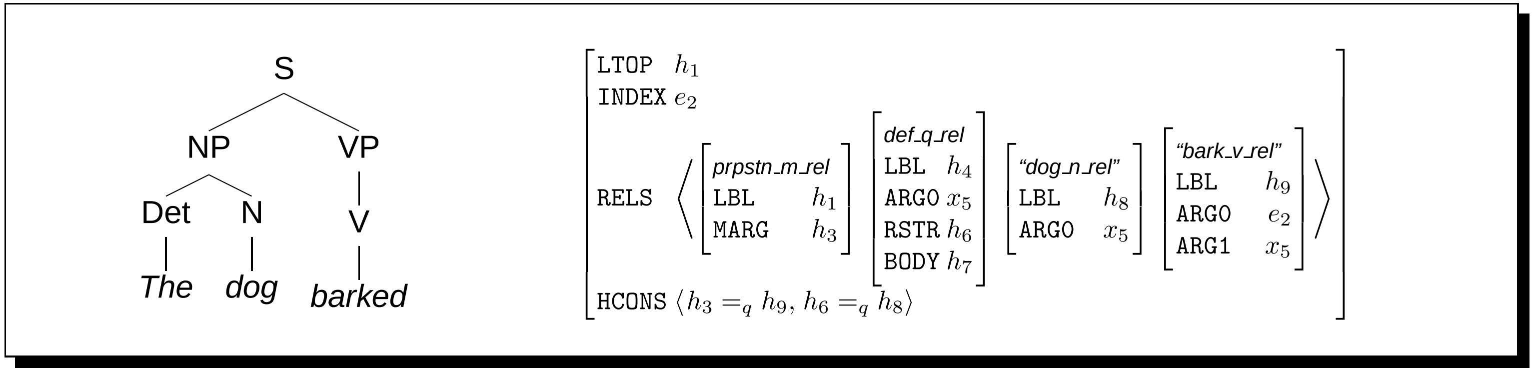


Algorithms for AI and NLP (INF4820 — PCFGs)

$$P(S \longrightarrow NP\ VP) = 1.0;\ \ P(NP \longrightarrow Det\ N) = 0.6$$

Stephan Oepen and Jonathon Read

Universitetet i Oslo

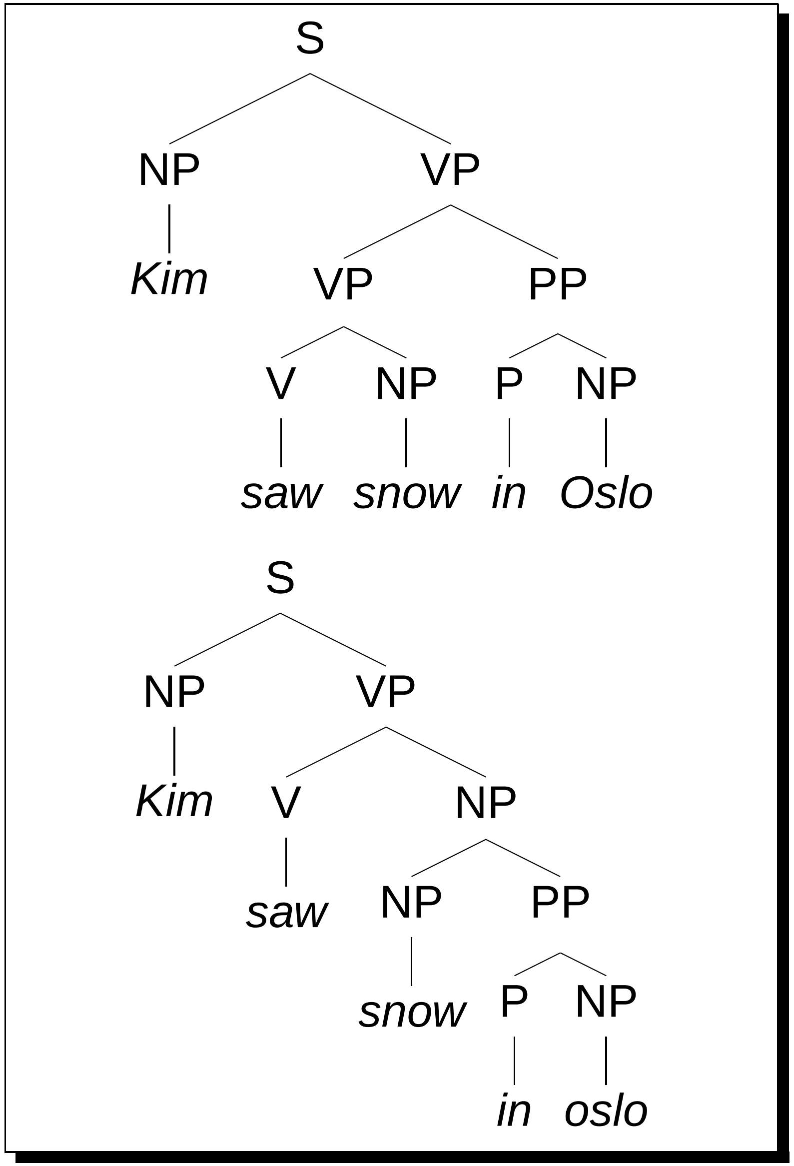
{ oe | jread }@ifi.uio.no

Parsing: Recognizing the Language of a Grammar

S $\rightarrow$ NP VP
VP $\rightarrow$ V | V NP | VP PP
NP $\rightarrow$ NP PP
PP $\rightarrow$ P NP
NP $\rightarrow$ Kim | snow | Oslo
V $\rightarrow$ saw
P $\rightarrow$ in

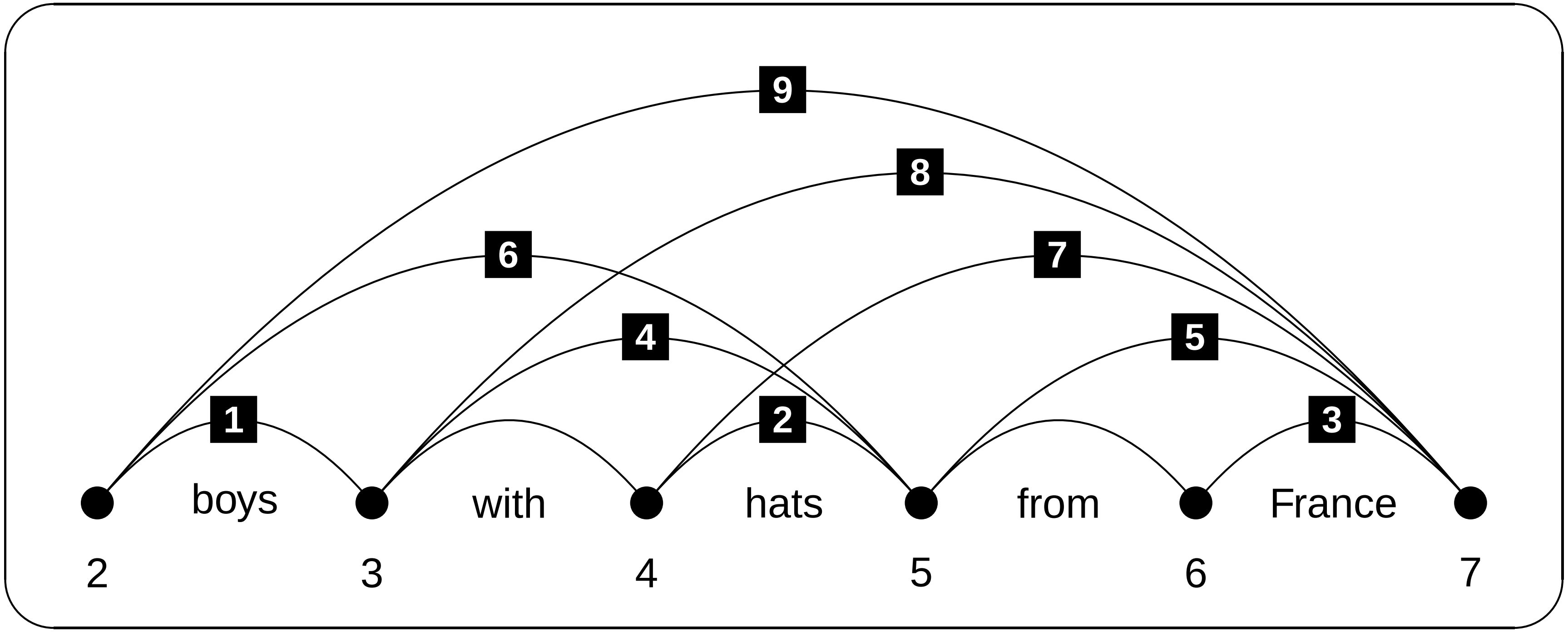
All Complete Derivations

- are rooted in the start symbol S;
- label internal nodes with categories $\in C$, leafs with words $\in \Sigma$;
- instantiate a grammar rule $\in P$ at each local subtree of depth one.



Bounding Ambiguity — The Parse Chart

- For many substrings, more than one way of deriving the same category;
- NPs: $\boxed{1}$ | $\boxed{2}$ | $\boxed{3}$ | $\boxed{6}$ | $\boxed{7}$ | $\boxed{9}$; PPs: $\boxed{4}$ | $\boxed{5}$ | $\boxed{8}$; $\boxed{9} \equiv \boxed{1} + \boxed{8} \mid \boxed{6} + \boxed{5}$;
- *parse forest* — a single item represents multiple trees [Billot & Lang, 89].



The CKY (Cocke, Kasami, & Younger) Algorithm

for $(0 \leq i < |\mathit{input}|)$ do
  $\mathit{chart}_{[i,i+1]} \leftarrow \{\alpha \mid \alpha \rightarrow \mathit{input}_i \in P\}$;
for $(1 \leq l < |\mathit{input}|)$ do
  for $(0 \leq i < |\mathit{input}| - l)$ do
    for $(1 \leq j \leq l)$ do
      if $(\alpha \rightarrow \beta_1\,\beta_2 \in P \wedge \beta_1 \in \mathit{chart}_{[i,i+j]} \wedge \beta_2 \in \mathit{chart}_{[i+j,i+l+1]})$ then
        $\mathit{chart}_{[i,i+l+1]} \leftarrow \mathit{chart}_{[i,i+l+1]} \cup \{\alpha\}$;

$$[0,2] \leftarrow [0,1] + [1,2]$$
$$\ldots$$
$$[0,5] \leftarrow [0,1] + [1,5]$$
$$[0,5] \leftarrow [0,2] + [2,5]$$
$$[0,5] \leftarrow [0,3] + [3,5]$$
$$[0,5] \leftarrow [0,4] + [4,5]$$

	1	2	3	4	5
0	NP		S		S
1		V	VP		VP
2			NP		NP
3				P	PP
4					NP

Limitations of the CKY Algorithm

Built-In Assumptions

- *Chomsky Normal Form* grammars: $\alpha \rightarrow \beta_1\beta_2$ or $\alpha \rightarrow \gamma$ ($\beta_i \in C$, $\gamma \in \Sigma$);
- breadth-first (aka exhaustive): always compute all values for each cell;
- rigid control structure: bottom-up, left-to-right (one diagonal at a time).

Generalized Chart Parsing

- Liberate order of computation: no assumptions about earlier results;
- *active edges* encode partial rule instantiations, 'waiting' for additional (adjacent and passive) constituents to complete: $[1, 2, \text{VP} \rightarrow \text{V} \bullet \text{NP}]$;
- parser can fill in chart cells in *any* order and guarantee completeness.

Backpointers: Recording the Derivation History

	0	1	2	3
0	2: S $\rightarrow \bullet$ NP VP 1: NP $\rightarrow \bullet$ NP PP 0: NP $\rightarrow \bullet$ kim	10: S $\rightarrow$ 8 $\bullet$ VP 9: NP $\rightarrow$ 8 $\bullet$ PP 8: NP $\rightarrow$ kim $\bullet$		17: S $\rightarrow$ 8 15 $\bullet$
1		5: VP $\rightarrow \bullet$ VP PP 4: VP $\rightarrow \bullet$ V NP 3: V $\rightarrow \bullet$ adored	12: VP $\rightarrow$ 11 $\bullet$ NP 11: V $\rightarrow$ adored $\bullet$	16: VP $\rightarrow$ 15 $\bullet$ PP 15: VP $\rightarrow$ 11 13 $\bullet$
2			7: NP $\rightarrow \bullet$ NP PP 6: NP $\rightarrow \bullet$ snow	14: NP $\rightarrow$ 13 $\bullet$ PP 13: NP $\rightarrow$ snow $\bullet$
3				

- Use edges to record derivation trees: backpointers to daughters;
- a single edge can represent multiple derivations: backpointer sets.

Ambiguity Packing in the Chart

General Idea

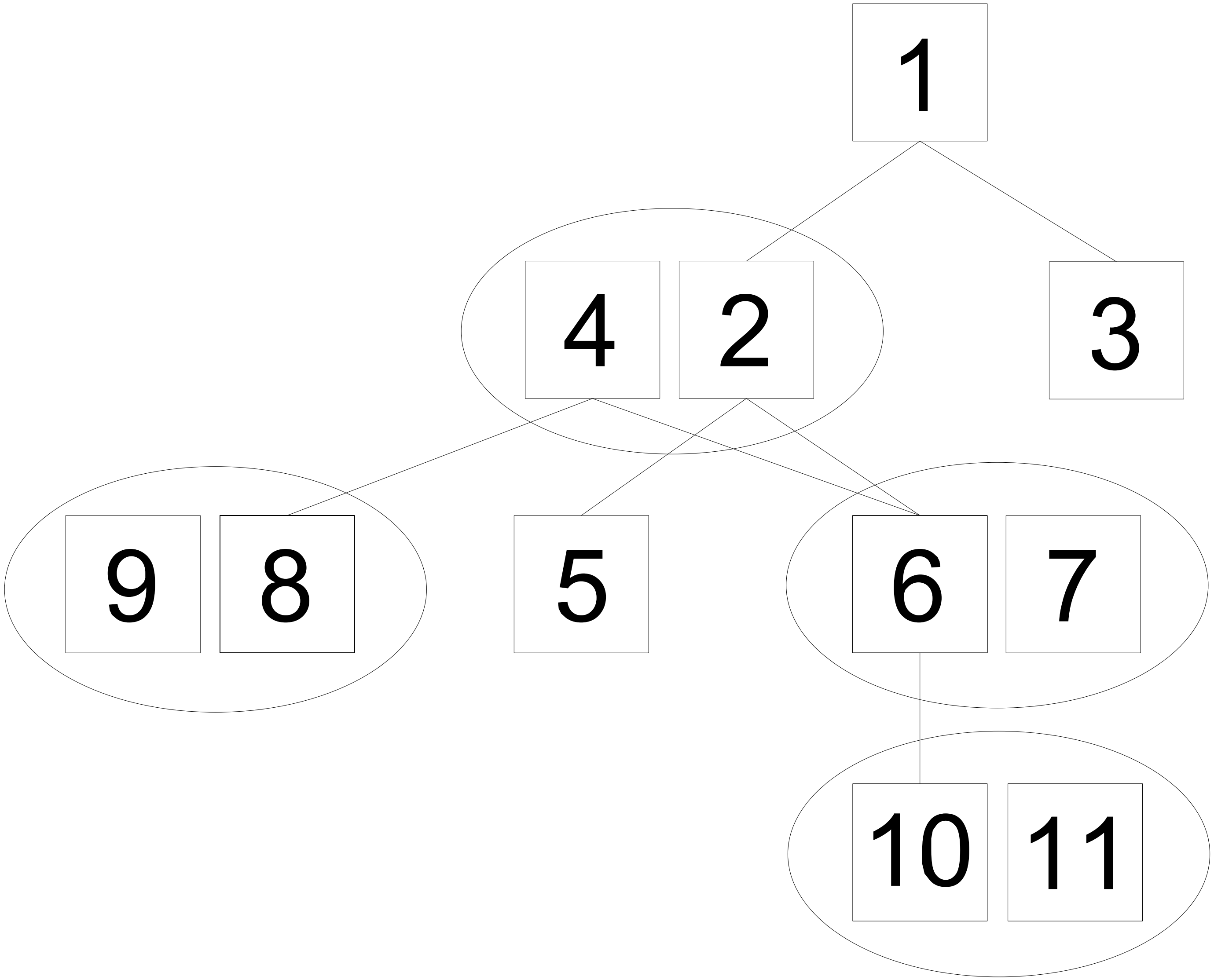
- Maintain only one edge for each α from i to j (the 'representative');
- record alternate sequences of daughters for α in the representative.

Implementation

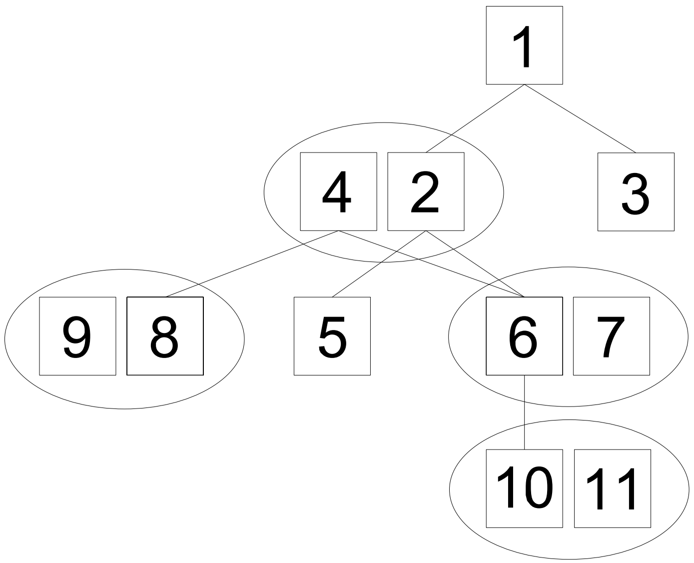
- Group passive edges into *equivalence classes* by identity of α, i, and j;
- search chart for existing equivalent edge (h, say) for each new edge e;
- when h (the 'host' edge) exists, *pack* e into h to record equivalence;
- e *not* added to the chart, no derivations with or further processing of e;

$\rightarrow$ *unpacking* multiply out all alternative daughters for all result edges.

An Example (Hypothetical) Parse Forest



Unpacking: Cross-Multiplying Local Ambiguity



$$\boxed{1} \longrightarrow \langle \boxed{2}\,\boxed{3} \rangle \mid \langle \boxed{4}\,\boxed{3} \rangle$$
$$\boxed{2} \longrightarrow \langle \boxed{5}\,\boxed{6} \rangle \mid \langle \boxed{5}\,\boxed{7} \rangle$$
$$\boxed{4} \longrightarrow \langle \boxed{8}\,\boxed{6} \rangle \mid \langle \boxed{8}\,\boxed{7} \rangle \mid \langle \boxed{9}\,\boxed{6} \rangle \mid \langle \boxed{9}\,\boxed{7} \rangle$$
$$\boxed{6} \longrightarrow \langle \boxed{10} \rangle \mid \langle \boxed{11} \rangle$$

How many complete trees in total?

Ambiguity Resolution Remains a (Major) Challenge

The Problem

- With broad-coverage grammars, even moderately complex sentences typically have multiple analyses (tens or hundreds, rarely thousands);
- unlike in grammar writing, exhaustive parsing is useless for applications;
- identifying the 'right' (intended) analysis is an 'AI-complete' problem;
- inclusion of (non-grammatical) sortal constraints is generally undesirable.

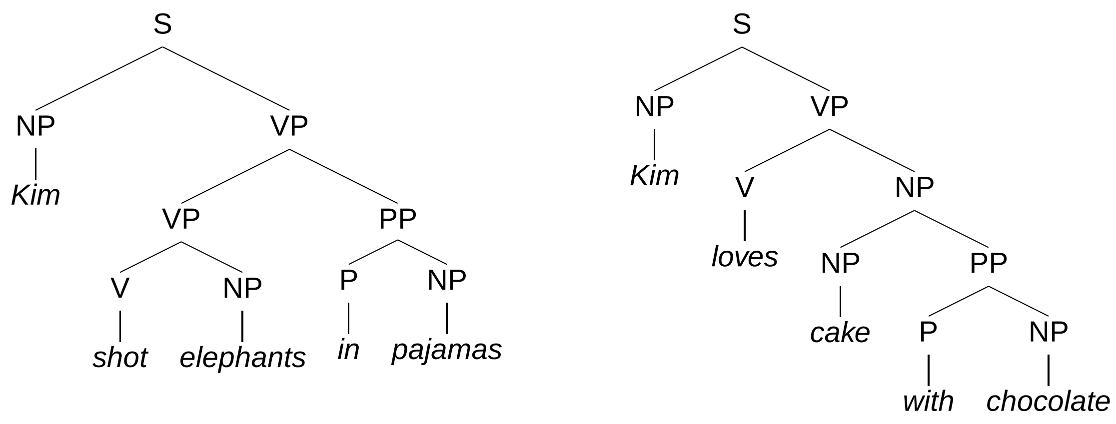
Typical Approaches

- Design and use statistical models to select among competing analyses;
- for string S, some analyses T_i are more or less likely: maximize $P(T_i|S)$;

$\rightarrow$ Probabilistic Context Free Grammar (PCFG) is a CFG plus probabilities.

Probabilistic Context-Free Grammars

A (Simplified) PCFG Estimation Example



P(RHS|LHS)	CFG Rule
	S $\longrightarrow$ NP VP
	VP $\longrightarrow$ VP PP
	VP $\longrightarrow$ V NP
	PP $\longrightarrow$ P NP
	NP $\longrightarrow$ NP PP
	VP $\longrightarrow$ V

- Estimate rule probability from observed distribution;

$\longrightarrow$ conditional probabilities:

$$P(\mathrm{RHS}|\mathrm{LHS}) = \frac{C(\mathrm{LHS}, \mathrm{RHS})}{C(\mathrm{LHS})}$$

Formally: Probabilistic Context-Free Grammars

- Formally, a *context-free grammar* (CFG) is a quadruple: $\langle C, \Sigma, P, S \rangle$

...

- P is a set of category rewrite rules (aka *productions*), each with a conditional probability P(RHS|LHS), e.g.

...

NP $\rightarrow$ Kim [0.6]
NP $\rightarrow$ snow [0.4]

...

- for each rule '$\alpha \rightarrow \beta_1, \beta_2, ..., \beta_n$' $\in P$: $\alpha \in C$ and $\beta_i \in C \cup \Sigma$; $1 \leq i \leq n$;

...

- for each $\alpha \in C$, the probabilities of all rules R '$\alpha \rightarrow$ ...' must sum to 1.

Background: The Penn Treebank (PTB)

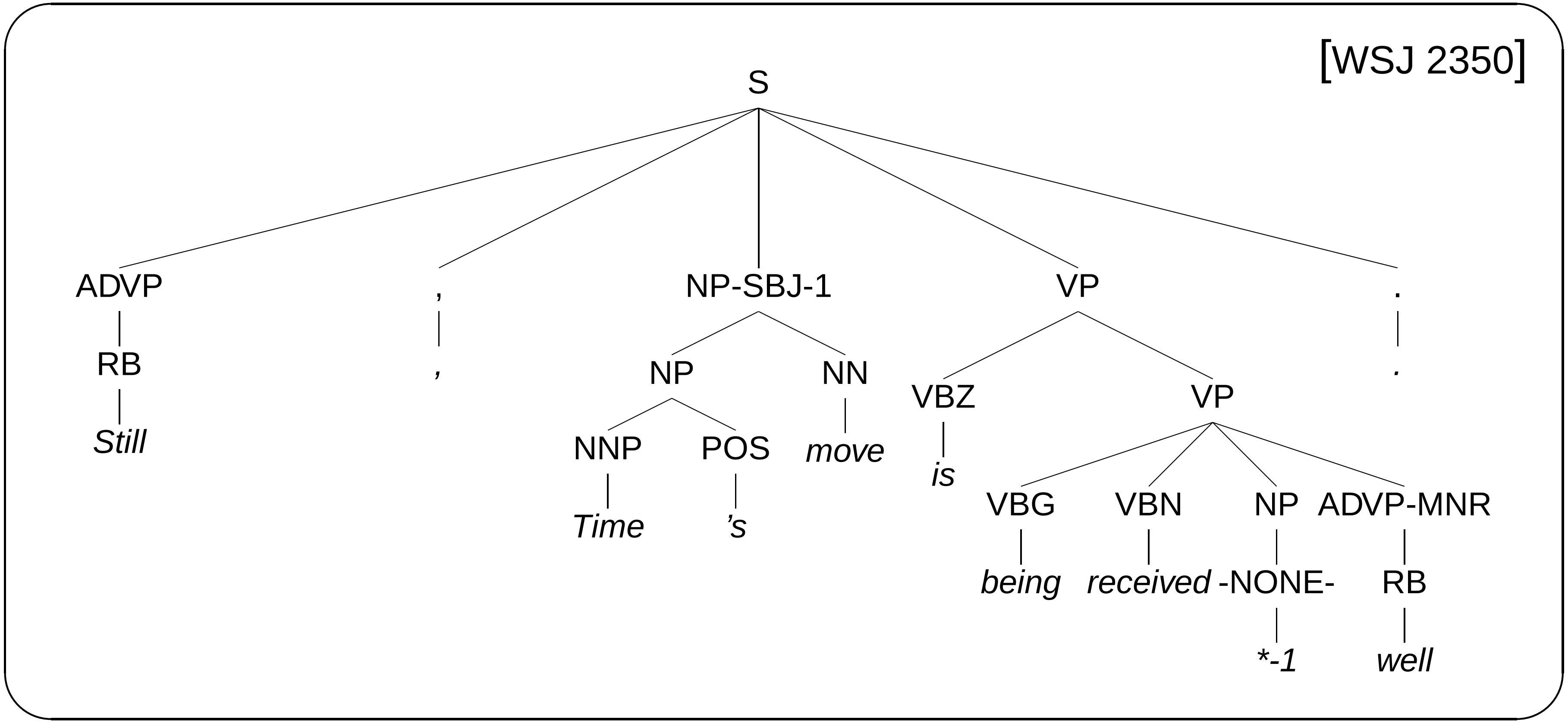
Quite Generally

- A *treebank* is a corpus paired with ‘gold-standard’ (syntactic) analyses;
- used for training and evaluation of NLP tasks, e.g. statistical parsing;
- variation in annotation types, e.g. phrase structure vs. dependencies;
- manual annotation vs. selection among parser outputs (plus correction).

Penn Treebank (Marcus et al., 1993)

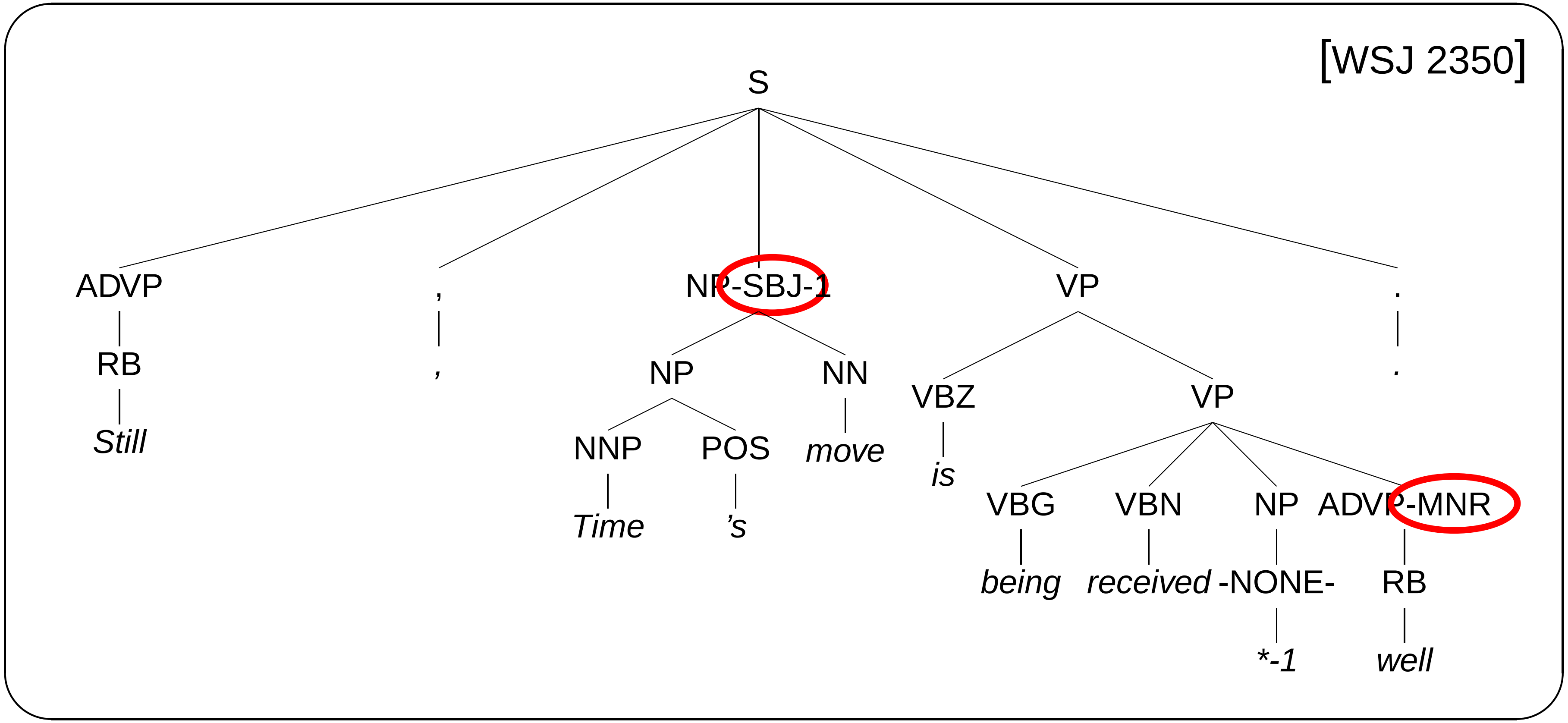
- About one million tokens of Wall Street Journal text (from late 1990s);
- hand-corrected PoS annotation using 45 word classes (the PTB tag set);
- manual syntactic annotation with (somewhat) coarse phrase structure.

One Example from the Penn Treebank



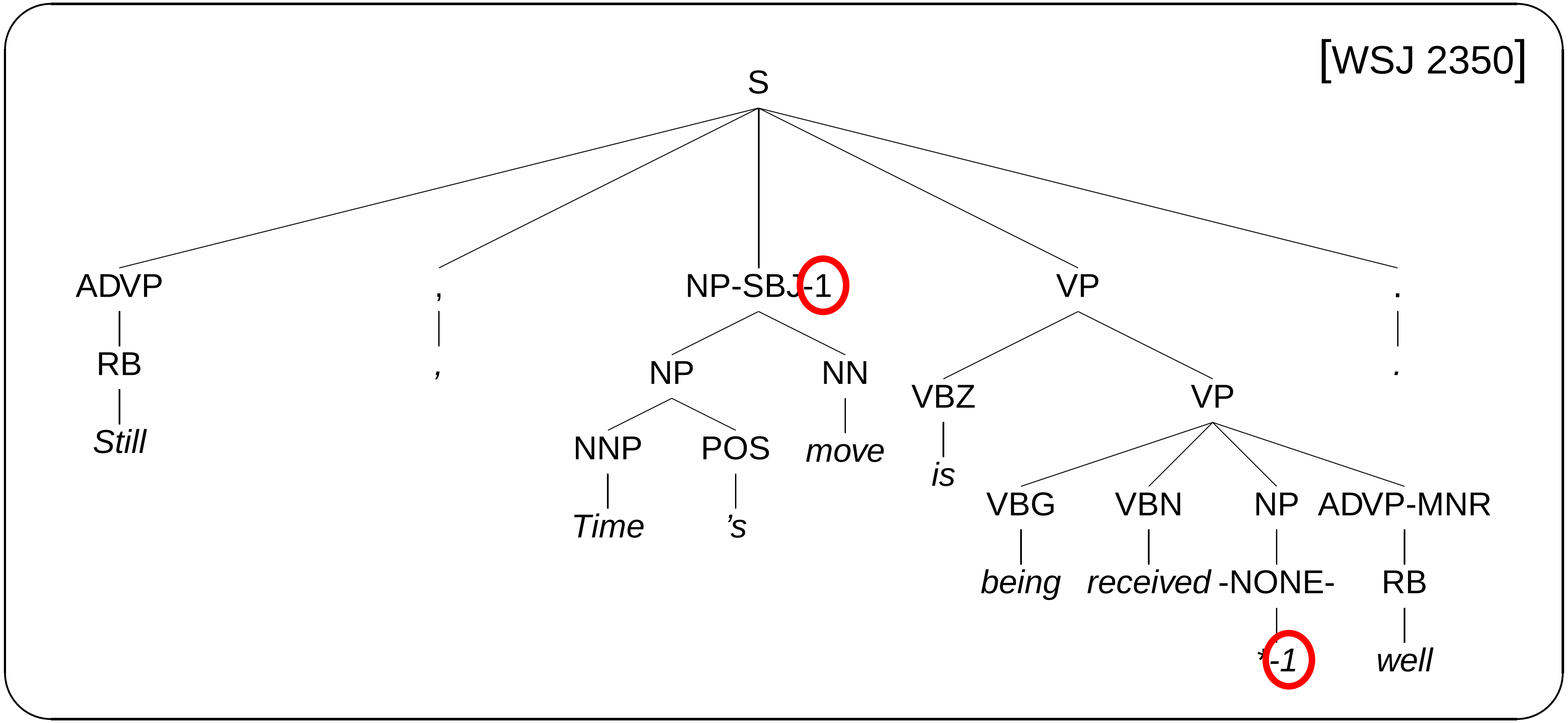
Still, Time's move is being received well.

One Example from the Penn Treebank



Still, Time's move is being received well.

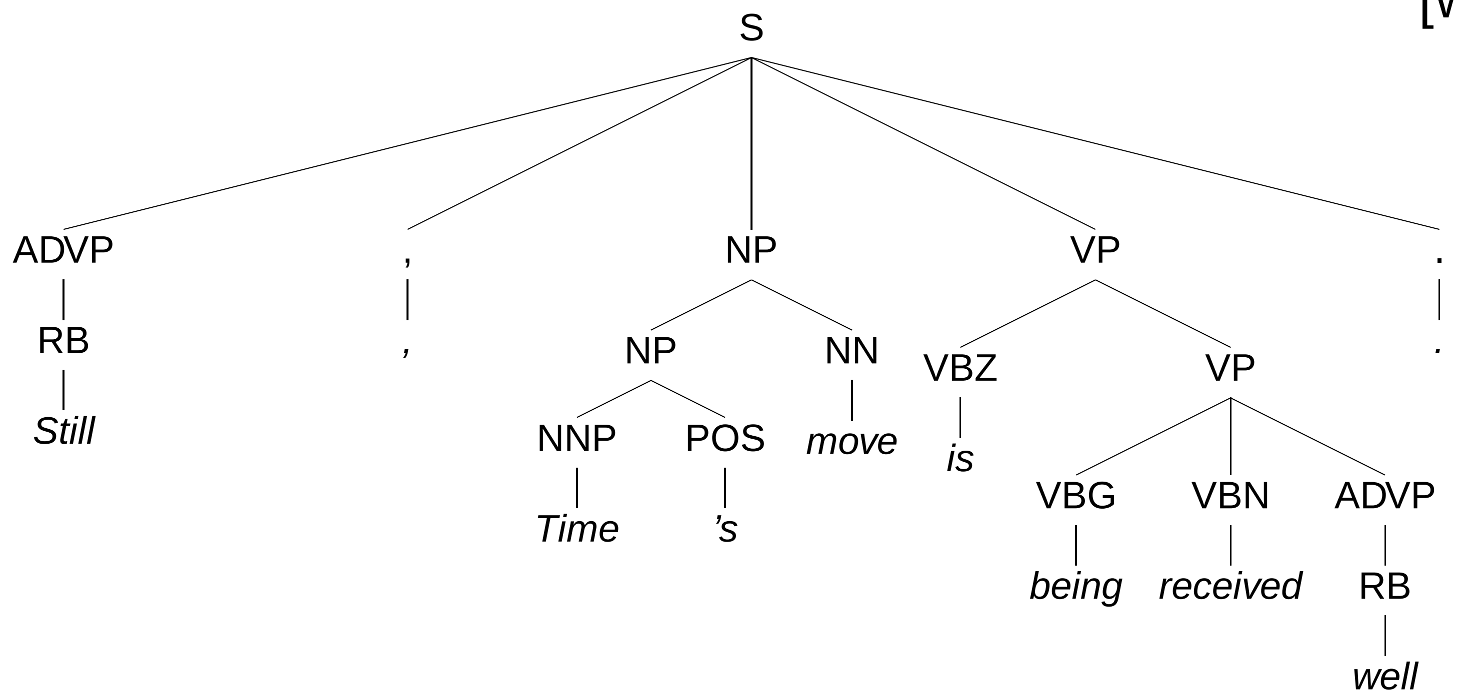
One Example from the Penn Treebank



Still, Time's move is being received well.

(Standard) Elimination of Traces and Functions

[WSJ 2350]



Still, Time's move is being received well.

How to Evaluate (Syntactic) Parsing Accuracy?

ParsEval — Constituent Overlap (Black, et al., 1991)

- Break up tree into bracketing plus labelling, for example:

$$\langle 0, 1, \text{ADVP} \rangle\ \langle 2, 5, \text{NP} \rangle\ \langle 5, 9, \text{VP} \rangle\ \langle 6, 9, \text{VP} \rangle\ \langle 0, 10, \text{S} \rangle$$

- quantify precision (P) and recall (R) of labelled bracketings, when contrasting the gold-standard tree vs. the actual parser output:

$$P = \frac{C(\textit{correct})}{C(\textit{parse})}; \quad R = \frac{C(\textit{correct})}{C(\textit{gold})};$$

- F Score, as the harmonic mean of precision and recall: $F_1 = \frac{2PR}{P+R}$;

$\rightarrow$ combined with crossing brackets, dominant metric in PTB parsing.