

Chart Parsing

- Parsing: Efficiency Issues
- Chart Parsing
 - Charts
 - Edges
 - Rules
- Chart Parsing in Prolog
- Demonstration

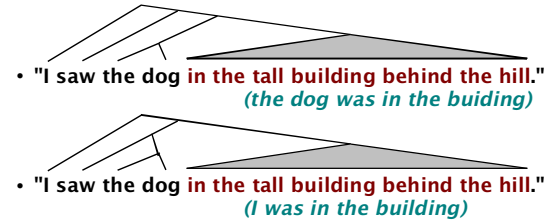


1

CSE-391

Parsing: Efficiency Issues

- Problems with simple top-down parsing:
 - "Left-recursive" rules can cause infinite loops
 - NP → NP and NP
 - Redundant parsing of phrases.



2

CSE-391

Parsing: Efficiency Issues (2)

- Problems with simple bottom-up parsing:
 - Builds structures that are *locally* valid but not *globally* useful.
- Solutions:
 - Re-use the sub-parses we've already computed
 - Combine top-down and bottom-up approaches
 - Get the "best of both worlds"
 - We need some *common representation* for the information from top-down and bottom-up approaches.
 - Use heuristics to decide when to use bottom-up or top-down approaches.

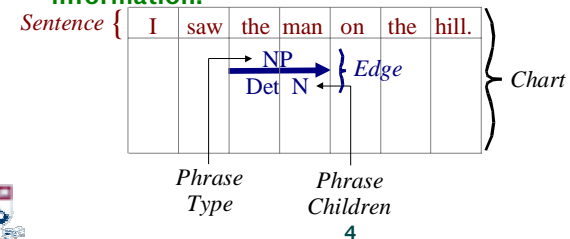


3

CSE-391

Chart Parsing

- Use a *chart* to record hypotheses about possible syntactic constituents.
 - A chart contains a set of *edges*.
- Each *edge* represents a possible phrase.
 - Edges provide a common representation for parse information.

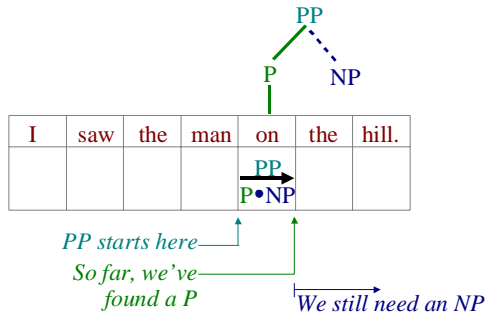


4

CSE-391

Edges

Edges can represent *partial* phrases.



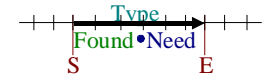
5

CSE-391

Edges (continued)

• An edge consists of:

- S: A start index (1...n)
- E: An end index (1...n)
- Type: A phrase type (NP, PP, etc.)
- Found: What we've found so far (list of phrase types)
- Need: What we still need (list of phrase types)



• Representing edges in Prolog:

- edge(S, E, Type, Found, Need)



6

CSE-391

Chart Parser Rules

- A chart parser *rule* adds new edges to the chart.
- Each chart parsing *strategy* defines a set of rules.
 - **Top down:**
 - top-down initialization rule
 - top-down rule
 - fundamental rule
 - **Bottom-up:**
 - bottom-up rule
 - fundamental rule

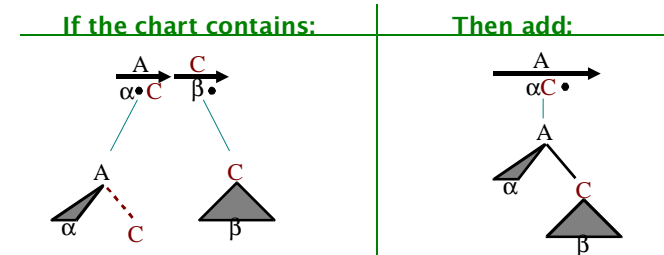


7

CSE-391

The Fundamental Rule

- The fundamental rule is used by both top-down and bottom-up strategies.



8

CSE-391

Top-Down Rules

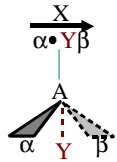
- **Top-down initialization:**

For any rule $S \rightarrow \alpha$:

- Add $\frac{S}{\alpha}$ to the left side of the chart (start=end=1).

- **Top-down rule:**

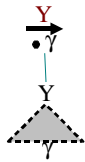
If the chart contains:



For each rule:

$Y \rightarrow \gamma$

Add:



9

CSE-391

Bottom-Up Rules

- **Bottom-Up Rule**

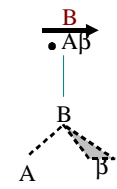
If the chart contains:



For each rule:

$B \rightarrow A\beta$

Add:



10

CSE-391

Chart Parsing in Prolog

- **Define edges using a relation:**

- `edge(S, E, Type, Found, Need, ParseTree)`

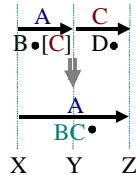
- **Define resulting parse trees using a structure:**

- `parsetree(Type, ChildList)`

- **Define rules as conditions on the edge relation.**

- **Example: Fundamental rule**

```
edge(X, Z, A, BC, [], parsetree(A,PT1,PT2)) :-
    edge(X, Y, A, B, [C], PT1),
    edge(Y, Z, C, D, [], PT2),
    append(B, [C], BC).
```



11

CSE-391

Chart Parsing Demo...



12

CSE-391