = w_1 \dots w_n$ is accepted by the grammar $\mathcal{G}$.*

Lemma 2. *If in grammar $\mathcal{G}$ we have $A \xrightarrow{*} w_i \dots w_j$, then item $[A, i, j]$ can be derived in the parsing deduction system.*

Proof. We proceed by induction on the length of the derivation. If the string is derived in one step, then it must be a production $A \rightarrow w_i$. But then the item $[A, i, i]$ is an axiom.

If the string is derived in more than one step, there is a production $A \rightarrow BC$ and an index $i \leq k \leq j-1$ such that $B \xrightarrow{*} w_i \dots w_k$ and $C \xrightarrow{*} w_{k+1} \dots w_j$. By induction the items $[B, i, k]$ and $[C, k+1, j]$ are generated in the deduction system. Now the inference rule $\dfrac{[B, i, k]\ [C, k+1, j]}{[A, i, j]}$ can applied to generate item $[A, i, j]$.

Corollary 2. *If the string $w = w_1 \dots w_n$ is accepted by the grammar $\mathcal{G}$, the goal item $[S, 1, n]$ can be generated in the deduction system.*

Proof. Apply lemma 2 to $S \xrightarrow{*} w_1 \dots w_n$.

2.3 Pure Top Down Parsing

In this section we present recursive-decent parsing in this logical perspective. Let $\mathcal{G} = (N, \Sigma, P, S)$ be an arbitrary CFG, $w = w_1 \dots w_n$ a string in Σ^* to be parsed. Consider items (formulas) $[\bullet\, \beta, j]$, where $0 \leq j \leq n$. Such an item asserts that the substring $w_1 \dots w_j$ followed by β is a sentential form of the language, that is, that $S \xrightarrow{*} w_1 \dots w_j \beta$.

Since $S \xrightarrow{*} S$ is trivially true, we start with axiom $[\bullet\, S, 0]$. Note that items of the form $[\bullet\, w_{j+1}\beta, j]$ and $[\bullet\, \beta, j+1]$ make the same claim, namely $S \xrightarrow{*} w_1 \dots w_j w_{j+1} \beta$. This motivates the following inference rule, called scanning rule:

$$\frac{[\bullet\, w_{j+1}\beta, j]}{[\bullet\, \beta, j+1]}.$$

If we have a sentential form of type $S \xrightarrow{*} w_1 \dots w_j B\beta$ and a production $B \rightarrow \gamma$ in the grammar, we can infer the sentential form $S \xrightarrow{*} w_1 \dots w_j \gamma\beta$. This leads to the prediction inference rule:

$$\frac{[\bullet\, B\beta, j]}{[\bullet\, \gamma\beta, j]}.$$

In summary, we can write down the following parsing deduction system for top down parsing in table 2. We give a formal proof of the correctness and

Table 2. Top-Down Deduction System

Item Form:	$[\bullet\, \beta, j]$	
Axioms:	$\overline{[\bullet\, S, 0]}$	
Scanning Rule:	$\dfrac{[\bullet\, w_{j+1}\beta, j]}{[\bullet\, \beta, j+1]}$	
Prediction Rule:	$\dfrac{[\bullet\, B\beta, j]}{[\bullet\, \gamma\beta, j]}$	$< B \rightarrow \gamma >$
Goals:	$[\bullet, n]$	

completeness of this deductive parser.

Lemma 3. *Let $[\bullet\, \beta, j]$ be an item which can be derived in the parsing deduction system specified in table 2. Then in grammar $\mathcal{G}$ we have $S \overset{*}{\rightarrow} w_1 \ldots w_j\beta$.*

Proof. It is clear that this is true for the axiom $[\bullet\, S, 0]$.

If the item $[\bullet\, \beta, j]$ is not an axiom it might result from an item $[\bullet\, w_j\beta, j-1]$ by application of the scanning rule. By induction, $[\bullet\, w_j\beta, j-1]$ implies the sentential form $S \overset{*}{\rightarrow} w_1 \ldots w_{j-1}w_jB\beta$, which proofs the claim.

The item may also result from prediction. In this case there are strings β' and γ such that $\beta = \gamma\beta'$ and a production $B \rightarrow \gamma$ such that item $[\bullet\, B\beta', j]$ infers $[\bullet\, \gamma\beta', j] = [\bullet\, \beta, j]$. By induction $S \overset{*}{\rightarrow} w_1 \ldots w_jB\beta'$, which by application of the production $B \rightarrow \gamma$ proofs the claim.

From lemma 3 applied to the goal item, correctness follows.

Corollary 3. *If the goal item $[\bullet, n]$ can be derived in the parsing deduction system specified in table 2, then the string $w = w_1 \ldots w_n$ is accepted by the grammar $\mathcal{G}$.*

Lemma 4. *Let $S \overset{*}{\rightarrow} w_1 \ldots w_j\beta$. be a sentential form in grammar $\mathcal{G}$, then item $[\bullet\, \beta, j]$ can be derived in the parsing deduction system specified in table 2.*

Proof. Define the rank of $S \overset{*}{\rightarrow} w_1 \ldots w_j\beta$ to be the sum of j and the length of a shortest leftmost derivation of β. The proof will proceed by induction on the rank.

If the rank of $S \overset{*}{\rightarrow} w_1 \ldots w_j\beta$ is zero, we have $j = 0$ and $\beta = S$. But then the corresponding item is $[\bullet\, S, 0]$ which is an axiom in the deduction system.

Let the rank r of $S \overset{*}{\rightarrow} w_1 \ldots w_j\beta$ be greater than 0 and suppose that the lemma is true for all instances of rank smaller than r. Then there are two cases.

Case 1: $S \xrightarrow{*} w_1 \dots w_j \beta$ may be a production. Then we apply the prediction rule to the axiom $[\bullet\, S, 0]$ to infer $[\bullet\, w_1 \dots w_j \beta, 0]$. Repeatedly scanning the terminals $w_1, \dots, w_j$ will finally derive item $[\bullet\, \beta, j]$.

Case 2: $S \xrightarrow{*} w_1 \dots w_j \beta$ in more than one step. Then we can decompose $\beta = \gamma\beta'$ and find a production $B \rightarrow w_{j-k+1} \dots w_j \gamma$ in $\mathcal{G}$ such that the derivation splits in the following way:

$$S \xrightarrow{*} w_1 \dots w_{j-k} B \beta' \rightarrow w_1 \dots w_j \beta.$$

Since the rank of $S \xrightarrow{*} w_1 \dots w_{j-k} B \beta'$ is strictly less than r, we can apply the induction hypothesis to generate item $[\bullet\, B\beta', j-k]$. By prediction the item $[\bullet\, w_{j-k} \dots w_j \gamma \beta', j-k]$ will be generated. Finally by k applications of the scanning rule the item $[\bullet\, \beta, j]$ will be inferred and the claim is proved.

Corollary 4. *If the string $w = w_1 \dots w_n$ is accepted by the grammar $\mathcal{G}$, the goal item $[\bullet, n]$ can be generated in the parsing deduction system.*

Proof. Apply lemma 4 to $S \xrightarrow{*} w_1 \dots w_n$.

2.4 Pure Bottom-Up Parsing

To construct a pure bottom-up parsing deduction system consider items of the form $[\alpha \bullet, j]$. Such an item asserts $\alpha w_{j+1} \dots w_n \xrightarrow{*} w_1 \dots w_n$ or equivalently $\alpha \xrightarrow{*} w_1 \dots w_j$. The deduction system is specified in table 3

Table 3. Pure Bottom-Up Deduction System

Item Form:	$[\alpha \bullet, j]$	
Axioms:	$\dfrac{}{[\bullet, 0]}$	
Shift:	$\dfrac{[\alpha \bullet, j]}{[\alpha w_{j+1} \bullet, j+1]}$	
Reduce:	$\dfrac{[\alpha\gamma \bullet, j]}{[\alpha B \bullet, j]}$	$< B \rightarrow \gamma >$
Goals:	$[S \bullet, n]$	

Lemma 5. *Let $[\alpha \bullet, j]$ be an item which can be derived in the parsing deduction system specified in table 3, then in grammar $\mathcal{G}$ we have $\alpha w_{j+1} \dots w_n \xrightarrow{*} w_1 \dots w_n$.*

Proof. The proof works by induction on the number of steps to infer the item. If the item can be derived in one step, it must be an axiom $[\bullet, 0]$. Then trivially $w_1 \dots w_n \xrightarrow{*} w_1 \dots w_n$.

If the number of steps to derive item $[\alpha \bullet, j]$ is greater than 1 there are two cases.

Suppose $[\alpha \bullet, j]$ is inferred by application of the shift rule. Then $\alpha = \alpha' w_j$ where the item $[\alpha' \bullet, j-1]$ is the antecedent of $[\alpha' w_j \bullet, j]$. By hypothesis of induction $\alpha' w_j \dots w_n \xrightarrow{*} w_1 \dots w_n$ is a derivation in $\mathcal{G}$, but this implies immediately the claim: $(\alpha' w_j) w_{j+1} \dots w_n = \alpha w_{j+1} \dots w_n \xrightarrow{*} w_1 \dots w_n$.

The item $[\alpha \bullet, j]$ may also result from application of the reduction rule. Then there is a production $B \to \gamma$ such that $\alpha = \alpha' B$ and $[\alpha' \gamma \bullet, j]$ is the antecedent of $[\alpha' B \bullet, j]$. By hypothesis of induction we have $\alpha' \gamma w_{j+1} \dots w_n \xrightarrow{*} w_1 \dots w_n$, but this implies $\alpha' B w_{j+1} \dots w_n \to \alpha' \gamma w_{j+1} \dots w_n \xrightarrow{*} w_1 \dots w_n$.

If we apply lemma 5 to the goal item, we can proof the correctness of the deduction system.

Corollary 5. *Suppose that the goal item $[S \bullet, n]$ can be derived in the deduction system, then $S \xrightarrow{*} w_1 \dots w_n$.*

Lemma 6. *Suppose $\alpha w_{j+1} \dots w_n \xrightarrow{*} w_1 \dots w_n$, then the item $[\alpha \bullet, j]$ can be derived in the deduction system.*

Proof. We proceed by induction on the length of a minimal reversed rightmost derivation. If $\alpha w_{j+1} \dots w_n \xrightarrow{*} w_1 \dots w_n$ in one step then $\alpha = A$, a nonterminal, and $A \to w_1 \dots w_j$ is a production in grammar $\mathcal{G}$. Applying j shift steps to the axiom $[\bullet, 0]$ yields item $[w_1 \dots w_j \bullet, j]$ and one application of the reduction rule infers the item $[A \bullet, j]$.

If $\alpha w_{j+1} \dots w_n \xrightarrow{*} w_1 \dots w_n$ in more than one step, there is a production $B \to \gamma w_{l+1} \dots w_j$, where $1 \le l \le j-1$ and a decomposition $\alpha = \alpha' B$ and $\alpha' B \to \alpha' \gamma w_{l+1} \dots w_j \xrightarrow{*} w_1 \dots w_j$. We can apply the induction hypothesis to the derivation $\alpha' \gamma \xrightarrow{*} w_1 \dots w_l$ to derive the item $[\alpha' \gamma \bullet, l]$. Repeatedly applying the shift rule yields the item $[\alpha' \gamma w_{l+1} \dots w_j \bullet, j]$ and one application of the reduction rule results in $[\alpha' B \bullet, j] = [\alpha \bullet, j]$, our claim.

Now the completeness of the deduction system follows from lemma 6.

Corollary 6. *If $S \xrightarrow{*} w_1 \dots w_n$ the goal item $[S \bullet, n]$ can be generated in the deduction system.*

2.5 Earley's Algorithm

The items in Earley's algorithm are dotted rules $[i, A \to \alpha \bullet \beta, j]$ where $A \to \alpha\beta$ is a production and $1 \le i \le j \le n$ denote positions in the string $w = w_1 \dots w_n$. Such an item asserts the top-down claim $S \xrightarrow{*} w_1 \dots w_i A \gamma$ for some string γ, and the bottom-up claim $\alpha w_{j+1} \dots w_n \xrightarrow{*} w_{i+1} \dots w_n$ or equivalently $\alpha \xrightarrow{*} w_{i+1} \dots w_j$. For technical reasons we introduce an auxiliary production $S' \to S$. The deduction system is specified in the table 4.

Table 4. Earley's Deduction System

Item Form:	$[i, A \to \alpha \bullet \beta, j]$	
Axioms:	$\dfrac{}{[0, S' \to \bullet S, 0]}$	
Scanning:	$\dfrac{[i, A \to \alpha \bullet w_{j+1}\beta, j]}{[i, A \to \alpha w_{j+1} \bullet \beta, j+1]}$	
Prediction:	$\dfrac{[i, A \to \alpha \bullet B\beta, j]}{[j, B \to \bullet \gamma, j]}$	$< B \to \gamma >$
Completion:	$\dfrac{[i, A \to \alpha \bullet B\beta, k] \quad [k, B \to \gamma \bullet, j]}{[i, A \to \alpha B \bullet \beta, j]}$	
Goals:	$[0, S' \to S \bullet, n]$	

Lemma 7. *If the item $[i, A \to \alpha \bullet \beta, j]$ can be generated in the deduction system specified in table 4, then in grammar $\mathcal{G}$ we have: $S \xrightarrow{*} w_1 \dots w_i A\gamma$ for some string γ and $\alpha w_{j+1} \dots w_n \xrightarrow{*} w_{i+1} \dots w_n$.*

Proof. The proof proceeds by induction on the number of steps to derive the item from the axiom.

The axiom item itself $[0, S' \to \bullet S, 0]$ asserts $S \xrightarrow{*} S$ and $w_1 \dots w_n \xrightarrow{*} w_1 \dots w_n$ which are trivially true.

Let be $[i, A \to \alpha \bullet \beta, j]$ an item. Several cases are to be considered:

Case 1. The latest inference rule to generate the item was scanning. Then $\alpha = \alpha' w_j$ and our item resulted from

$$\frac{[i, A \to \alpha' \bullet w_j\beta, j-1]}{[i, A \to \alpha' w_j \bullet \beta, j]}.$$

The induction hypothesis can be applied to the antecedent, so we have $S \xrightarrow{*} w_1 \dots w_i A\gamma$ for some string γ and $\alpha' w_j \dots w_n \xrightarrow{*} w_{i+1} \dots w_n$. The claim follows from the identity $(\alpha' w_j) w_{j+1} \dots w_n = \alpha w_{j+1} \dots w_n$.

Case 2. The latest inference rule to generate the item was completion. Then $\alpha = \alpha' B$, where B is a nonterminal and our item resulted from

$$\frac{[i, A \to \alpha' \bullet B\beta, k] \quad [k, B \to \gamma \bullet, j]}{[i, A \to \alpha' B \bullet \beta, j]}.$$

The induction hypothesis can be applied to both antecedents, so we have $S \xrightarrow{*} w_1 \dots w_i A\gamma$ for some string δ and $\alpha' w_{k+1} \dots w_n \xrightarrow{*} w_{i+1} \dots w_n$ and also $S \xrightarrow{*} w_1 \dots w_k B\delta'$ for some string δ' and $\gamma w_{j+1} \dots w_n \xrightarrow{*} w_{k+1} \dots w_n$. The first claim for our item is trivially true, the second results from

$$\alpha' B w_{j+1} \dots w_n \to \alpha' \gamma w_{j+1} \dots w_n \xrightarrow{*} \alpha' w_{k+1} \dots w_n \xrightarrow{*} w_{i+1} \dots w_n.$$

Case 3. The latest inference rule to generate the item was prediction. Then $\alpha = \epsilon$, $i = j$, $A \rightarrow \gamma$ is a production and our item resulted from

$$\frac{[i, A' \rightarrow \alpha' \bullet A\beta, j]}{[j, A \rightarrow \bullet \gamma, j]}.$$

We can apply the induction hypothesis to the antecedent. Hence we have $S \overset{*}{\rightarrow} w_1 \ldots w_i A' \gamma'$ for some string γ' and $\alpha' w_{j+1} \ldots w_n \overset{*}{\rightarrow} w_{i+1} \ldots w_n$. Now

$$S \overset{*}{\rightarrow} w_1 \ldots w_i A' \gamma' \overset{*}{\rightarrow} w_1 \ldots w_i \alpha' A\beta\gamma' \overset{*}{\rightarrow} w_1 \ldots w_i w_{i+1} \ldots w_j A\beta\gamma'.$$

The second claim is trivial since $\epsilon \overset{*}{\rightarrow} \epsilon$.

If we apply lemma 7 to the goal item, we can proof the correctness of the deduction system.

Corollary 7. *Suppose that the goal item* $[0, S' \rightarrow S \bullet, n]$ *can be derived in the deduction system, then* $S \overset{*}{\rightarrow} w_1 \ldots w_n$.

3 Deduction for Tree Adjoining Grammars (TAG)

In this section we present a deduction system for parsing tree adjoining grammars.

3.1 Introduction to tree adjoining grammars

A tree adjoining grammar (TAG) is a quintuple $\mathcal{G} = (N, \Sigma, S, I, A)$ where N is a set of nonterminals, Σ a set of terminals, S a distinguished nonterminal, the start symbol, I a set of initial trees and A a set of auxiliary trees. The trees in $I \cup A$ are called elementary.

In an initial tree the root and all inner nodes are labelled with nonterminals, all frontier nodes are labelled with terminals.

In an auxiliary tree the root and all inner nodes are labelled with nonterminals; all frontier nodes are labelled with terminal symbols except one, the foot node, which is labelled with the same terminal as the root. This is represented in figure 1.

Let α be a tree with an inner node ν labelled B and β an auxiliary tree with root and foot node labelled with the same nonterminal B. Then an operation adjunction is defined in the following way: The subtree of α rooted by ν (and labelled with B) is cut off and β is inserted at ν. Then the previously excised subtree is appended at the foot node of β. Adjunction is illustrated in figures 2 and 3.

Let α be a tree, every node in α can be specified by its address, a list of integers defined recursively by:

The address of the root is the empty list *nil*; if the address of a node ν is a the address of its k-th child node is ak.

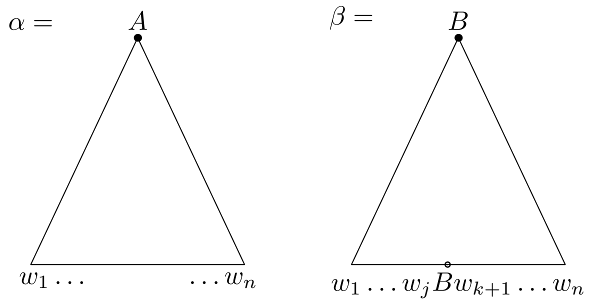

Fig. 1. *Initial tree* α: the root is labelled with a nonterminal A, all frontier nodes are labelled with terminal symbols. *Auxiliary tree* β: the root is labelled with a nonterminal B, all frontier node are labelled with terminal symbols except one, the foot node, which is labelled with the same nonterminal B as the root.

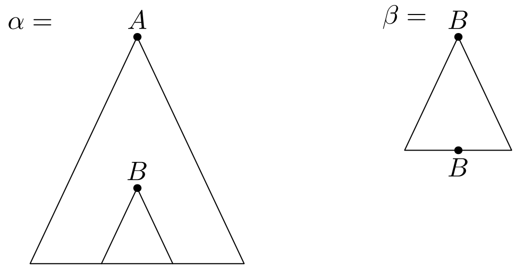

Fig. 2. Root and foot node of the auxiliary tree β are labelled B. β can be adjoint to tree α at node ν labelled B.

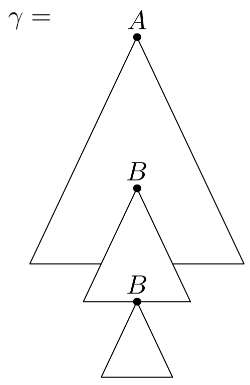

Fig. 3. Tree γ results from adjoining β to α at node ν labelled B.

If α is a tree we denote by $\alpha@a$ the node in α with address a and by $\alpha|a$ the subtree of α rooted by the node $\alpha@a$. The grammar symbol that labels node ν is denoted by Label(ν). Given an elementary tree node ν, Adj(ν) is defined as the set of auxiliary trees that can be adjoined at ν.

Suppose that α is a tree, $a_1, \ldots, a_k$ are distinct addresses in α and $\beta_1, \ldots, \beta_k$ are auxiliary trees such that adjoining β_i to the node at address a_i is defined for $1 \leq i \leq k$. Then we denote by $\alpha[\beta_1 \mapsto a_1, \ldots, \beta_k \mapsto a_k]$ the tree which results from these adjoining operations. If α is an initial tree the tree α' derived by these operations will have no foot node, whereas if α is an auxiliary tree α' will have a foot node.

Definition 1. *We define the set $D(\mathcal{G})$ of derivable trees recursively: $D(\mathcal{G})$ is the smallest set such that*

$I \cup A \subseteq D(\mathcal{G})$ (all elementary trees are derivable)

For all elementary trees α the set $D(\alpha, \mathcal{G})$ of trees $\alpha[\beta_1 \rightarrow l_1, \ldots, \beta_k \rightarrow l_k]$ where $\beta_1, \ldots \beta_k \in D(\mathcal{G})$, is a subset of $D(\mathcal{G})$: $D(\alpha, \mathcal{G}) \subseteq D(\mathcal{G})$.

The valid derivations in $\mathcal{G}$ are the trees in $D(\alpha_S, \mathcal{G})$ where α_S is an initial tree whose root is labelled with the start symbol S.

The set $L(\mathcal{G})$ of terminal strings appearing in the frontier of such trees the string language of $\mathcal{G}$.

3.2 Parsing Deduction System for TAG

Parsers for TAG can be described just as those for CFG, as deduction systems. We present a CYK-like parsing deduction system for tree adjoining grammars. We shall therefor assume that any node in an elementary tree has at most two children. Suppose we want to parse an elementary tree with frontier $w = w_1 \ldots w_n$.

To describe this system, we need the notion of a dotted tree; this can be specified as an elementary tree α, an address a in that tree and a marker to indicate the position of the dot relative to the node at address a. This position can be above or below the node. We will use the notation $\nu^{\bullet}$ and $\nu_{\bullet}$ for dotted trees with dot above and below node ν respectively. The dot in lower position, $\nu_{\bullet}$, specifies that in the derivation of the subtree of α rooted at ν, adjunction must not be involved.

In order to track the portion of the string $w = w_1 \ldots w_n$ covered by the derivation up to the dot position, in general four indices are needed. One pair (i, l) of indices to specify the left edge and the right edge of the parsed portion of the tree, but possibly another pair (j, k) to specify the gap where the foot node occurs in an auxiliary tree.

Hence we consider items $[\nu^{\bullet}, i, j, k, l]$ and $[\nu_{\bullet}, i, j, k, l]$ where ν is a node in an elementary tree α, $0 \leq i \leq l$ are string positions, j and k may be undefined or instantiated to positions $i \leq j \leq k \leq l$ (the latter only if α is an auxiliary tree).

An item $[\alpha@a^\bullet, i, \_, \_, l]$ specifies that there is a tree $\tau \in D(\alpha|a)$, with no foot node, such that the frontier of τ is the string $w_{i+1} \dots w_l$. The position of the dot above specifies that at node $\alpha@a$ an adjunction may have occured to derive τ.

An item $[\alpha@a^\bullet, i, j, k, l]$ specifies that there is a tree $\tau \in D(\alpha|a)$, with foot node, such that the frontier of τ is the string $w_{i+1} \dots w_j A w_{k+1} \dots w_l$, where A is the label of the foot node of α. Again, the position of the dot above specifies that at the node $\alpha@a$ an adjunction may have occured in the derivation of τ.

Items $[\alpha@a_\bullet, i, \_, \_, l]$ and $[\alpha@a_\bullet, i, j, k, l]$ specify similar invariants except that the derivation of τ must not involve adjunction at node $\alpha@a$.

This is illustrated in figure 4.

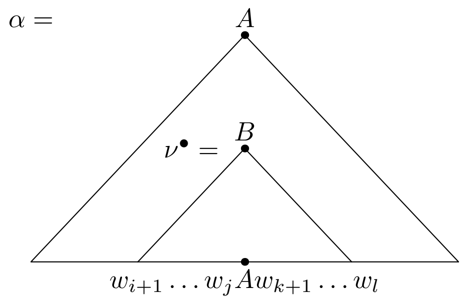

Fig. 4. Tree α illustrates item $[\nu^\bullet, i, j, k, l]$.

The algorithm preserves this invariant while traversing the derived tree from bottom to top, starting with items corresponding to the terminal string symbols themselves which follow from the axioms

$$\frac{}{[\nu^\bullet, i, \_, \_, l]} \quad < \mathrm{Label}(\nu) = w_{i+1} > .$$

Completed subtrees are combined to larger ones and subtrees before adjunction are combined with derived auxiliary trees to form subtrees after adjunction.

The detailed inference rules are depicted in table 5. In order to reduce the number of cases, we define the notation $i \cup j$ for two indices i and j as follows:

$$p \cup q = \begin{cases} p & \text{if } p \text{ is defined} \\ q & \text{otherwise.} \end{cases}$$

In order to prove correctness of the deduction system, we need the following lemma.

Lemma 8. *Let $[\nu^\bullet, i, j, k, l]$ (resp. $[\nu^\bullet, i, j, k, l]$) be a derivable item in the deduction system specified in table 5, then there is an elementary tree α with inner node $\nu^\bullet$ (resp.$\nu_\bullet$) and a derived tree τ in $D(\nu, \mathcal{G})$ whose frontier string is equal to $w_{i+1} \dots w_j \mathrm{Label}(\alpha) w_{k+1} \dots w_l$.*

Table 5. The CYK deductive parsing system for tree adjoining grammars

Item Form:	$[\nu^\bullet, i, j, k, l]$ $[\nu_\bullet, i, j, k, l]$	
Terminal Axiom:	$\dfrac{}{[\nu^\bullet, i, \_, \_, i+1]}$	$<$ Label$(\nu) = w_{i+1} >$
Empty String Axiom:	$\dfrac{}{[\nu^\bullet, i, \_, \_, i]}$	$<$ Label$(\nu) = \epsilon >$
Foot Axiom:	$\dfrac{}{[\beta @ Foot(\beta)_\bullet, j, j, k, k]}$	$< \beta \in A >$
Goals:	$[\alpha @ \epsilon^\bullet, 0, \_, \_, n]$	$< \alpha \in I,$ Label$(\alpha @ \epsilon) = S >$
Inference Rules:		
Complete Unary:	$\dfrac{[\alpha @ a1^\bullet, i, j, k, l]}{[\alpha @ a_\bullet, i, j, k, l]}$	$< \alpha @ a2$ notdefined $>$
Complete Binary:	$\dfrac{[\alpha @ a1^\bullet, i, j, k, l] \ [\alpha @ a2^\bullet, l, j', k', m]}{[\alpha @ a_\bullet, i, j \cup j', k \cup k', m]}$	
No Adjoin:	$\dfrac{[\nu_\bullet, i, j, k, l]}{[\nu^\bullet, i, j, k, l]}$	
Adjoin:	$\dfrac{[\beta @ \epsilon^\bullet, i, p, q, l] \ [\nu_\bullet, p, j, k, q]}{[\nu^\bullet, i, j, k, l]}$	$< \beta \in$ Adj$(\nu) >$

Proof. We proceed by induction on the length of the sequence of inference steps which are performed to derive item $[\nu_\bullet, i, j, k, l]$.

If the item is derived in one step, it must be one of the axiom items. But then the side conditions assert the claim.

If the item is derived by more than one step, an inference rule must have been applied. Several cases are to be considered, one for each rule.

Case 1. The item is of the form $[\alpha @ a_\bullet, i, j, k, m]$ and results from binary Completion. We assume that the items which correspond to the two children of the node $\alpha @ a$ have the form $[\alpha @ a1^\bullet, i, j, k, l]$ and $[\alpha @ a2^\bullet, l, \_, \_, m]$. This means that the indices j and k are defined, hence there is a foot node, and this foot node is in the frontier of the leftmost child. In this case the binary Completion rule reads:

$$\frac{[\alpha @ a1^\bullet, i, j, k, l] \ [\alpha @ a2^\bullet, l, \_, \_, m]}{[\alpha @ a_\bullet, i, j, k, m]}$$

The induction hypothesis can be applied to both antecedents. Hence, the tree τ_1 derived from $\alpha @ a1$ dominates the string $w_{i+1} \dots w_j$Label$(\alpha) w_{k+1} \dots w_l$ and the tree τ_2 derived from $\alpha @ a2$ dominates the string $w_{l+1} \dots w_m$. Then the node $\alpha @ a$ itself dominates the string $w_{i+1} \dots w_j$Label$(\alpha) w_{k+1} \dots w_m$. This is illustrated in figure 5. The other cases involving binary completion are proved analogously. We mention another interesting case: adjunction.

Case 2. The item has the form $[\nu^\bullet, i, j, k, l]$ and is generated by the adjunction rule. Hence, items $[\nu_\bullet, p, j, k, q]$ and $[\beta @ \epsilon^\bullet, i, p, q, l]$, where $\beta \in$ Label(ν), are

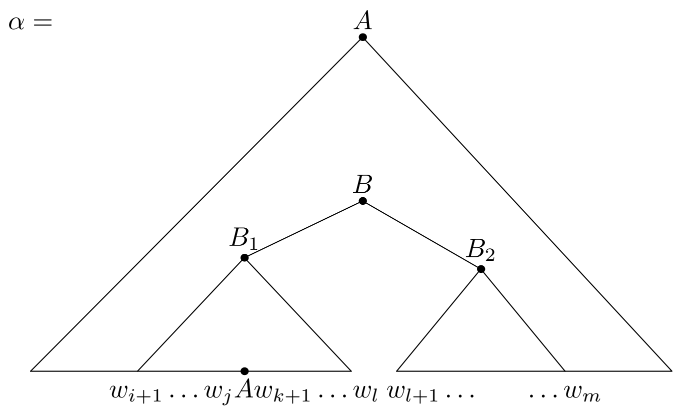
$\alpha =$
A
B
B_1
B_2
$w_{i+1} \ldots w_j A w_{k+1} \ldots w_l \; w_{l+1} \ldots \quad \ldots w_m$

Fig. 5. Tree α illustrates binary completion.

generated in the deduction system and the item in question is inferred from the rule

$$\frac{[\beta@\epsilon^\bullet, i, p, q, l] \; [\nu_\bullet, p, j, k, q]}{[\nu^\bullet, i, j, k, l]}.$$

Again, the induction hypothesis can be applied to both antecedents. Hence, there is a derived tree τ rooted at ν with frontier $w_{p+1} \ldots w_j \mathrm{Label}(\alpha) w_{k+1} \ldots w_q$ and a tree derived from β which dominates $w_{i+1} \ldots w_p \mathrm{Label}(\beta) w_{q+1} \ldots w_l$. Now we need only to adjoin β to α at node ν to obtain a tree with frontier string $w_{i+1} \ldots w_j \mathrm{Label}(\alpha) w_{k+1} \ldots w_l$. The dot changes from the lower to the upper position, since adjunction has been operated. This proves the claim.

The proof of the remaining cases is omitted since the reasonings are quite similar. An analogous result can be shown for items without foot node. Correctness of the deduction system can be shown by application of lemma 8 to the goal item.

Corollary 8. *If the goal item $[\alpha@\epsilon^\bullet, 0, \_, \_, n]$, where $\alpha \in I$, $\mathrm{Label}(\alpha@\epsilon) = S$, can be derived in the deduction system, then the string $w_1 \ldots w_n$ can be derived in the TAG $\mathcal{G}$.*

For completeness we need to prove the following lemma.

Lemma 9. *Suppose that an elementary tree α with inner node $\nu^\bullet$ (resp.$\nu_\bullet$) and derived tree tree τ in $D(\nu, \mathcal{G})$ with frontier string $w_{i+1} \ldots w_j \mathrm{Label}(\alpha) w_{k+1} \ldots w_l$ (resp. $w_{i+1} \ldots \ldots w_l$) can be derived in the TAG. Then the item $[\nu^\bullet, i, j, k, l]$ (resp. $[\nu^\bullet, i, j, k, l]$) can derived in the above specified deduction system (Note that j, k may not be defined).*

Proof. Let r be the length of a shortest derivation of the specified tree. The proof proceeds by induction on r.

If r is one, the dotted tree must elementary. Let us explain the case where it is an initial tree. Then starting with the terminal axioms $[\nu^\bullet, p, \_, \_, p+1]$,

$i \leq p \leq l-1$, and repeated application of binary completion we can infer the item $[\alpha_{\bullet}, i, \_, \_, l]$. The other cases are treated analogously.

Suppose $r > 1$; among the various cases to consider, we specify just one, since they are all treated similarly.

Suppose for example that the node $\nu = \alpha@a$ and has only one child $\alpha@a1$. The tree rooted at this child can be derived in at most $r-1$ steps. Hence we can conclude by induction that the corresponding item $[\alpha@a1^{\bullet}, i, j, k, l]$ can be derived in the deduction system. Now apply unitary completion to infer the item $[\alpha@a1^{\bullet}, i, j, k, l]$.

Corollary 9. *Suppose that the string $w = w_1 \dots w_n$ can be derived in the TAG. Then the goal item $[\alpha^{\bullet}, 0, \_, \_, n]$ can be derived in the deduction system.*

Proof. By hypothesis there is an initial tree with root labelled S whose frontier is the string $w_1 \dots w_n$. Application of lemma 9 yields the claim.

4 Control

In this section we describe a deduction procedure to operate over the inference rules of any parsing deduction system. It uses a chart. Items should be added to the chart as they are proved. However, every new item may itself generate new consequences. The issue as to when these consequences should be computed is subtil. A standard solution is to keep a separate agenda of items that have been proved but whose consequences have not been proved. When an item is removed from the agenda and added to the chart, its consequences are computed and themselves added to the agenda for later consideration.

Thus the general form of a agenda-driven, chart-based deduction procedure is as follows:

1. Initialize the chart to the empty set and the agenda to the set of axioms of the deduction system.
2. Repeat the following steps until the agenda is exhausted:
   (a) Select an item from the agenda, called the trigger item, and remove it.
   (b) Add the trigger item to the chart, if necessary.
   (c) If the trigger item was added to the chart, generate all items that are new immediate consequences of the trigger item together with all the items in the chart, and add these generated items to the agenda.
3. If a goal item is in the chart, the goal is proved and the string is recognized, otherwise it is not.

We show the correctness of this procedure.

Proposition 1. *Suppose that in the above described procedure the agenda has been initialized with items $A_1, \dots A_k$ and item I has been placed in the chart, then $A_1, \dots, A_k \vdash I$.*

Proof. Since every item in the chart must have been in the agenda, and been placed in the chart by step ($2b$), it is sufficient to show that $A_1, \ldots, A_k \vdash I$ for any I in the agenda. We show this by induction on the stage $\sharp(I)$ of I. This is the number of the iteration of step (2) at which I has been added to the agenda, or 0 if I has been placed in the agenda at step (1).

If $\sharp(I) = 0$, I must be an axiom. Thus the trivial derivation consisting of I alone is a derivation of I from $A_1, \ldots, A_k$.

Assume that $\sharp(I) = n$ and the claim is true for any item J with $\sharp(J) < n$. Then I must have been added to the agenda by step ($2c$). Thus there are items $J_1, \ldots J_m$ in the chart and a rule instance such that

$$\frac{J_1 \ldots J_m}{I}$$

and the side conditions on $J_1, \ldots J_m, I$ are satisfied. Since $J_1, \ldots J_m$ are in the chart, they must have been added to the agenda at the latest at the beginning of iteration n of step (2), that is, $\sharp(J_i) < n$ for each $1 \leq i \leq m$. By the induction hypothesis each J_i must have a derivation Δ_i from $A_1, \ldots, A_k$. But then the concatenation of $\Delta_1, \ldots, \Delta_m$ followed by I is a derivation of I from $A_1, \ldots, A_k$.

We show the completeness of this procedure.

Proposition 2. *Suppose that $A_1, \ldots, A_k \vdash I$ in the parsing deduction system. Then item I is in the chart at step* (3).

Proof. We show that item I is eventually added to the chart, if we assume some kind of "fairness" for the agenda. That is we assume that items with smaller stage are removed from the agenda by step ($2a$) before items with greater stage.

We show completeness by induction on the length of any derivation $D_1, \ldots, D_n$ of I from $A_1, \ldots, A_k$.

If $n = 1$, we have $D_1 = I$ and I is an axiom A_i for some i. I will thus be placed in the agenda at step (1) and $\sharp(I) = 0$. By the fairness assumption I will be removed from the agenda after at most k iterations of step (2). When this is done, I will be added to the chart or the chart already contains the same item.

Let $n \geq 1$ and assume the claim for derivations of length less than n. Consider a derivation $D_1, \ldots, D_n = I$ of I from $A_1, \ldots, A_k$. Either I is an axiom, in which case we just have shown the claim, or there are indices $i_1, \ldots, i_m < n$ such that there is an inference rule

$$\frac{D_{i_1} \ldots D_{i_m}}{I} \quad \langle \text{side conditions} \rangle$$

with side conditions satisfied. By definition of derivation, each prefix $D_1, \ldots, D_{i_j}$, ($1 \leq j \leq m$), of $D_1, \ldots, D_n$ is a derivation of D_{i_j} from $A_1, \ldots, A_k$. By induction hypothesis, all items D_{i_j} are in the chart. Note I_p the item among the D_{i_j}' that was added latest to the chart. Then it will be the trigger item for the application of the above rule. Thus I will be added to the agenda. Since step ($2c$) can only add a finite number of items to the agenda, item I will eventually be considered at steps ($2a$) and ($2b$) and added to the chart, if not already there.

References

1. Shieber, S.M., Schabes, Y. Pereira, F.C.N.: Principles and Implementation of Deductive Parsing. *J*ournal of Logic Programming, 23(1–2):3–36, July-August 1995.
2. Vijay-Shanker, K., Joshi, A.K. Some Computational Properties of Tree Adjoining Grammars. *P*roc. of ACL'85, pp. 82–93, Chicago, USA.
3. Wegener, I.: Theoretische Informatik - eine algorithmenorientierte Einführung, 2. Auflage, Leitfäden der Informatik. B.G. Teubner Stuttgart-Leipzig 1999 ISBN 3-519-12123-9