Parsing
Parsing as Deduction

Rena Vamadevan
Elina Magamaeva
Yuankun LUO

Referat für Automatische Syntaxanalyse (Parsing) Seminar
Univ.-Prof. Dr. Laura Kallmeyer
Heinrich-Heine-Universität Düsseldorf
Wintersemester 2012/2013
Overview

- Motivation
- Parsing Schemata
- Top-Down Parsing
- Bottom-Up Parsing
- Implementation issues
Motivation

- Frage

Was ist das?

Algorithm:

\[ C_{i,1} := \{ A \mid A \rightarrow w_i \in P \} \]

for all \( l \in [1..n] \):

for all \( i \in [1..n] \):

for every \( A \rightarrow B \ C \):

if there is a \( l_1 \in [1..l-1] \) such that

\[ B \in C_{i,l_1} \text{ and } C \in C_{i+l_1,l-l_1} \],

then \( C_{i,l} := C_{i,l} \cup \{ A \} \)
Motivation

• Pesudo-code

  + relatively close to the proper implementation
  - makes a lot of choices that do not belong to the parsing strategy of the algorithm
Motivation

- **Chart as a data structure**

Example: \( S \rightarrow C_aC_b \mid C_aS_B, S_B \rightarrow SC_b, C_a \rightarrow a, C_b \rightarrow b \). (From \( S \rightarrow aSb \mid ab \) with transformation into CNF.)

\[ w = aaabbb. \]

<table>
<thead>
<tr>
<th>( l )</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>S_B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>S_B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>( C_a )</td>
<td>( C_a )</td>
<td>( C_a )</td>
<td>( C_b )</td>
<td>( C_b )</td>
<td>( C_b )</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>( i )</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>b</td>
<td>b</td>
<td></td>
</tr>
</tbody>
</table>

“It introduces data structures and control structures.”

“do not belong to the parsing strategy of the algorithm”
Motivation

- Parsing as Deduction

Item form: \([A, i, j]\)

Axioms: \([A, i, i + 1] \quad A \rightarrow w_{i+1}\)

Goals: \([S, 0, n]\)

Inference rules:

\[
\frac{[B, i, j] \quad [C, j, k]}{[A, i, k]} \quad A \rightarrow B \ C
\]

Figure 1: The CYK deductive parsing system.
Motivation

• Parsing as Deduction
  » Concentration on parsing strategy;
  » Facilitation of proofs (e.g., soundness and completeness of an algorithm).
  » Soundness: if algo yields true for $w$, then $w \in L(G)$. Completeness: if $w \in L(G)$, then also yields true for $w$.
  » Complexity of an algorithm sometimes easier to determine.
Schemata

Parsing Schemata understand parsing as a deductive process.

Deduction of new items from existing ones can be described using inference rules.

General form:

$$\frac{\text{antecedent}}{\text{consequent}} \quad \text{side conditions}$$

antecedent, consequent: lists of items
Motivation

• Parsing as Deduction

Item form: \([A, i, j]\)

Axioms: \([A, i, i + 1] \quad A \rightarrow w_{i+1}\)

Goals: \([S, 0, n]\)

Inference rules: \(\frac{[B, i, j] \quad [C, j, k]}{[A, i, k]} \quad A \rightarrow B C\)

Figure 1: The CYK deductive parsing system.
Parsing schemata

- Item Form

Simple Example of CYK parsing algorithm
+G=<N, T, P, S>

+Items of the logic are of the form [A, i, j ]
Parsing schemata

• Then whenever we know that

\[ B \Rightarrow^* w_{i+1} \cdots w_j \quad [B, i, j]\]
\[ C \Rightarrow^* w_{j+1} \cdots w_k \quad [C, j, k]\]
\[ A \Rightarrow^* w_{i+1} \cdots w_k \quad [A, i, k]\]
Parsing schemata

- Inference Rules

\[ [B, i, j] \quad [C, j, k] \quad [A, i, k] \]

Inference rules:

\[ \frac{[B, i, j] \quad [C, j, k]}{[A, i, k]} \quad A \rightarrow B \ C \]
Parsing schemata

• Axiom

\[ A \Rightarrow * \]

\[ w_{i+1} \cdots w_j \]

Item form: \([A, i, j]\)

Axioms:

\[ [A, i, i+1] \quad A \rightarrow w_{1+1} \]
Parsing schemata

- Goals

\[ S \Rightarrow^* w_1 \cdots w_n = w. \]

Item form: \([S, 0, n]\)

Goals:

\([S, 0, n]\)
Parsing schemata

Item form: \([A, i, j]\)

Axioms: \([A, i, i + 1] \quad A \rightarrow w_{i+1}\)

Goals: \([S, 0, n]\)

Inference rules: \[
\frac{[B, i, j] \quad [C, j, k]}{[A, i, k]} \quad A \rightarrow B \ C
\]

Figure 1: The CYK deductive parsing system.
Top-Down Parsing

• We use a new item From

\[ S \Rightarrow w_1 \cdots w_j \beta \]

Dort \( \bullet \): break piont in sentential form
Between the portion that has benn recongized (up to j-th element) and the part that has not (\( \beta \))
Top-Down Parsing

• Axiom

\[ S \Rightarrow^* w_1 \cdots w_j \beta \]

[\bullet \, S, \, 0]

Dort \bullet: break point in sentential form
Between the portion that has been recognized (up to j-th element) and the part that has not (\beta)
Top-Down Parsing

- Goal

\[ S \Rightarrow * \; w_1 \cdots w_j \beta \]

\[[\bullet, n]\]

Dort \( \bullet \): break point in sentential form
Between the portion that has been recognized (up to j-th element) and the part that has not (\( \beta \))
Top-Down Parsing

• Inference rules:

\[
[ \bullet w_{j+1} \beta, j ] \quad [ \bullet \beta, j + 1 ]
\]

Was ist die Beziehrung zwischen die beiden Item Form?
Mit Hilfe general form of a rule of inference:

\[
antecedent \quad side \ conditions \\
\text{consequent}
\]
Top-Down Parsing

• Inference rules:

\[
\frac{[\bullet w_{j+1} \beta, j]}{[\bullet \beta, j + 1]} \quad \text{Scanning}
\]
Top-Down Parsing

• Inference rules:
• Whenever the topmost stack symbol is A and there is an B-production $B \rightarrow \gamma$, we can predict this (here with check on length of sentential form):

$$
\frac{[\bullet B\beta, j]}{[\bullet \gamma\beta, j]} \quad B \rightarrow \gamma \quad \text{Prediction}
$$
Top-Down Parsing

Item form: \([ \bullet \beta, j ]\)

Axioms: \([ \bullet S, 0 ]\)

Goals: \([ \bullet, n ]\)

Inference rules:

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

Prediction \[
\frac{[ \bullet B \beta, j ]}{[ \bullet \gamma \beta, j ]} \quad B \rightarrow \gamma
\]

Figure 2: The top-down recursive-descent deductive parsing system.
Top-Down Parsing

- Example

\[ w = w_1 w_2 w_3 = \text{a program halts} \]

<table>
<thead>
<tr>
<th>Grammar Rule</th>
<th>Production</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow NP \ VP )</td>
<td></td>
</tr>
<tr>
<td>( NP \rightarrow Det \ N \ OptRel )</td>
<td></td>
</tr>
<tr>
<td>( NP \rightarrow PN )</td>
<td></td>
</tr>
<tr>
<td>( VP \rightarrow TV \ NP )</td>
<td></td>
</tr>
<tr>
<td>( VP \rightarrow IV )</td>
<td></td>
</tr>
<tr>
<td>( OptRel \rightarrow RelPro \ VP )</td>
<td></td>
</tr>
<tr>
<td>( OptRel \rightarrow \epsilon )</td>
<td></td>
</tr>
<tr>
<td>( Det \rightarrow a )</td>
<td></td>
</tr>
<tr>
<td>( N \rightarrow \text{program} )</td>
<td></td>
</tr>
<tr>
<td>( PN \rightarrow \text{Terry} )</td>
<td></td>
</tr>
<tr>
<td>( PN \rightarrow \text{Shrdlu} )</td>
<td></td>
</tr>
<tr>
<td>( IV \rightarrow \text{halts} )</td>
<td></td>
</tr>
<tr>
<td>( TV \rightarrow \text{writes} )</td>
<td></td>
</tr>
<tr>
<td>( RelPro \rightarrow \text{that} )</td>
<td></td>
</tr>
</tbody>
</table>
Top-Down Parsing

1. [• $S$, 0]
2. [• $NP$ $VP$, 0]
3. [• $Det$ $N$ $OptRel$ $VP$, 0]
4. [• a $N$ $OptRel$ $VP$, 0]
5. [• $N$ $OptRel$ $VP$, 1]
6. [• program $OptRel$ $VP$, 1]
7. [• $OptRel$ $VP$, 2]
8. [• $VP$, 2]
9. [• $IV$, 2]
10. [• halts, 2]
11. [• , 3]
Pure Bottom-Up Parsing

• New item form

\[[\alpha \bullet, j]\]
Pure Bottom-Up Parsing

Item form: $[\alpha \bullet, j]$

Axioms: $[ullet, 0]$

Goals: $[S \bullet, n]$

Inference Rules:

Shift

\[
\frac{[\alpha \bullet, j]}{[\alpha w_{j+1} \bullet, j + 1]}
\]

Reduce

\[
\frac{[\alpha \gamma \bullet, j]}{[\alpha B \bullet, j]} \quad B \rightarrow \gamma
\]
Down-Up Parsing

- Exsample
- \( w = w_1 w_2 w_3 \) = a program halts

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow Det \ N \ OptRel \\
NP & \rightarrow PN \\
VP & \rightarrow TV \ NP \\
VP & \rightarrow IV \\
OptRel & \rightarrow RelPro \ VP \\
OptRel & \rightarrow \epsilon \\
Det & \rightarrow a \\
N & \rightarrow program \\
PN & \rightarrow Terry \\
PN & \rightarrow Shrdlu \\
IV & \rightarrow halts \\
TV & \rightarrow writes \\
RelPro & \rightarrow that
\end{align*}
\]
Pure Bottom-Up Parsing

1  [●, 0]                       AXIOM
2  [a ●, 1]                     SHIFT from 1
3  [Det ●, 1]                   REDUCE from 2
4  [Det program ●, 2]          SHIFT from 3
5  [Det N ●, 2]                REDUCE from 4
6  [Det N OptRel ●, 2]         REDUCE from 5
7  [NP ●, 2]                   REDUCE from 6
8  [NP halts ●, 3]             SHIFT from 7
9  [NP IV ●, 3]                REDUCE from 8
10 [NP VP ●, 3]                REDUCE from 9
11 [S ●, 3]                    REDUCE from 10
Chart Parsing and Tabulation

Wozu?

+ store intermediate parse results
+ 3-dimensional table
+ adding back-pointers to the items in the charts

\[
[A, i, j]
\]
Chart Parsing and Tabulation

CFG:

\[
T = \{a, b, c\} \\
N = \{S, A, B, C, D, T_a, T_b, T_c\} \\
S \rightarrow AC \quad S \rightarrow BD \\
A \rightarrow T_aA \quad A \rightarrow a \\
C \rightarrow T_bH \quad H \rightarrow CT_c \quad C \rightarrow T_bT_c \\
B \rightarrow T_aG \quad G \rightarrow BT_b \quad B \rightarrow T_aT_b \\
D \rightarrow T_cD \quad D \rightarrow c \\
T_a \rightarrow a \quad T_b \rightarrow b \quad T_c \rightarrow c
\]
# Chart Parsing and Tabulation

<table>
<thead>
<tr>
<th>( j )</th>
<th>( S, ((; ;), (; ;)) )</th>
<th>( S )</th>
<th>( C )</th>
<th>( H )</th>
<th>( D )</th>
<th>( T_c, D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>( T_a, A )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>( i )</td>
</tr>
</tbody>
</table>

05.11.12 Parsing as Deduction 31
Conclusion

+ characterize partial parsing results via items;
+ characterize parsing as a deductive process;
+ allow to separate the proper algorithm from data structures and control structures;
+ facilitate tabulation and computation sharing.
References

