Sequentially Indexed Grammars

Jan van Eijck
CWI and ILLC, Amsterdam, Uil-OTS, Utrecht

Draft, January 2005

Abstract

This paper defines the grammar class of sequentially indexed grammars. Sequentially indexed grammars are the result of a change in the index stack handling mechanism of indexed grammars [Aho68, Aho69]. Sequentially indexed grammars are different from linear indexed grammars [Gaz88]. Like indexed languages, sequentially indexed languages are a fully abstract language class. Unlike indexed languages, sequentially indexed languages allow polynomial parsing algorithms. We give a polynomial algorithm for parsing with sequentially indexed gramamrs that is an extension of the Earley algorithm for parsing with context free grammars.

Keywords: Deductive parsing, context free grammars, indexed languages, nested stack automata, Earley parsing algorithm.

1 Introduction

Indexed grammars [Aho68] result from adding a stack index mechanism to context free grammars.

Definition 1 *Indexed grammars are quadruples $G = (N, T, P, S)$, where N is a finite nonterminal alphabet, T is a finite terminal alphabet, $N \cap T = \emptyset$, P is a finite set of productions of the form (X, α) with $X \in N \cup N^\circ$, $\alpha \in (N \cup N^\circ \cup T)^*$, where N° is the set of all X_Y with $X, Y \in N$, and $S \in N$ is the start symbol.*

A production (X, α) of an indexed grammar is written as $X \to \alpha$.

In what follows, we use [] for the empty stack, and $X : \zeta$ for the stack with top element X. A stack $X_1 : X_2 : \cdots : X_n : []$ will be written in the usual list notation, as $[X_1, \ldots, X_n]$, or as $X_1, \ldots, X_n$.

Let $G = (N, T, P, S)$ be an indexed grammar. A pair $(X, [X_1, \ldots, X_n])$, with $X, X_1, \ldots, X_n \in N$ is called an indexed nonterminal. Indexed nonterminals are written as $X_{X_1 \cdots X_n}$. Let $N^\bullet$ be the set of all $X_{X_1 \cdots X_n}$, with $X, X_1, \ldots, X_n \in N$. Then a sentential form for G is a string α in $(N^\bullet \cup T)^*$.

To define the one step derivation relation, we need a preliminary definition:

Definition 2 *If $\delta \in (N \cup N^\circ \cup T)^*$ and $\gamma \in N^*$, then δ_γ is given by the following recursion:*

$$\begin{aligned} \epsilon_\zeta &= \epsilon \\ (w:\delta)_\zeta &= w:\delta_\zeta \text{ if } w \in T, \\ (Y:\delta)_\zeta &= Y_\zeta:\delta_\zeta \text{ if } Y \in N, \\ (Y_Z:\delta)_\zeta &= Y_{(Z:\zeta)}:\delta_\zeta \text{ if } Y_Z \in N^\circ, \end{aligned}$$

Note that as a special case of this, we have that $(Y_Z)_\theta = Y_{Z:\theta}$.

Using the definition of δ_γ, we can define one step derivations:

Definition 3 *Let α, β be sentential forms for indexed grammar G. Then $\alpha \overset{G}{\Rightarrow} \beta$ iff*

1. *$\alpha = \gamma_1 X_\zeta \gamma_2$, $X \to \delta$ is a production of the grammar, and $\beta = \gamma_1 \delta_\zeta \gamma_2$, or,*
2. *$\alpha = \gamma_1 X_{(Y:\zeta)} \gamma_2$, $X_Y \to \delta$ is a production of the grammar, and $\beta = \gamma_1 \delta_\zeta \gamma_2$*

In terms of this, $\alpha \overset{G}{\Rightarrow}^* \beta$ is defined in the usual way. This definition is equivalent to the definition in [Aho68].

2 Example Grammars, Example Derivations

An example of an indexed grammar:

$$\begin{aligned} S &\to A_X \\ A &\to aA_Y \mid B \\ B_Y &\to bBc \\ B_X &\to \epsilon \end{aligned}$$

This grammar generates the non-CF language $\{a^n b^n c^n \mid n \in \mathbb{N}\}$.

$$\begin{aligned} S &\Rightarrow A_X \\ &\Rightarrow aA_{YX} \\ &\Rightarrow aaA_{YYX} \\ &\Rightarrow aaB_{YYX} \\ &\Rightarrow aabB_{YX}c \\ &\Rightarrow aabbB_X cc \\ &\Rightarrow aabbcc \end{aligned}$$

Here is another example.

$$\begin{aligned} S &\rightarrow A_X \\ A &\rightarrow aA_Y \mid bA_Z \mid B \\ B_Y &\rightarrow Ba \\ B_Z &\rightarrow Bb \\ B_X &\rightarrow \epsilon \end{aligned}$$

This grammar generates the non-CF language $\{ww \mid w \in \{a,b\}^*\}$.

$$\begin{aligned} S &\Rightarrow A_X \\ &\Rightarrow aA_{YX} \\ &\Rightarrow aaA_{YYX} \\ &\Rightarrow aabbA_{ZYYX} \\ &\Rightarrow aabB_{ZYYX} \\ &\Rightarrow aabB_{YYX}b \\ &\Rightarrow aabB_{YX}ab \\ &\Rightarrow aabB_{X}aab \\ &\Rightarrow aabB_{X}ab \\ &\Rightarrow aabaab \end{aligned}$$

Note that in this and the previous example, the index X serves the purpose of triggering the erasing rule for B. Suppose in the last grammar we had the production $B \rightarrow \epsilon$ instead of $B_X \rightarrow \epsilon$. Then we could have derived:

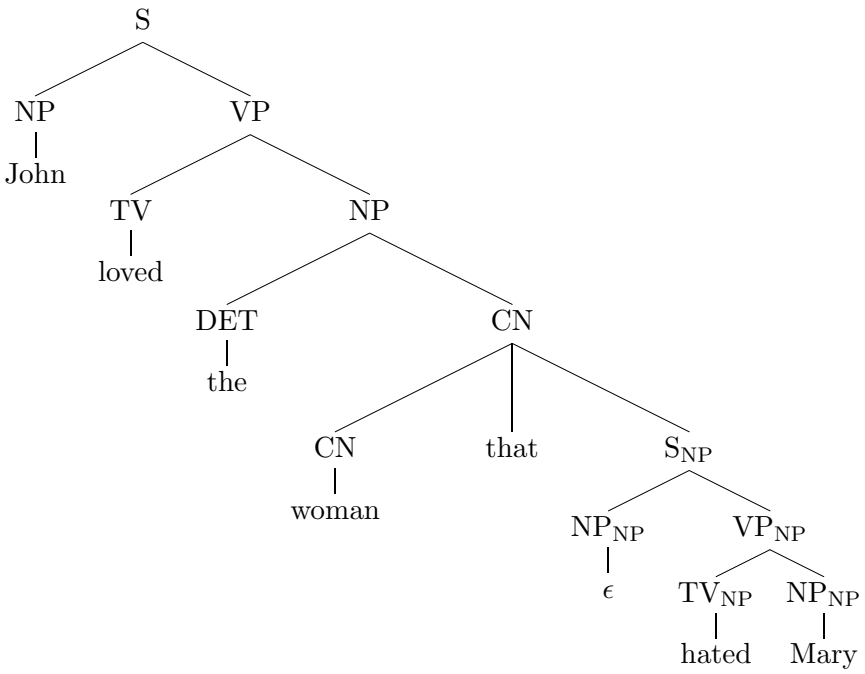
$$\begin{aligned} S &\Rightarrow A_X \\ &\Rightarrow aA_{YX} \\ &\Rightarrow aaA_{YYX} \\ &\Rightarrow aabbA_{ZYYX} \\ &\Rightarrow aabB_{ZYYX} \\ &\Rightarrow aabB_{YYX}b \\ &\Rightarrow aabb \end{aligned}$$

This is because $B_{YYX} \Rightarrow \epsilon$ is a one-step derivation with rule $B \rightarrow \epsilon$, for the index stack gets copied on all the nonterminals in the righthand side of a rule and disappears on all terminals.

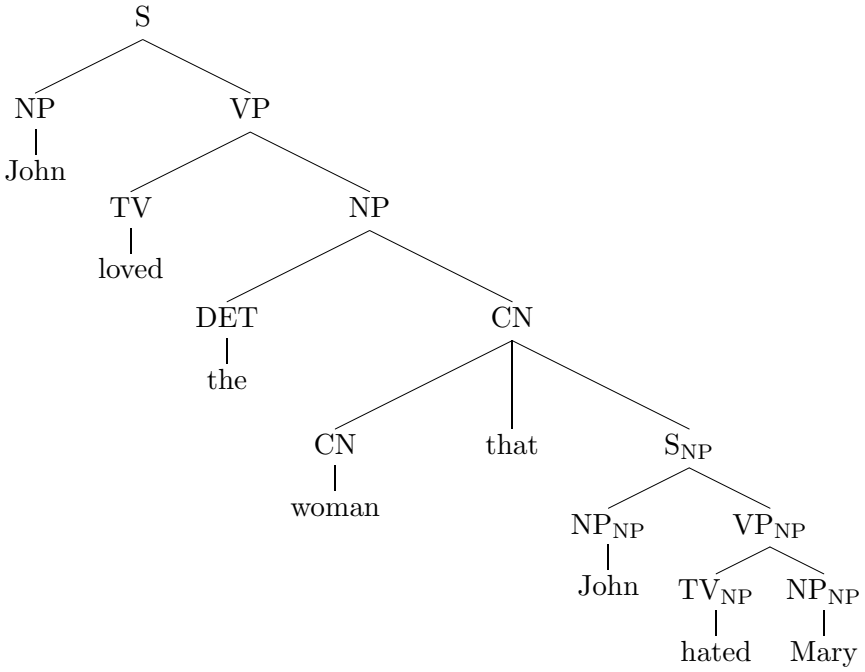
A grammar for a toy fragment of natural language:

$$
\begin{aligned}
\text{S} &\rightarrow \text{NP VP} \\
\text{VP} &\rightarrow \text{TV NP} \mid \text{talked} \mid \text{smiled} \\
\text{NP} &\rightarrow \text{DET CN} \mid \text{John} \mid \text{Mary} \\
\text{TV} &\rightarrow \text{loved} \mid \text{hated} \\
\text{DET} &\rightarrow \text{the} \mid \text{some} \\
\text{CN} &\rightarrow \text{man} \mid \text{woman} \mid \text{CN that } \text{S}_{\text{NP}} \\
\text{NP}_{\text{NP}} &\rightarrow \epsilon
\end{aligned}
$$

This grammar allows us to illustrate why indexed grammars are not quite what we want for natural language analysis. The grammar uses the index mechanism to define a relative sentence as a sentence with an NP gap in it. When parsing with this grammar using the stack mechanism of indexed grammars, this gap is propagated *everywhere* in the relative sentence, due to the stack copying mechanism in the definition of indexed grammar derivations:



Thus, we will also get:



3 Sequentially Indexed Grammars

Sequentially indexed grammars use indices that get pushed to an *arbitrary* nonterminal in the righthand side of a production. Sequentially indexed grammars look just like indexed grammars, but the definition of *derivation* is different. The following definition uses list concatenation If ζ is the result of concatenating ζ_1 and ζ_2, we denote this as $\zeta = \zeta_1 +\!\!+ \zeta_2$.

Definition 4 *If $\delta \in (N \cup N^\circ \cup T)^*$ and $\gamma \in N^*$, then $(\delta)_\zeta$ is the* **subset** *of $(N^\bullet \cup T)^*$ defined recursively as:*

$$\begin{aligned}
(\epsilon)_{[]} &= \{\epsilon\} \\
(\epsilon)_\zeta &= \emptyset \text{ if } \zeta \neq [], \\
(w:\delta)_\zeta &= \{w:\delta' \mid \delta' \in \delta_\zeta\} \text{ if } w \in T, \\
(C:\delta)_\zeta &= \{C_{\zeta_1}:\delta' \mid \zeta = \zeta_1 +\!\!+ \zeta_2, \delta' \in \delta_{\zeta_2}\} \text{ if } C \in N \\
(C_Y:\delta)_\zeta &= \{C_{Y:\zeta_1}:\delta' \mid \zeta = \zeta_1 +\!\!+ \zeta_2, \delta' \in \delta_{\zeta_2}\} \text{ if } C_Y \in N^\circ.
\end{aligned}$$

The relation of one-step derivation is defined in terms of $(\delta)_\zeta$, as follows:

Definition 5 *Let α, β be sentential forms for indexed grammar G. Then $\alpha \overset{G}{\Rightarrow} \beta$ iff*

1. *$\alpha = \gamma_1 B_\zeta \gamma_2$, $B \to \delta$ is a production of the grammar, and $\beta = \gamma_1 \delta' \gamma_2$, where δ' in $(\delta)_\gamma$, or*
2. *$\alpha = \gamma_1 B_{(Y:\zeta)} \gamma_2$, $B_Y \to \delta$ is a production of the grammar, and $\beta = \gamma_1 \delta' \gamma_2$, where $\delta' \in (\delta)_\zeta$.*

In derivations with simple indexed grammars, stacks are never allowed to disappear, and stacks are never allowed to get duplicated. In particular, a production $B \to \epsilon$ will not allow a one-step derivation like $B_{YYX} \Rightarrow \epsilon$, and a production $B \to CD$ will not allow a one-step derivation like $B_{YYX} \Rightarrow C_{YYX}D_{YYX}$ (but it will allow $B_{YYX} \Rightarrow C_{YYX}D$, $B_{YYX} \Rightarrow C_{YY}D_X$, $B_{YYX} \Rightarrow C_Y D_{YX}$, and $B_{YYX} \Rightarrow CD_{YYX}$).

A production like $B_X \to \epsilon$ can lead to a one-step derivation $B_X \Rightarrow \epsilon$. This effectively treats X as a 'trace'.

Sequentially indexed grammars are different from an earlier proposal for a restricted form of indexed grammars, in [Gaz88]. Gazdar proposed to use index lists that get copied to a single nonterminal in the righthand sides of productions, but in such a way that this heir-nonterminal has to be indicated in the rule.

4 Simplification of Earlier Example Grammars

Because of the fact that stacks are not allowed to disappear, we can now carry through some simplifications of the earlier example grammars for $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ and $\{ww \mid w \in \{a,b\}^*\}$, as follows.

Here is the earlier example grammar for $\{a^n b^n c^n \mid n \in \mathbb{N}\}$.

$$\begin{array}{rcl} S & \to & A_X \\ A & \to & aA_Y \mid B \\ B_Y & \to & bBc \\ B_X & \to & \epsilon \end{array}$$

This grammar simplifies to:

$$\begin{array}{rcl} S & \to & aS_X \mid A \\ A_X & \to & bAc \\ A & \to & \epsilon \end{array}$$

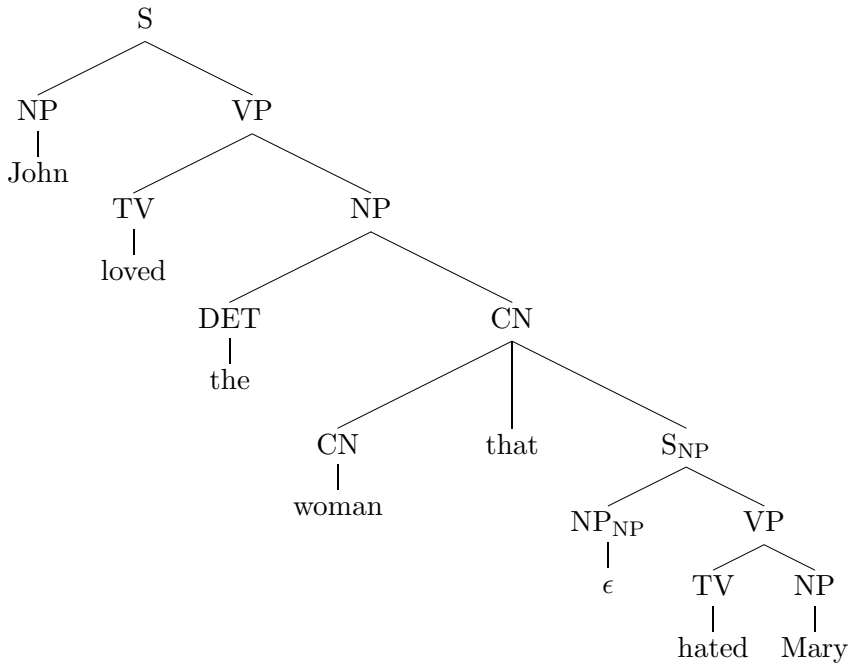
The example grammar for $\{ww \mid w \in \{a,b\}^*\}$:

$$\begin{array}{rcl} S & \to & A_X \\ A & \to & aA_Y \mid bA_Z \mid B \\ B_Y & \to & Ba \\ B_Z & \to & Bb \\ B_X & \to & \epsilon \end{array}$$

This grammar simplifies to:

$$\begin{aligned} S &\rightarrow aS_X \mid bS_Y \mid A \\ A_X &\rightarrow Aa \\ A_Y &\rightarrow Ab \\ A &\rightarrow \epsilon \end{aligned}$$

The example natural language grammar will behave as intended when treated as a sequentially indexed grammar. We get the following derivation tree:



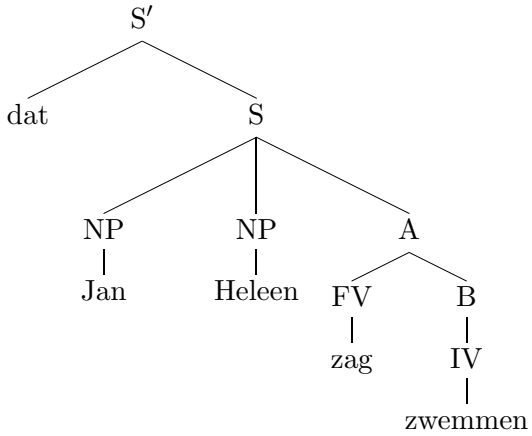
5 Dutch Cross Serial Dependencies

A sequentially indexed grammar for Dutch cross serial dependencies (see, e.g., [BKPZ82]):

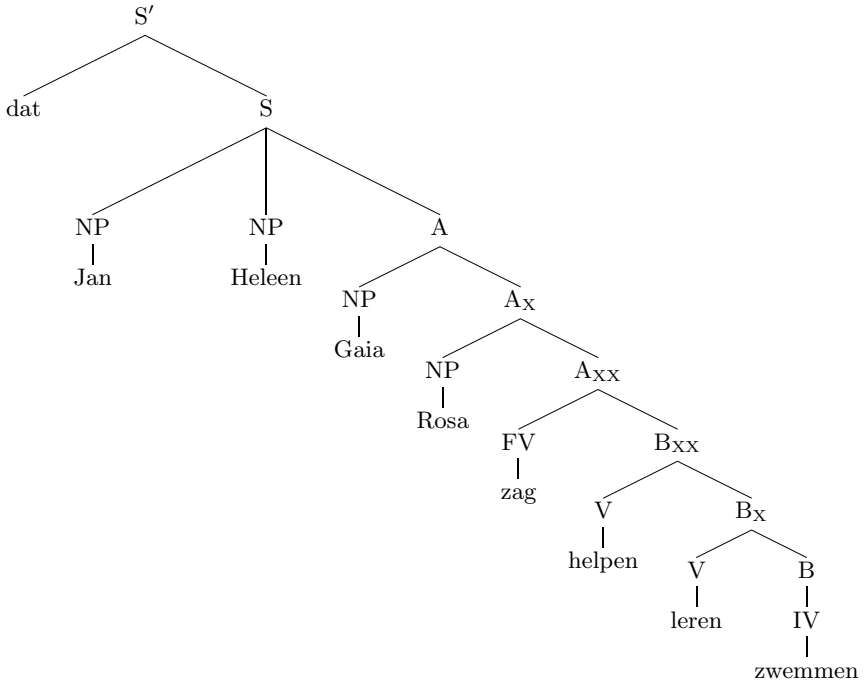
$$
\begin{aligned}
S' &\rightarrow \text{dat S} \\
S &\rightarrow \text{NP NP A} \\
NP &\rightarrow \text{Jan} \mid \text{Heleen} \mid \text{Gaia} \mid \text{Rosa} \\
A &\rightarrow \text{NP } A_X \\
A &\rightarrow \text{FV B} \\
FV &\rightarrow \text{zag} \mid \text{hoorde} \mid \text{liet} \mid \text{leerde} \mid \text{hielp} \\
B_X &\rightarrow \text{V B} \\
B &\rightarrow \text{IV} \\
V &\rightarrow \text{zien} \mid \text{horen} \mid \text{laten} \mid \text{leren} \mid \text{helpen} \\
IV &\rightarrow \text{zwemmen} \mid \text{lopen} \mid \text{rennen}
\end{aligned}
$$

This gives the following parses for the Dutch subordinate clauses in (1) and (2).

1 *(dat) Jan Heleen zag zwemmen.*



2 *(dat) Jan Heleen Gaia Rosa zag helpen leren zwemmen.*



6 Deductive Earley-style Parsing with CF Grammars

Earley items for context free parsing [Ear70] are of the form

$$i, A \rightarrow \alpha \bullet \beta, j.$$

$A \rightarrow \alpha\beta$ is a grammar production. $\bullet$ indicates the part of its righthand side that was recognized so far, i points to the position in the input where the rule $A \rightarrow \alpha\beta$ was invoked (the position of the parent node), and j points to the position in the input that recognition has reached.

If an input string of terminals has length n, we denote that string as $w_0, \ldots, w_{n-1}$. After recognition of j terminals (i.e., after recognition of $w_0, \ldots, w_{j-1}$), the recognition pointer has value j.

There is one axiom. It has the form

$$0, S' \rightarrow \bullet S, 0$$

where S is the start symbol and S' is a new start symbol.

There is one goal. It has the form

$$0, S' \rightarrow S \bullet, n$$

where S, S' are as in the axiom, and n is the length of the input. This goal item says that the rule $S' \rightarrow S$ was invoked at the start of the input, n input symbols were recognized (i.e., all of the input was recognized), and the righthand side of rule $S' \rightarrow S$ was completed.

Scanning The scanning rule for Earley parsing with CF grammars is the rule that shifts the bullet across a terminal:

$$\frac{i, A \to \alpha \bullet w_j \beta, j}{i, A \to \alpha w_j \bullet \beta, j+1}$$

Prediction The prediction rule for Earley parsing is the rule that initializes a new rule $B \to \gamma$ on the basis of a premisse indicating that B is expected at the current point in the input:

$$\frac{i, A \to \alpha \bullet B\beta, j}{j, B \to \bullet\gamma, j} B \to \gamma$$

Completion The completion rule for Earley parsing is the rule that shifts the bullet across a non-terminal. It has two premisses:

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

Setting up the Deduction Engine Follow the set-up of [SSP95].

A chart plus agenda is a pair of item lists.

Use the agenda for those items that have been proved, but whose direct consequences have not yet been derived,

Use the chart for the proved items the direct consequences of which have also been computed.

The Parsing Process

- Initialisation: empty chart, and list of all axioms on the agenda.
- Next, successively tackle each item I on the agenda:
  - add the direct consequences of I to the agenda.
  - move I from the agenda to the chart (as the direct consequences of I have been computed).
- Finally, check whether a goal item has been found, and return the list of goal items.

Adaptation for Parsing with Indexed Grammars

Adapting this for parsing with indexed grammars, we extend the nonterminals with stacks. The item format becomes

$$i, A_\eta \to \alpha \bullet \beta, j,$$

with $\eta \in N^*$, $\alpha, \beta \in (N^\bullet \cup T)^*$. The item $i, A_\eta \to \alpha \bullet \beta, j$ indicates that at the point of invocation of rule $A \to \alpha\beta$, the index stack has shape η, or, in case $\eta = (X : \eta')$, that at the point of invocation of rule $A_X \to \alpha\beta$, the index stack has shape η'.

Again, there is one axiom. It takes the shape

$$0, S'_{[]} \rightarrow \bullet S_{[]}, 0,$$

indicating that the stack component is empty at the beginning of the parse.

There is a single goal, The goal shape becomes:

$$0, S'_{[]} \rightarrow S_{[]} \bullet, n.$$

Again, there a three kinds of consequence rules, for scanning, prediction and completion.

Scanning Scanning does not change the index stacks, so we get:

$$\frac{i, A_\eta \rightarrow \alpha \bullet w_j \beta, j}{i, A_\eta \rightarrow \alpha w_j \bullet \beta, j+1}$$

Prediction In the case of Earley-style parsing with indexed grammars prediction splits into two rules. The first variant on the prediction rule has as side condition a grammar production with lhs in N (the lhs is a nonterminal without index). This rule leaves the index stack unaffected.

$$\frac{i, A_\eta \rightarrow \alpha \bullet B_\zeta \beta, j}{j, B_\zeta \rightarrow \bullet \gamma_\zeta, j} B \rightarrow \gamma$$

The second variant of the prediction rule pops the index stack. This variant has as side condition a grammar production with lhs in N° (the lhs is a nonterminal with as its index the nonterminal at the top of the index stack).

$$\frac{i, A \rightarrow \alpha \bullet B_{X:\zeta} \beta, j}{j, B_{X:\zeta} \rightarrow \bullet \gamma_\zeta, j} B_X \rightarrow \gamma$$

Completion For the case of Earley-style parsing with indexed grammars, this takes the following shape:

$$\frac{i, A_\eta \rightarrow \alpha \bullet B_\zeta \beta, k \quad k, B_\zeta \rightarrow \gamma \bullet, j}{i, A_\eta \rightarrow \alpha B_\zeta \bullet \beta, j}$$

Simple-minded parsing with sequentially indexed grammars A simple-minded way of adapting the above algorithm to the case of parsing with sequentially indexed grammars is as follows.

Nothing changes, except for the definition of the prediction rules:

$$\frac{i, A_\eta \rightarrow \alpha \bullet B_\zeta \beta, j}{j, B_\zeta \rightarrow \bullet \delta, j} B \rightarrow \gamma$$

where $\delta \in (\gamma)_\zeta$, with $(\gamma)_\zeta$ as in definition 4.

$$\frac{i, A_\eta \rightarrow \alpha \bullet B_{X:\zeta} \beta, j}{j, B_{X:\zeta} \rightarrow \bullet \delta, j} B_X \rightarrow \gamma$$

again with $\delta \in (\gamma)_\zeta$.

This approach works, but it has the disadvantage that it yields an algorithm that is, like the algorithm for Earley parsing with indexed grammars, exponential in the size of the list of input tokens.

Even if we limit the size of the index stacks to the length of the remaining input, a grammar with k index symbols has k^j index stacks of length j. The number of relevant items to consider if the input has length n is $c \cdot k^n$.

7 Polynomial Parsing with Sequentially Indexed Grammars

Earley items For polynomial parsing of sequentially indexed languages, Earley items will be adapted by using three extra components:

1. A stack of the indices at the point where the rule was invoked,
2. A stack of indices for the first nonterminal to the right of $\bullet$,
3. A stack of indices for the tail of the nonterminal list to the right of $\bullet$.

We will use Greek letters η, ζ, θ for index stacks,

The item format now becomes:

$$i, \theta, A \rightarrow \alpha \bullet \beta, \eta, \zeta, j$$

where θ, η, ζ are stacks of indices (nonterminals). The item indicates the following:

- grammar rule $A \rightarrow \alpha\beta$ was invoked at point i,
- at the point of invocation, the top node A has associated stack θ,
- at point j, part α of the righthand side of the rule has been successfully recognized,
- η is the stack for the first nonterminal among β (if β has no nonterminals, then η is empty),
- ζ is the stack for the remainder of the nonterminals in β (if β has less than two nonterminals, then ζ is empty).

Axiom In the case of Earley parsing with CF grammars, there is one axiom. It has the form $0, S' \rightarrow \bullet S, 0$, where S is the start symbol of the grammar and S' is a new start symbol. Adapting this to the case of parsing with sequentially indexed grammars, the axiom takes the shape

$$0, [], S' \rightarrow \bullet S, [], [], 0,$$

indicating that at the beginning of the parse, there is one pending nonterminal, and all stack components are empty.

Goal In the case of Earley parsing with CF grammars, there is one goal. It has the form $0, S' \rightarrow S\bullet, n$, where S is the start symbol of the grammar, S' is the new start symbol used in the axiom, and n is the length of the input. For the case of Earley style parsing with indexed grammars, we also require that the index stack components are empty at the end of the parse, so the goal shape becomes:

$$0, [], S' \rightarrow S\bullet, [], [], n.$$

Consequences As in the case of Earley parsing with CF grammars, there are three kinds of consequences, for scanning, prediction and completion.

Scanning The scanning rule for Earley parsing with CF grammars is the rule that shifts the bullet across a terminal. For parsing sequentially indexed languages, three index stack components are added to this. Scanning does not change the index stacks θ, η, ζ.

$$\frac{i, \theta, A \rightarrow \alpha \bullet w\beta, \eta, \zeta, j}{i, \theta, A \rightarrow \alpha w \bullet \beta, \eta, \zeta, j+1}$$

Prediction The prediction rule for Earley parsing is the rule that initializes a new rule $B \rightarrow \gamma$ on the basis of a premisse indicating that B is expected at the current point in the input. In the case of Earley-style parsing with sequentially indexed grammars this splits into four rules. The rules split the first index stack. For this we need some terminology.

If γ is a list of grammar symbols and η, η', η'' are index stacks, then $c(\gamma)$ is the number of nonterminals in γ, and $C(\eta, \eta', \eta'', \gamma)$ is the following constraint:

$$\eta = \eta' +\!\!+ \eta'' \wedge (c(\gamma) = 0 \Rightarrow \eta = []) \wedge (c(\gamma) = 1 \Rightarrow \eta'' = []).$$

The first prediction rule covers the case of an expected nonterminal B matched against a rule with head B. The rule distributes the appropriate stack over the new item, in accordance with the constraint imposed by the number of nonterminals in the righthand side of the grammar rule used in the prediction.

$$\frac{i, \theta, A \rightarrow \alpha \bullet B\beta, \eta, \zeta, j}{j, \eta, B \rightarrow \bullet\gamma, \eta', \eta'', j} B \rightarrow \gamma, C(\eta, \eta', \eta'', \gamma)$$

The second rule covers the case of an expected nonterminal B matched against a rule with head B_X. This rule pops the index stack associated with B.

$$\frac{i, \theta, A \rightarrow \alpha \bullet B\beta, (X : \eta), \zeta, j}{j, \eta, B_X \rightarrow \bullet\gamma, \eta', \eta'', j} B_X \rightarrow \gamma, C(\eta, \eta', \eta'', \gamma)$$

The third rule covers the case of an expected nonterminal B_Y matched against a rule $B \rightarrow \gamma$:

$$\frac{i, \theta, A \rightarrow \alpha \bullet B_Y\beta, \eta, \zeta, j}{j, (Y : \eta), B \rightarrow \bullet\gamma, \eta', \eta'', j} B \rightarrow \gamma, C(Y : \eta, \eta', \eta'', \gamma), n - j > |\eta|$$

Note the side condition on the rule. The side condition prevents unlimited growth of the stack. This is needed to prevent a rule like $A \to A_Y$ from causing an unbounded number of pushes.

The fourth rule covers the case of an expected nonterminal B_Y matched against a rule $B_Y \to \gamma$:

$$\frac{i, \theta, A \to \alpha \bullet B_Y \beta, \eta, \zeta, j}{j, \eta, B_Y \to \bullet \gamma, \eta', \eta'', j} B_Y \to \gamma, C(\eta, \eta', \eta'', \gamma)$$

Completion The completion rule for Earley parsing is the rule that shifts the bullet across a non-terminal. For the case of Earley-style parsing with sequentially indexed grammars, this splits into four rules, as follows.

The first rule checks that the lefthand tail index stack of the first premisse matches the head index stack of the second premisse, for the case of a match of expected symbol B against completed rule $B \to \gamma$.

$$\frac{i, \theta, A \to \alpha \bullet B \beta, \eta, \zeta, k \quad k, \eta, B \to \gamma \bullet, [], [], j}{i, \theta, A \to \alpha B \bullet \beta, \zeta', \zeta'', j} C(\zeta, \zeta', \zeta'', \beta)$$

The second rule covers the case of a match of expected symbol B against completed rule $B_Y \to \gamma$.

$$\frac{i, \theta, A \to \alpha \bullet B \beta, (Y : \eta), \zeta, k \quad k, \eta, B_Y \to \gamma \bullet, [], [], j}{i, \theta, A \to \alpha B \bullet \beta, \zeta', \zeta'', j} C(\zeta, \zeta', \zeta'', \beta)$$

The third rule covers the case of a match of expected symbol B_Y against completed rule $B \to \gamma$.

$$\frac{i, \theta, A \to \alpha \bullet B_Y \beta, \eta, \zeta, k \quad k, (Y : \eta), B \to \gamma \bullet, [], [], j}{i, \theta, A \to \alpha B_Y \bullet \beta, \zeta', \zeta'', j} C(\zeta, \zeta', \zeta'', \beta)$$

The fourth rule covers the case of a match of expected symbol B_Y against completed rule $B_Y \to \gamma$.

$$\frac{i, \theta, A \to \alpha \bullet B_Y \beta, \eta, \zeta, k \quad k, \eta, B_Y \to \gamma \bullet, [], [], j}{i, \theta, A \to \alpha B_Y \bullet \beta, \zeta', \zeta'', j} C(\zeta, \zeta', \zeta'', \beta)$$

This completes the description of the inference rules. In terms of these, we define item derivability.

Definition 6 (Derivability) *Let G be a sequentially indexed grammar, and $w_0, \ldots, w_{n-1}$ a list of terminal symbols. Then*

$$\vdash^G_{w_0, \ldots, w_{n-1}} i, \theta, A \to \alpha \bullet \beta, \eta, \zeta, j$$

expresses that item $i, \theta, A \to \alpha \bullet \beta, \eta, \zeta, j$ is derivable in the parsing calculus from the axiom by means of the inference rules.

8 Soundness and Completeness of the Parsing Calculus

Soundness We have to show that the parsing calculus is sound: if

$$\vdash^{G}_{w_0,\ldots,w_{n-1}} 0, [], S' \to S\bullet, [], [], n,$$

then

$$S \overset{G}{\Rightarrow}^{*} w_0, \ldots, w_{n-1}.$$

For this, we will use Correctness Lemma 8, but first we introduce notation for the right adjoint of $+\!\!+$, as follows:

Definition 7 *If $\eta, \zeta \in N^*$ and ζ is a suffix of η, then η/ζ is the stack θ with $\eta = \theta +\!\!+ \zeta$.*

This definition is used in the formulation of Lemma 8.

Lemma 8 (Correctness) *Let an indexed grammar G and an input sequence $w_0, \ldots, w_{n-1}$ be given. Then*

$$\vdash^{G}_{w_0,\ldots,w_{n-1}} i, \theta, A \to \alpha \bullet \beta, \eta, \zeta, j$$

implies that the following conditions are fulfilled:

1. *$A \to \alpha\beta$ is a production of G,*
2. *there are $\gamma, \delta \in (N^\bullet \cup T)^*$ with*

$$S \overset{G}{\Rightarrow}^{*} \gamma A_\theta \delta \text{ and } \gamma \overset{G}{\Rightarrow}^{*} w_0, \ldots, w_{i-1},$$

3. *θ has $\eta +\!\!+ \zeta$ as a suffix,*
4. *there is an $\alpha' \in (N^\bullet \cup T)^*$ with $\alpha' \in (\alpha)_{\theta/(\eta +\!\!+ \zeta)}$ and*

$$\alpha' \overset{G}{\Rightarrow}^{*} w_i, \ldots, w_{j-1}.$$

Proof. Induction on the number of items which were added to the item database before item $i, \theta, A \to \alpha \bullet \beta, \eta, \zeta, j$ got added.

For the basis, observe that the first items that got in were triggered by the axiom $0, [], S' \to \bullet S, [], 0$, so they must have been the result of a predict action for S. In other words, these items are of the form $0, [], S \to \bullet\gamma, [], [], 0$ or $0, [], S_X \to \bullet\gamma, [], [], 0$. It is clear that in both cases γ, δ, α' can be found with the required properties (namely, $\gamma = \delta = \alpha' = \epsilon$).

Induction step. Suppose that the induction hypothesis holds for all items presently in the store.

Consider a new item that got added as the result of a scan. Then this new item has the form $i, \theta, A \to \alpha w \bullet \beta, \eta, \zeta, j + 1$, and it derives from $i, \theta, A \to \alpha \bullet w\beta, \eta, \zeta, j$. The result follows from the fact that the latter item satisfies the induction hypothesis.

Consider a new item that got added as the result of a prediction. Then there are four cases to check. As an example, we check the first case. The new item has the form

$$j, \eta, B \to \bullet\chi, \eta', \eta'', j,$$

and it derives from an item

$$i, \theta, A \to \alpha \bullet B\beta, \eta, \zeta, j.$$

The latter item satisfies the induction hypothesis, so there are γ, δ with

$$S \stackrel{G}{\Rightarrow}^* \gamma A_\theta \delta \text{ and } \gamma \stackrel{G}{\Rightarrow}^* w_0, \ldots, w_{i-1},$$

and there is an α' with $\alpha' \in (\alpha)_{\theta/(\eta +\!\!+ \zeta)}$, $\alpha' \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{j-1}$.

It follows that $S \stackrel{G}{\Rightarrow}^* \gamma\alpha' B_\eta \beta' \delta$, for some $\beta' \in (\beta)_\zeta$, and $\epsilon_{[]} \stackrel{G}{\Rightarrow}^* \epsilon$, so the property holds.

Consider a new item that got added as the result of a completion step. Then again there are four cases to check. As an example, we check the first case. The new item has the form $i, \theta, A \to \alpha B \bullet \beta, \zeta', \zeta'', j$, and it derives from premisses:

$$i, \theta, A \to \alpha \bullet B\beta, \eta, \zeta, k \quad k, \eta, B \to \chi\bullet, [], [], j.$$

Invoking the induction hypothesis on the first premiss we get γ, δ with

$$S \stackrel{G}{\Rightarrow}^* \gamma A_\theta \delta \text{ and } \gamma \stackrel{G}{\Rightarrow}^* w_0, \ldots, w_{i-1},$$

and $\alpha' \in (\alpha)_{\theta/(\eta +\!\!+ \zeta)}$ with $\alpha' \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{k-1}$.

Invoking the induction hypothesis on the second premiss we get that for some $\chi' \in (\chi)_\eta$, $\chi' \stackrel{G}{\Rightarrow}^* w_k, \ldots, w_{j-1}$.

Together with $B \to \chi$ this gives $B_\eta \stackrel{G}{\Rightarrow}^* w_k, \ldots, w_{j-1}$, and together with $\alpha' \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{k-1}$ we get:

$$\alpha' B_\eta \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{j-1}.$$

It follows that the required property holds. □

Theorem 9 (Soundness) *The parsing calculus is sound.*

Proof. Apply Lemma 8 to the special case of item $0, [], S' \to S\bullet, [], [], n$. □

Completeness We have to show: if $S \stackrel{G}{\Rightarrow}^* w_0, \ldots, w_{n-1}$ then the parsing calculus for G on input $w_0, \ldots, w_{n-1}$ allows us to prove this fact, i.e., the calculus yields:

$$\vdash^G_{w_0,\ldots,w_{n-1}} 0, [], S' \to S\bullet, [], [], n.$$

The completeness result follows from Adequacy Lemma 10.

Lemma 10 (Adequacy) *Let an indexed grammar G and an input sequence $w_0, \ldots, w_{n-1}$ be given. Suppose that the following conditions are fulfilled:*

1. *$A \to \alpha\beta$ is a production of G,*
2. *there are $\gamma, \delta \in (N^\bullet \cup T)^*$ with*

$$S \stackrel{G}{\Rightarrow}^* \gamma A_\theta \delta \text{ and } \gamma \stackrel{G}{\Rightarrow}^* w_0, \ldots, w_{i-1},$$

3. *θ has $\eta +\!\!+ \zeta$ as a suffix,*
4. *there is an $\alpha' \in (N^\bullet \cup T)^*$ with $\alpha' \in (\alpha)_{\theta/(\eta +\!\!+ \zeta)}$ and*

$$\alpha' \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{j-1}$$

Then it follows that:

$$\vdash^G_{w_0,\ldots,w_{n-1}} i, \theta, A \to \alpha \bullet \beta, \eta, \zeta, j$$

Proof. The proof is a variation on the proof of the Adequacy Lemma for Earley parsing with context free grammars (see, e.g., [AU72]).

An instance of the implication that we have to show can be characterized by a list

$$[i, \theta, A, \alpha, \beta, \eta, \zeta, j, \alpha', \gamma, \delta].$$

Let $\mathcal{I}$ range over such instances. The rank $r(\mathcal{I})$ of an instance $\mathcal{I}$ is given by

$$r(\mathcal{I}) = \tau_1(\mathcal{I}) + 2(j + \tau_2(\mathcal{I}) + \tau_3(\mathcal{I}))$$

where

$$\begin{aligned}
\tau_1(\mathcal{I}) &= \text{the length of the shortest derivation } S \stackrel{G}{\Rightarrow}^* \gamma A_\theta \delta \\
\tau_2(\mathcal{I}) &= \text{the length of the shortest derivation } \gamma \stackrel{G}{\Rightarrow}^* w_0, \ldots, w_{i-1} \\
\tau_3(\mathcal{I}) &= \text{the length of the shortest derivation } \alpha' \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{j-1}.
\end{aligned}$$

We prove the implication by induction on instance rank.

Base case. If the rank of an instance $\mathcal{I}$ is 0, then it is easily seen that

$$\tau_1(\mathcal{I}) = j = \tau_2(\mathcal{I}) = \tau_3(\mathcal{I}) = 0.$$

Therefore,

$$\alpha = \alpha' = \gamma = \delta = \epsilon \text{ and } \theta = \eta = \zeta = [].$$

Thus A must be equal to the start symbol S, and what we need to show is that presence of rule $S \to \beta$ in the grammar implies that item

$$0, [], S \to \bullet\beta, [], [], 0$$

is in the store. This follows from the fact that this item is predicted from the axiom $0, [], S' \rightarrow \bullet S, [], [], 0$.

Induction step. Consider instance

$$\mathcal{I} = [i, \theta, A, \alpha, \beta, \eta, \zeta, j, \alpha', \gamma, \delta],$$

and assume that the implication holds for all instances with rank $< r(\mathcal{I})$. Assume that the antecedent of the implication holds for the instance. There are three cases to consider, depending on the form of α.

Case 1. Suppose $\mathcal{I}$ has $\alpha = \chi w$ for some terminal symbol w. Since it is given that $\alpha' \in (\alpha)_{\theta/(\eta +\!\!+ \zeta)}$ and

$$\alpha' \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{j-1}$$

we can conclude that $w = w_{j-1}$ and that $\alpha' = \chi' w$ with $\chi' \in (\chi)_{\theta/(\eta +\!\!+ \zeta)}$, and therefore:

$$\chi' \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{j-2}.$$

Consider instance $[i, \theta, A, \chi, w\beta, \eta, \zeta, j-1, \alpha', \gamma, \delta]$. This instance has rank $< r(\mathcal{I})$, and it satisfies the antecedent of the induction hypothesis implication. So by induction hypothesis we get that item $i, \theta, A \rightarrow \chi \bullet w\beta, \eta, \zeta, j-1$ is in the store. Then by an application of the scan rule, item $i, \theta, A \rightarrow \alpha \bullet \beta, \eta, \zeta, j$ is also in the store.

Case 2. Suppose $\mathcal{I}$ has $\alpha = \chi B$ for some nonterminal symbol B. From $\alpha' \in (\alpha)_{\theta/(\eta +\!\!+ \zeta)}$ and $\alpha = \chi B$ we can conclude that α' has the form $\chi' B_\rho$, with $\chi' \in (\chi)_{\theta/(\rho +\!\!+ \eta +\!\!+ \zeta)}$.

Then it follows from $\alpha' \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{j-1}$ that there is some k with $i \leq k \leq j$ with $\chi' \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{k-1}$ and $B_\rho \stackrel{G}{\Rightarrow}^* w_k, \ldots, w_{j-1}$. From instance

$$[i, \theta, A, \chi, B\beta, \rho, \eta +\!\!+ \zeta, j, \alpha', \gamma, \delta],$$

with rank $< r(\mathcal{I})$, we conclude that item

$$i, \theta, A \rightarrow \chi \bullet B\beta, \rho, \eta +\!\!+ \zeta, j$$

is in store. Let $B \rightarrow \mu$ be the grammar rule that is used in the first step $B \stackrel{G}{\Rightarrow} \mu'$ (with $\mu' \in (\mu)_\rho$) of a minimum length derivation $B_\rho \stackrel{G}{\Rightarrow}^* w_k, \ldots, w_{j-1}$. Now for each $\beta' \in (\beta)_{\eta +\!\!+ \zeta}$, consider the instance

$$[k, \rho, B, \mu, \epsilon, [], [], j, \mu', \gamma\chi', \beta'\delta],$$

Let $\mathcal{I}'$ be such an instance. We show that $r(\mathcal{I}') < r(\mathcal{I})$.

We have $S \stackrel{G}{\Rightarrow}^* \gamma A_\theta \delta \stackrel{G}{\Rightarrow} \gamma\chi' B_\rho \beta' \delta$, so $\tau_1(\mathcal{I}') \leq \tau_1(\mathcal{I}) + 1$.

Let n_1 be the minimum number of steps in a derivation $\chi' \stackrel{G}{\Rightarrow}^* w_i, \ldots, w_{k-1}$, and n_2 the minimum number of steps in a derivation $B_\rho \stackrel{G}{\Rightarrow}^* w_k, \ldots, w_{j-1}$. Then $\tau_3(\mathcal{I}) = n_1 + n_2$. Since $B_\rho \stackrel{G}{\Rightarrow} \mu' \stackrel{G}{\Rightarrow}^* w_k, \ldots, w_{j-1}$, we see that $\tau_3(\mathcal{I}') = n_2 - 1$. Observe that $\tau_2(\mathcal{I}') = \tau_2(\mathcal{I}) + n_1$. It follows that

$$\tau_2(\mathcal{I}') + \tau_3(\mathcal{I}') = \tau_2(\mathcal{I}) + n_1 + n_2 - 1 = \tau_2(\mathcal{I}) + \tau_3(\mathcal{I}) - 1.$$

Thus,

$$\tau_1(\mathcal{I}') + 2(j + \tau_2(\mathcal{I}) + \tau_3(\mathcal{I})) < r(\mathcal{I}),$$

i.e., $r(\mathcal{I}') < r(\mathcal{I})$. By the induction hypothesis we conclude that item $k, \rho, B \rightarrow \mu\bullet, [], [], j$ is in the store. An application of the completion rule yields that

$$i, \theta, A \rightarrow \chi B \bullet \beta, \eta, \zeta, j$$

is in the store too.

Case 3. Suppose $\mathcal{I}$ has $\alpha = \epsilon$. Then $i = j$, $\theta = \eta +\!\!+ \zeta$, and $\tau_3(\mathcal{I}) = 0$. From the given $r(\mathcal{I}) > 0$ it follows that any derivation $S \stackrel{G}{\Rightarrow}^* \gamma A_\theta \delta$ has positive length. For suppose the shortest derivation $S \stackrel{G}{\Rightarrow}^* \gamma A_\theta \delta$ has length 0. Then $\tau_1(\mathcal{I}) = 0$, $\gamma = \epsilon$ and $i = 0$, and therefore $\tau_2(\mathcal{I}) = 0$, and contradiction with the given that $r(\mathcal{I}) > 0$.

So there has to be some rule $B \rightarrow \mu A \nu$ used in the penultimate step of a shortest derivation $S \stackrel{G}{\Rightarrow}^* \gamma A_\theta \delta$, for some stack ρ, i.e.,

$$S \stackrel{G}{\Rightarrow}^* \gamma' B_\rho \delta' \stackrel{G}{\Rightarrow} \gamma' \mu' A_\theta \nu' \delta',$$

with $\gamma'\mu' = \gamma$, $\nu'\delta' = \delta$, and

$$\rho = \rho'' +\!\!+ \theta +\!\!+ \rho', \mu' \in (\mu)_{\rho''}, \nu' \in (\nu)_{\rho'},$$

and $\gamma' B_\rho \delta'$ is the penultimate step in some shortest derivation $S \stackrel{G}{\Rightarrow}^* \gamma A_\theta \delta$. Now consider the following instance $\mathcal{I}'$:

$$[k, \rho, B, \mu, A\nu, \theta, \rho', j, \mu', \gamma', \delta'],$$

for k an integer with

$$\gamma' \stackrel{G}{\Rightarrow}^* w_0, \ldots, w_{k-1} \text{ and } \mu' \stackrel{G}{\Rightarrow}^* w_k, \ldots, w_{j-1}.$$

Let the smallest length of a derivation $\gamma' \stackrel{G}{\Rightarrow}^* w_0, \ldots, w_{k-1}$ be n_1 and let the smallest length of a derivation $\mu' \stackrel{G}{\Rightarrow}^* w_k, \ldots, w_{j-1}$ be n_2. Then:

$$\tau_2(\mathcal{I}') = n_1, \quad \tau_3(\mathcal{I}') = n_2, \text{ and } \tau_2(\mathcal{I}) = n_1 + n_2.$$

By the choice of B, μ', ν', $\tau_1(\mathcal{I}') = \tau_1(\mathcal{I}) - 1$. From these facts, combined with the fact that $\tau_3(\mathcal{I}) = 0$, we get that

$$r(\mathcal{I}') = \tau_1(\mathcal{I}) - 1 + 2(j + \tau_2(\mathcal{I})) = r(\mathcal{I}) - 1.$$

Since item $\mathcal{I}'$ satisfies the antecent of the induction implication, item $k, \rho, B \rightarrow \mu \bullet A\nu, \theta, \rho', j$ is in the store. Since $\theta = \eta +\!\!+ \zeta$ and $A \rightarrow \beta$ is a rule of the grammar, we get by an application of the prediction rule that item $j, \theta, A \rightarrow \bullet\beta, \eta, \zeta, j$ is in the store. □

Theorem 11 (Completeness) *The parsing calculus is complete.*

Proof. Apply Lemma 10 to the special case of instance $[0, [], S', S, \epsilon, [], [], n, S, \epsilon, \epsilon]$. □

9 Complexity Analysis

Consider the general form of an item:

$$(i, \theta, A \to \alpha \bullet \beta, \eta, \zeta, j).$$

Constraints are:

- $i \leq j$: given j, there are $j + 1$ possible choices for i,
- $|\theta| \leq n - j$,
- $\eta +\!\!+ \zeta$ is a suffix of θ: there are at most $|\theta| + 1 = n - j + 1$ such η, ζ.

The possible choices for A, α, β depend on the grammar. Assume for given G there are k such choices. Then it follows that there are

$$\sum_{j=0}^{n-1} k(j+1)(n-j)(n-j+1) = \frac{1}{12}k(n+1)(n+2)(n+3)$$

different items to consider. Thus, if we can prove that the parsing algorithm can be organized in such a way that it considers each item only once, and that the time it spends on the derivation step does not depend of the length on the input, then we have proved that the parsing time is $O(n^3)$.

Lemma 12 *Let G be an unambiguous grammar and let $w_0, \ldots, w_{n-1}$ be a string of terminal symbols. Then the parsing deduction process for G and $w_0, \ldots, w_{j-1}$ can be organized in such a way that item*

$$(i, \theta, A \to \alpha \bullet \beta, \eta, \zeta, j)$$

is derived at most once if $\alpha \neq \epsilon$.

Proof. Let an unambiguous grammar G and a string of terminal symbols $w_0, \ldots, w_{n-1}$ be given. Consider an item of the form

$$(i, \theta, A \to \alpha \bullet \beta, \eta, \zeta, j),$$

with $\alpha \neq \epsilon$. There are three cases: (1) the last symbol of α is a terminal, (2) the last symbol of α is a nonterminal B, or (3) the last symbol of α is an indexed nonterminal B_X.

In case (1), the item can only be derived by the scan rule. Since the scan rule shifts the bullet across the terminal, it is clear that this rule can only be applied once.

In case (2), the item may have been derived in two ways, by means of the completion rule. Case (2.1): $\alpha = \xi B$:

$$\frac{i, \theta, A \to \xi \bullet B\beta, \eta, \zeta, k \quad k, \eta, B \to \gamma\bullet, [], [], j}{i, \theta, A \to \xi B \bullet \beta, \zeta', \zeta'', j} C(\zeta, \zeta', \zeta'', \beta)$$

Suppose there is another derivation, either

$$\frac{i,\theta,A\to\xi\bullet B\beta,\eta',\zeta,k' \quad k',\eta',B\to\gamma'\bullet,[],[],j}{i,\theta,A\to\xi B\bullet\beta,\zeta',\zeta'',j}C(\zeta,\zeta',\zeta'',\beta),$$

or,

$$\frac{i,\theta,A\to\xi\bullet B\beta,(Y:\eta'),\zeta,k' \quad k',\eta',B_Y\to\gamma'\bullet,[],[],j}{i,\theta,A\to\xi B\bullet\beta,\zeta',\zeta'',j}C(\zeta,\zeta',\zeta'',\beta)$$

First look at the first possibility, where $B\to\gamma'$ is involved. Assume $k\neq k'$. Then from $\vdash^G_{w_0,\ldots,w_{n-1}} k,\eta,B\to\gamma\bullet,[],[],j$ and $\vdash^G_{w_0,\ldots,w_{n-1}} k',\eta',B\to\gamma'\bullet,[],[],j$ it follows by Lemma 8 that there are $\gamma_1,\delta_1,\gamma_2,\delta_2,\xi',\xi'',\beta'$ with $\xi'\in(\xi)_\eta$, $\xi''\in(\xi)_{\eta'}$, $\beta'\in(\beta)_\zeta$,

$$S\overset{G}{\Rightarrow}{}^*\gamma_1A_\theta\delta_1\overset{G}{\Rightarrow}\gamma_1\xi'B_\eta\beta'\delta_1\overset{G}{\Rightarrow}{}^*w_0\cdots w_{j-1}\beta\delta_1$$

and

$$S\overset{G}{\Rightarrow}{}^*\gamma_2A\delta_2\overset{G}{\Rightarrow}\gamma_2\xi''B_{\eta'}\beta'\delta_2\overset{G}{\Rightarrow}{}^*w_0\cdots w_{j-1}\beta\delta_2.$$

In the first of these derivations, $\gamma_1\xi'\overset{G}{\Rightarrow}{}^*w_0\cdots w_{k-1}$, and in the second one, $\gamma_2\xi''\overset{G}{\Rightarrow}{}^*w_0\cdots w_{k'-1}$. It follows that there are two distinct derivation trees for $w_0\cdots w_{j-1}$, and contradiction with the non-ambiguity of the grammar.

Assume $k=k'$. Then $\gamma\neq\gamma'$, and again it is easy to find two distinct derivation trees for $w_0\cdots w_{j-1}$, and hence a contradiction with the non-ambiguity of the grammar.

Now look at the possibility that the rule involved in the alternative derivation has the form $B_Y\to\gamma$.

Assume $k=k'$. Then ****

Case (2.2): $\alpha=\xi B$:

$$\frac{i,\theta,A\to\xi\bullet B\beta,(Y:\eta),\zeta,k \quad k,\eta,B_Y\to\gamma\bullet,[],[],j}{i,\theta,A\to\xi B\bullet\beta,\zeta',\zeta'',j}C(\zeta,\zeta',\zeta'',\beta)$$

****

□

10 Closure Properties of Sequentially Indexed Languages

Sequentially indexed languages are closed under union, concatenation, Kleene star, homomorphism, inverse homomorphism and intersection with regular languages (details to be spelled out). Thus, sequentially indexed languages are a so-called fully abstract family of languages [GG67].

Open questions are the following:

- do sequentially indexed grammars generate the same languages as linear indexed grammars in the sense of [Gaz88]?
- how do the classes of sequentially indexed langauges and indexed languages relate? (Does it hold, e.g., that every sequentially indexed language is an indexed language?)

References

[Aho68] Alfred V. Aho. Indexed grammars — an extension of context-free grammars. *Journal of the ACM*, 15(4):647– 671, 1968.

[Aho69] Alfred V. Aho. Nested stack automata. *Journal of the ACM*, 16(3):383–406, 1969.

[AU72] A.V. Aho and J.D. Ullman. *Theory of Parsing, Translation and Compiling*, volume Volume I: Parsing. Englewood Cliffs (NJ), 1972.

[BKPZ82] J. Bresnan, R.M. Kaplan, S. Peters, and A. Zaenen. Cross-serial dependencies in Dutch. *Linguistic Inquiry*, 1982.

[Ear70] J. Earley. An efficient context-free parsing algorithm. *Communications of the ACM*, 13:94–102, 1970.

[Gaz88] G. Gazdar. Applicability of indexed grammars to natural languages. In U. Reyle and C. Rohrer, editors, *Natural Language Parsing and Linguistic Theories*, pages 69–94. Reidel, Dordrecht, 1988.

[GG67] S. Ginsburg and S.A. Greibach. Abstract families of languages. In *Eight Annual Symposium on Switching and Automata Theory*, pages 128–139. IEEE, 1967.

[SSP95] S.M. Shieber, Y. Schabes, and F.C.N. Pereira. Principles and implementation of deductive parsing. *Journal of Logic Programming*, 24:3–36, 1995.