Chart Parsers

chart: data structure that stores partial results of the parsing process in such a way that they can be reused. The chart for an n-word sentence consists of:

- \( n + 1 \) vertices
- a number of edges that connect vertices

Slide CS474–17

Chart Parsing: The General Idea

The process of parsing an n-word sentence consists of forming a chart with \( n + 1 \) vertices and adding edges to the chart one at a time.

- Goal: To produce a complete edge that spans from vertex 0 to \( n \) and is of category \( S \).
- There is no backtracking.
- Everything that is put in the chart stays there.
- Chart contains all information needed to create parse tree.

Slide CS474–18

Bottom-UP Chart Parsing Algorithm

Do until there is no input left:

1. If the agenda is empty, get next word from the input, look up word categories, add to agenda (as constituent spanning two positions).
2. Select a constituent from the agenda: constituent \( C \) from \( p_1 \) to \( p_2 \).
3. Insert \( C \) into the chart from position \( p_1 \) to \( p_2 \).
4. For each rule in the grammar of form \( X \to C \cdot X_1 \ldots X_n \), add an active edge of form \( X \to C \cdot X_1 \ldots X_n \) from \( p_1 \) to \( p_2 \).
5. Extend existing edges that are looking for a \( C \).
   (a) For any active edge of form \( X \to X_1 \ldots \circ C \cdot X_n \) from \( p_0 \) to \( p_1 \), add a new active edge \( X \to X_1 \ldots C \cdot X_n \) from \( p_0 \) to \( p_2 \).
   (b) For any active edge of form \( X \to X_1 \ldots X_n \circ C \) from \( p_0 \) to \( p_1 \), add a new (completed) constituent of type \( X \) from \( p_0 \) to \( p_2 \) to the agenda.

Slide CS474–19

Slide CS474–20
Grammar and Lexicon

Grammar:
1. $S \rightarrow NP \ VP$
2. $NP \rightarrow ART \ N$
3. $NP \rightarrow ART \ ADJ \ N$
4. $VP \rightarrow V \ NP$

Lexicon:
the: ART
man: N, V
old: ADJ, N
boat: N

Sentence: 1 The 2 old 3 man 4 the 5 boat 6
Efficient Parsing

\[ n = \text{sentence length} \]

Time complexity for naive algorithm: exponential in \( n \)

Time complexity for bottom-up chart parser: \( O(n^3) \)

Options for improving efficiency:

1. Don’t do twice what you can do once.
2. Don’t represent distinctions that you don’t need.
   - Fall leaves fall and spring leaves spring.
3. Don’t do once what you can avoid altogether.
   - The can holds the water. ("can": AUX, V, N)

---

Earley Algorithm: Top-Down Chart Parser

For all S rules of the form \( S \rightarrow X_1 \ldots X_k \), add a (top-down) edge from 1 to 1 labeled: \( S \rightarrow \circ X_1 \ldots X_k \).

Do until there is no input left:

1. If the agenda is empty, look up word categories for next word, add to agenda.
2. Select a constituent from the agenda: constituent \( C \) from \( p_1 \) to \( p_2 \).
3. Using the (bottom-up) edge extension algorithm, combine \( C \) with every active edge on the chart (adding \( C \) to chart as well). Add any new constituents to the agenda.
4. For any active edges created in Step 3, add them to the chart using the top-down edge introduction algorithm.

---

Grammar and Lexicon

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Lexicon</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( S \rightarrow \text{NP } \text{VP} )</td>
<td>the: ART</td>
</tr>
<tr>
<td>2. ( \text{NP} \rightarrow \text{ART } \text{ADJ } \text{N} )</td>
<td>large: ADJ</td>
</tr>
<tr>
<td>3. ( \text{NP} \rightarrow \text{ART } \text{N} )</td>
<td>can: N, AUX, V</td>
</tr>
<tr>
<td>4. ( \text{NP} \rightarrow \text{ADJ } \text{N} )</td>
<td>hold: N, V</td>
</tr>
<tr>
<td>5. ( \text{VP} \rightarrow \text{AUX } \text{VP} )</td>
<td>water: N, V</td>
</tr>
<tr>
<td>6. ( \text{VP} \rightarrow \text{V } \text{NP} )</td>
<td></td>
</tr>
</tbody>
</table>

Sentence: 1. The 2 large 3 can 4 can 5 hold 6 water 7