Parsing

Left-Corner Parsing

Laura Kallmeyer

Heinrich-Heine-Universität Düsseldorf

Winter 2017/18

Table of contents

1. Motivation
2. Algorithm
3. Look-ahead
4. Chart Parsing

Motivation

Problems with pure TD/BU approaches:

- Top-Down does not check whether the actual input corresponds to the predictions made.
- Bottom-Up does not check whether the recognized constituents correspond to anything one might predict starting from S.

Mixed approaches help to overcome these problems:

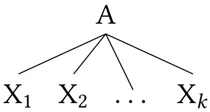
- Left-Corner Parsing parses parts of the tree top down, parts bottom-up.
- Earley-Parsing is a chart-based combination of top-down predictions and bottom-up completions.

Idea

In a production $A \to X_1 \dots X_k$, the first righthand side element X_1 is called the **left corner** of this production.
Notation: $\langle A, X_1 \rangle \in LC$.

Idea:

- Parse the left corner bottom-up while parsing $X_2, \dots, X_k$ top-down.
- In other words, in order to predict the subtree



  a parse tree for X_1 must already be there.

Algorithm (1)

We assume a CFG without ε-productions and without loops.
We need the following three stacks:

- a stack Γ_{compl} containing completed elements that can be used as potential left corners for applying new productions.
  Initial value: w
- a stack Γ_{td} containing the top-down predicted elements of a rhs (i.e., the rhs without the left corner)
  Initial value: S
- a stack Γ_{lhs} containing the lhs categories that are waiting to be completed. Once all the top-down predicted rhs symbols are completed, the category is moved to Γ_{compl}.
  Initial value: ε

Algorithm (2)

Item form $[\Gamma_{compl}, \Gamma_{td}, \Gamma_{lhs}]$ with

- $\Gamma_{compl} \in (N \cup T)^*$,
- $\Gamma_{td} \in (N \cup T \cup \{\$\})^*$ where \$ is a new symbol marking the end of a rhs,
- $\Gamma_{lhs} \in N^*$.

Whenever the symbols $X_2, \dots, X_k$ from a rhs are pushed onto Γ_{td}, they are preceded by \$ to mark the end of a rhs (i.e., the point where a category can be completed).

Axiom: $\dfrac{}{[w, S, \varepsilon]}$

Algorithm (3)

Reduce can be applied if the top of Γ_{compl} is the left corner X_1 of some rule $A \to X_1 X_2 \dots X_k$. Then X_1 is popped, $X_2 \dots X_k \$$ is pushed onto Γ_{td} and A is pushed onto Γ_{lhs}:

$$\text{Reduce: } \frac{[X_1\alpha, B\beta, \gamma]}{[\alpha, X_2 \dots X_k \$ B\beta, A\gamma]} \; A \to X_1 X_2 \dots X_k \in P, B \neq \$$$

Once the entire righthand side has been completed (top of Γ_{td} is \$), the completed category is moved from Γ_{lhs} to Γ_{compl}:

$$\text{Move: } \frac{[\alpha, \$\beta, A\gamma]}{[A\alpha, \beta, \gamma]} \; A \in N$$

Algorithm (4)

A completed category can be a left corner (then reduce is applied) or it can be the next symbol on the Γ_{td} stack, then both can be popped:

Remove: $\dfrac{[X\alpha, X\beta, \gamma]}{[\alpha, \beta, \gamma]}$

The recognizer is successfull if $\Gamma_{compl} = \Gamma_{td} = \Gamma_{lhs} = \varepsilon$:

Goal item: $[\varepsilon, \varepsilon, \varepsilon]$

Algorithm (5)

Example: Left Corner Parsing

Productions:
$S \to aSa \mid bSb \mid c$
input $w = abcba$.

Γ_{compl}	Γ_{td}	Γ_{lhs}	operation
$abcba$	S	ε	
$bcba$	$Sa\$S$	S	reduce
cba	$Sb\$Sa\$S$	SS	reduce
ba	$\$Sb\$Sa\$S$	SSS	reduce
Sba	$Sb\$Sa\$S$	SS	move
ba	$b\$Sa\$S$	SS	remove
a	$\$Sa\$S$	SS	remove
Sa	$Sa\$S$	S	move
a	$a\$S$	S	remove
ε	$\$S$	S	remove
S	S	ε	move
ε	ε	ε	remove

Algorithm (6)

Problematic for left-corner parsing:

- ε-productions: there is no left corner that can trigger a reduce step with an ε-production. If we allow ε-productions to be predicted in reduce steps without a left corner, we would add them an infinite number of times.
- loops: as in the LL-parsing case, loops can cause an infinite sequence of reduce and move steps. This problem is already avoided with the item-based formulation since we would only try to create the same items again.

Both problems can be overcome using the chart-based version with dotted productions described later.

Look-ahead (1)

Idea:

- build the reflexive transitive closure LC^* of the left corner relation LC,
- before applying *reduce*, check whether the top of Γ_{td} stands in the relation LC^* to the lhs of the new production we predict:

Reduce:

$$\frac{[X_1\alpha, B\beta, \gamma]}{[\alpha, X_2 \dots X_k\$B\beta, A\gamma]} \; A \to X_1X_2 \dots X_k \in P, \langle B, A \rangle \in LC^*$$

Difference between LC^* and *First*: LC^* for a given non-terminal can be non-terminals and terminals, while the *First* sets contain only terminals.
$LC^* = \{\langle A, X \rangle \mid A \stackrel{*}{\Rightarrow} X\alpha\}$

Look-ahead (2)

Example:

VP $\to$ V NP, VP $\to$ VP PP, V $\to$ sees,
NP $\to$ Det N, Det $\to$ the, N $\to$ N PP, N $\to$ girl, N $\to$ telescope,
PP $\to$ P NP, P $\to$ with

LC:
$\langle$VP, V$\rangle$, $\langle$VP, VP$\rangle$, $\langle$V, sees$\rangle$ $\langle$NP, Det$\rangle$, $\langle$Det, the$\rangle$,
$\langle$N, N$\rangle$, $\langle$N, girl$\rangle$, $\langle$N, telescope$\rangle$, $\langle$PP, P$\rangle$, $\langle$P, with$\rangle$

$LC^* = LC \cup$:
$\{$$\langle$VP, sees$\rangle$, $\langle$V, V$\rangle$, $\langle$NP, NP$\rangle$, $\langle$NP, the$\rangle$,
$\langle$PP, PP$\rangle$, $\langle$PP, with$\rangle$, $\langle$P, P$\rangle$$\}$

Chart Parsing (1)

Problem of left corner parsing: non-deterministic.

In order to avoid computing partial results several times, we can use tabulation, i.e., adopt chart parsing.

Items we need to tabulate:

- Completely recognized categories: passive items $[X, i, l]$
- Partially recognized productions: active items $[A \to \alpha \bullet \beta, i, l]$ with $\alpha \in (N \cup T)^+, \beta \in (N \cup T)^*$

(i index of first terminal in yield, l length of the yield)

Chart Parsing (2)

Let us again assume a CFG without ε-productions.
We start with the initial items $[w_i, i, 1]$.
The operations *reduce*, *remove* and *move* are then as follows:

- Reduce: If $[X_1, i, l]$ and $A \rightarrow X_1X_2 \dots X_k \in P$, then we add $[A \rightarrow X_1 \bullet X_2 \dots X_k, i, l]$.
- Move: If $[A \rightarrow X_1X_2 \dots X_k \bullet, i, l]$, then we add $[A, i, l]$
- Remove: If $[X, i, l]$ and $[A \rightarrow \alpha \bullet X\beta, j, i - j]$ then we add $[A \rightarrow \alpha X \bullet \beta, j, i - j + l]$.

Chart Parsing (3)

Parsing Schema:

Scan: $\dfrac{}{[w_i, i, 1]}$ $1 \leq i \leq n$

Reduce: $\dfrac{[X, i, l]}{[A \rightarrow X \bullet \alpha, i, l]}$ $A \rightarrow X\alpha \in P$

Remove: $\dfrac{[A \rightarrow \alpha \bullet X\beta, i, l_1], [X, j, l_2]}{[A \rightarrow \alpha X \bullet \beta, i, l_1 + l_2]}$ $j = i + l_1$

Move: $\dfrac{[A \rightarrow \alpha X\bullet, i, l]}{[A, i, l]}$

Goal item: $[S, 1, n]$.

(This is actually the same algo as the CYK with dotted productions seen earlier in the course, except for different names of the rules and a different use of indices.)

Chart Parsing (4)

Example: Left Corner Chart Parsing

Productions: $S \to aSa \mid bSb \mid c$, input $w = abcba$.

item(s)	rule	antecedens items
$[a, 1, 1], [b, 2, 1], [c, 3, 1], [b, 4, 1], [a, 5, 1]$ (axioms)		
$[S \to a \bullet Sa, 1, 1]$	reduce	$[a, 1, 1]$
$[S \to b \bullet Sb, 2, 1]$	reduce	$[b, 2, 1]$
$[S \to c\bullet, 3, 1]$	reduce	$[c, 3, 1]$
$[S \to a \bullet Sb, 4, 1]$	reduce	$[b, 4, 1]$
$[S \to b \bullet Sa, 5, 1]$	reduce	$[a, 5, 1]$
$[S, 3, 1]$	move	$[S \to c\bullet, 3, 1]$
$[S \to bS \bullet b, 2, 2]$	remove	$[S \to b \bullet Sb, 2, 1], [S, 3, 1]$
$[S \to bSb\bullet, 2, 3]$	remove	$[S \to bS \bullet b, 2, 2], [b, 4, 1]$
$[S, 2, 3]$	move	$[S \to bSb\bullet, 2, 3]$
$[S \to aS \bullet a, 1, 4]$	remove	$[S \to a \bullet Sa, 1, 1], [S, 2, 3]$
$[S \to aSa\bullet, 1, 5]$	remove	$[S \to aS \bullet a, 1, 4], [a, 5, 1]$
$[S, 1, 5]$	move	$[S \to aSa\bullet, 1, 5]$

Conclusion

- The left corner of a production is the first element of its rhs.
- Predict a production only if its left corner has already been found.
- In general non-deterministic.
- Problematic for ε-productions and loops.
- Can be implemented as a chart parser with passive and active items.
- In the chart parser, ε-productions can be dealt with (they require an additional Scan-ε rule) and loops are no longer a problem.