Probabilistic Parsing

Detlef Prescher and Khalil Sima'an

Institute For Logic, Language and Computation

Universiteit van Amsterdam

SESSION **Data Oriented Parsing**

LECTURER **Khalil Sima'an**

EUROPEAN SUMMER SCHOOL IN LOGIC LANGUAGE AND INFORMATION (ESSLLI 2003)

Beyond Treebank PCFGs

Given a treebank TB that consists of sentence-parse pairs sampled from language

$$P^* : V^+ \times \mathcal{T} \rightarrow [0,1]$$

Treebank PCFG: Read off PCFG from treebank implies the assumption

P^* belongs to the family of possible PCFG models

Question: *If treebank TB is a sample from an unknown language P^*, why should we assume that P^* is member of the PCFG family?*

Now: we assume that P^* is an interpolation of all possible models that use
(1) subtrees as grammar productions and
(2) the substitution operation for rewriting!

Data Oriented Parsing (DOP)

- The competence-performance distinction.
- Why move away from enriching linguistic Phrase-Structure rules with probabilities?
- Examples where problems arise with the Probabilitic linguistic-CFG.
- Remko Scha's original DOP model (1990).
- A first instantiation: DOP1 (Bod 1992).
- Stochastic Tree-Substitution Grammars (STSGs).
- Comparison between DOP1 and PCFG.
- Where things might go wrong?

Competence or Performance models

The competence/performance distinction

- A **competence model** aims at *characterizing a person's knowledge of a language*
- A **performance model** aims at *describing the actual production and perception of natural language sentences in concrete situations*

We are building performance models

"Competence probabilistic grammars"

Performance=probabilities?

"take your favorite linguistic theory and extract probabilities for the linguistic rules"

Example: Phrase-Structure PCFG

Why? Are probabilities over competence production/rewrite units sufficient for performance?

What should probabilities in a probabilistic grammar capture?

What should probabilities capture?

In syntactic parsing, we would expect probabilities to deal with

Further linguistic factors: semantic, contextual and discourse

Beyond competence: Factors such as world-knowledge

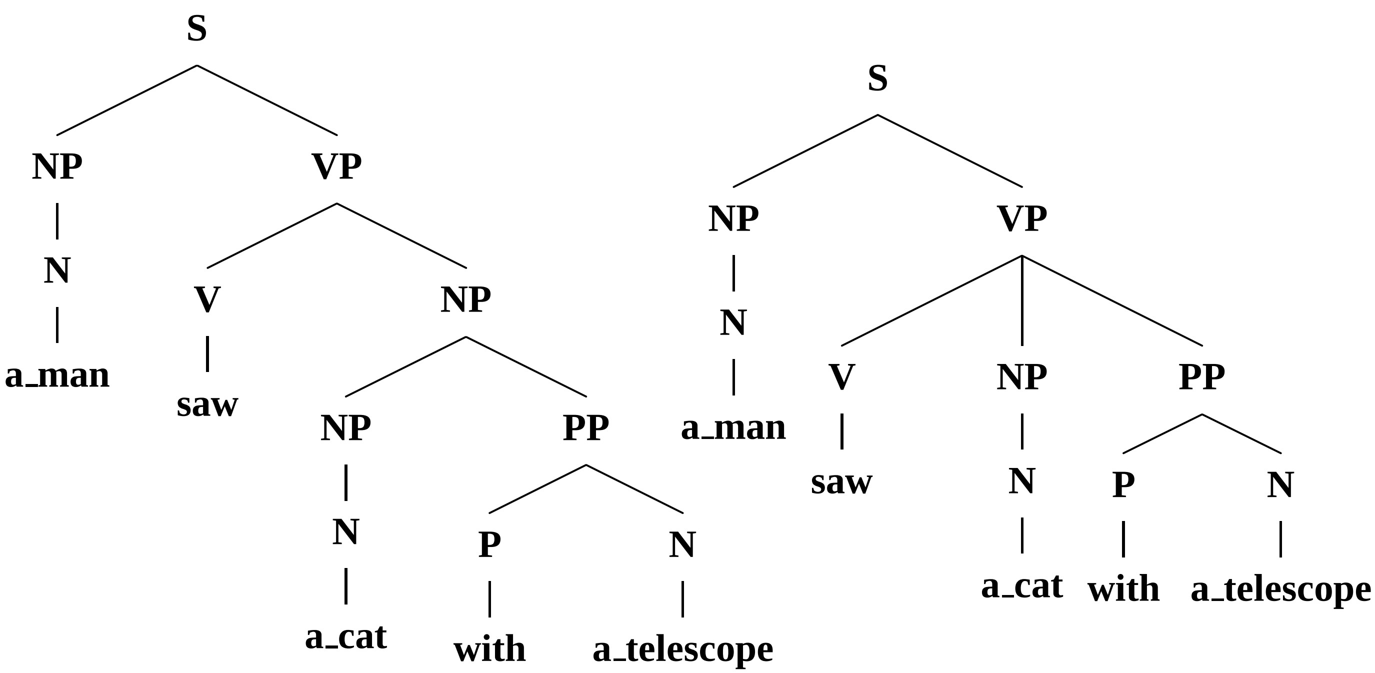
`Object(eat, Pizza)` vs. `Tool(eat, fork)`

`P(bark(dog))` $>>$ `P(bark(snake))`

Frequency effects: preference for more frequent in disambiguation

Uncertainty: hazard and error in the environment

Probabilistic competence grammars?



S	$\rightarrow$	NP VP
VP	$\rightarrow$	V NP
VP	$\rightarrow$	V NP PP
NP	$\rightarrow$	NP PP
PP	$\rightarrow$	P N
N	$\rightarrow$	a_man
N	$\rightarrow$	a_cat
N	$\rightarrow$	a_t.scope
P	$\rightarrow$	with
V	$\rightarrow$	saw

Which tree is more plausible?

Scha 1990 section 6

Current stochastic grammars operate with units that are too small: rewrite rules which describe one level of the constituent structure, and whose application probabilities are supposed to be context-independent. Instead, we would like to *use the statistical approach while working with larger units.*

There is in fact a linguistic tradition which has been thinking in this direction. Bolinger (1961, 1976), $\cdots$, have distanced themselves emphatically from the usual formal grammars. They assign a central role to the *concrete language data*; they view new utterances as built up out of fragments culled from previously processed text; *idiomaticity is the rule rather than the exception*.

$\vdots$

The human language interpretation process has a strong preference for recognizing sentences, phrases and patterns that have occurred before. Structures and interpretations which occurred frequently are preferred above alternatives which have not or rarely been experienced before.

⋮

The amount of information that is necessary for a realistic performance-model is therefore much larger than the grammars that we are used to. The language experience of an adult language user consists of a large number of utterances. And every utterance contains a multitude of constructions: not only the whole sentence, and all its constituents, but also all patterns that we can abstract from these by substituting "free variables" for lexical elements or complex constituents.

The intuitive DOP idea: a sketch

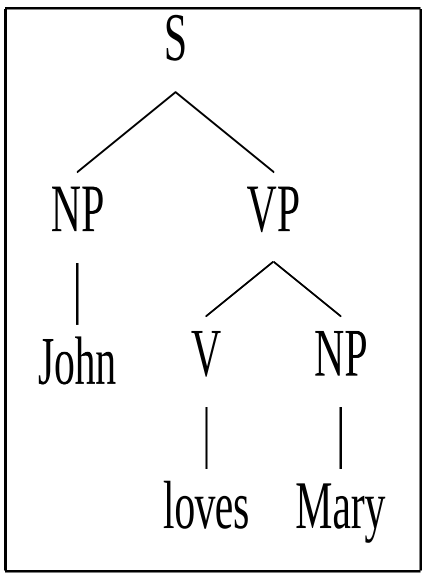
Parsing a new sentence proceeds by

(1) combining "fragments" extracted from the parse-trees in the training tree-bank into parses for the input sentence.

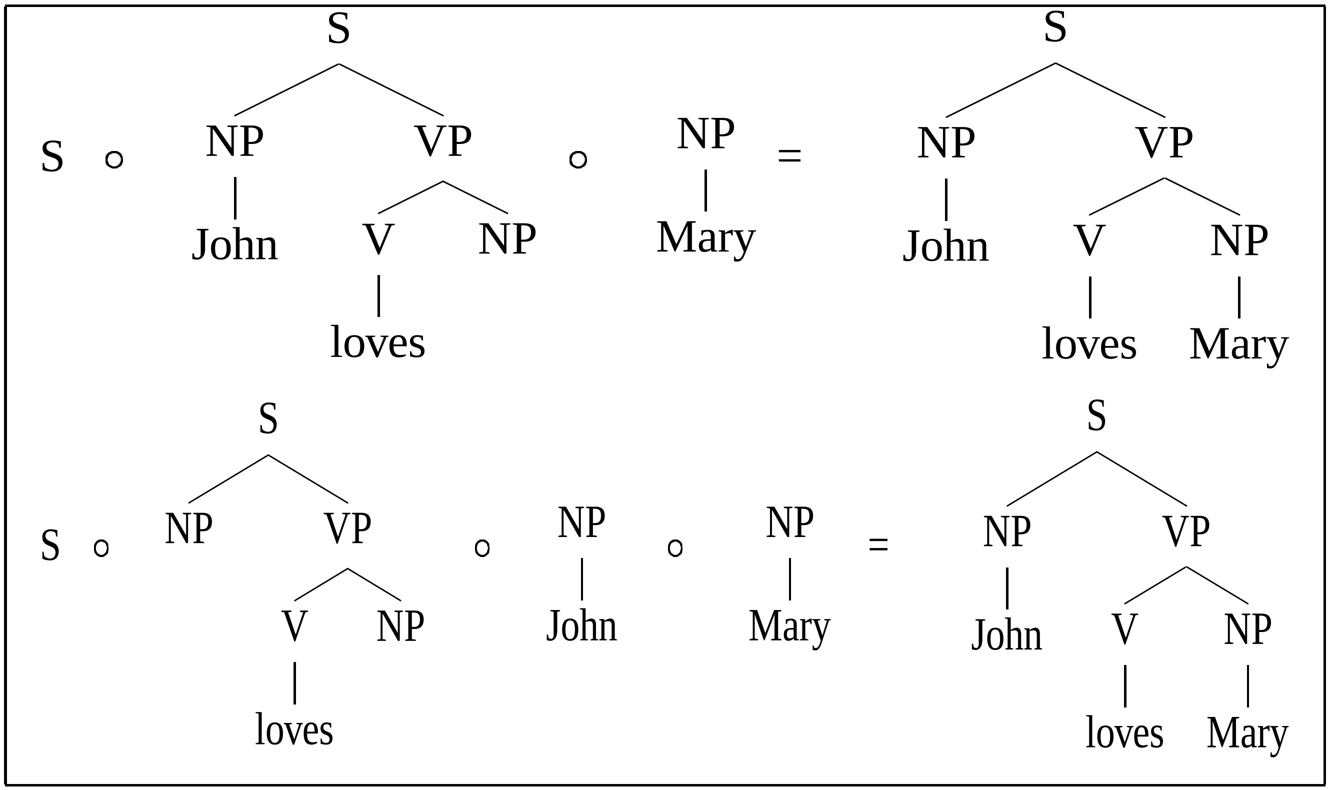
(2) select the most probable parse given the input sentence according to the probabilities of the fragments

A tree-bank stands for a a memory of "fragments" with frequencies

An example



tree-bank



Composing fragments

The DOP Framework (Bod 1995)

A DOP model consists of four elements

Representation: a specification of the form of the parse-trees

Fragments: a specification of the “production units” (rewrite-events),

Composition operation: a specification of the operation for combining rewrite-events

Probability calculation: a specification of how to calculate the probabilities of derivations, parse-trees and sentences from the probabilities of rewrite-events.

This is the specification of all Treebank Grammars!

Instantiation: The DOP1 model (Bod 1992,95,98)

Representations: *Phrase-Structure*

Fragments: *subtrees* – will be defined next

Composition operation: *substitution* (same as in PCFG)

Probability calculation: will be defined next.

DOP1 Training:

Input: a treebank of phrase-structure parse-trees

Output: the set of subtrees, each with a probability

How do we extract, count and employ the subtrees for parsing?

Extracting subtrees and their probabilities

Definitions:

Subtree: a subtree conforms to the following

(1) a connected subgraph of a tree-bank parse-tree

(2) consists of at least one phrase-structure level rule

(3) every internal node dominates all its children or none of them

Subtree probability: simple estimation from the tree-bank by relative-frequency like PCFGs

$$p(t|root(t)) = \frac{freq(t)}{\sum_{t_i:root(t_i)==root(t)} freq(t_i)}$$

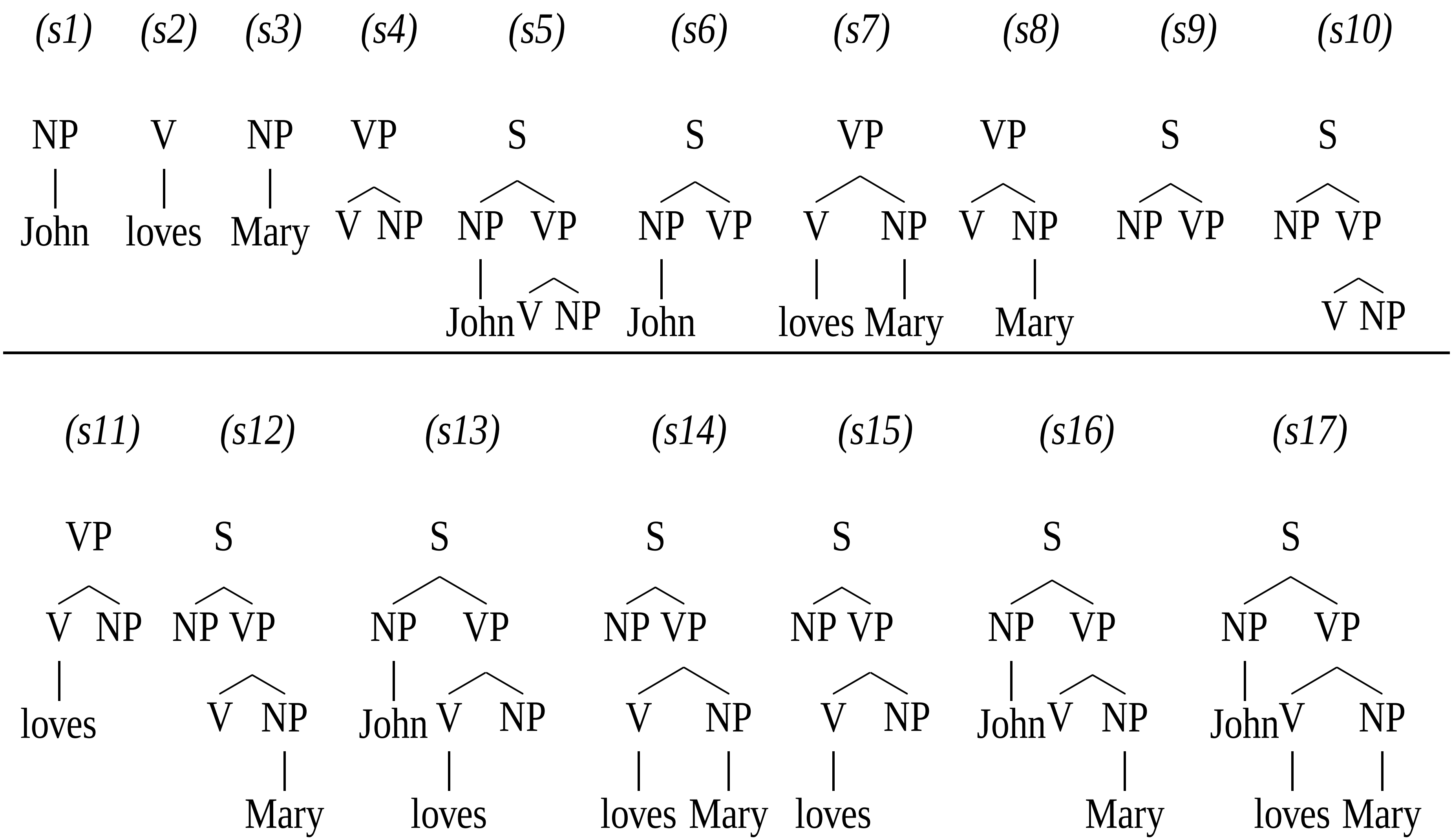


Figure 1: The space of all subtrees

Stochastic Tree-Substitution Grammars (STSG)

The subtrees of DOP are cast into an STSG.

An STSG is a five tuple (like PCFG's are):

Terminals: V_T

Nonterminals: V_N

Start nonterminal: S

Productions: $\mathcal{R}$ is the set of all subtrees

Probability: $P : \mathrm{R} \rightarrow (0,1]$ such that for all $A \in V_N$

$$\sum_{t_i \in \mathcal{R} : root(t_i) == A} P(t_i|A) = 1.0$$

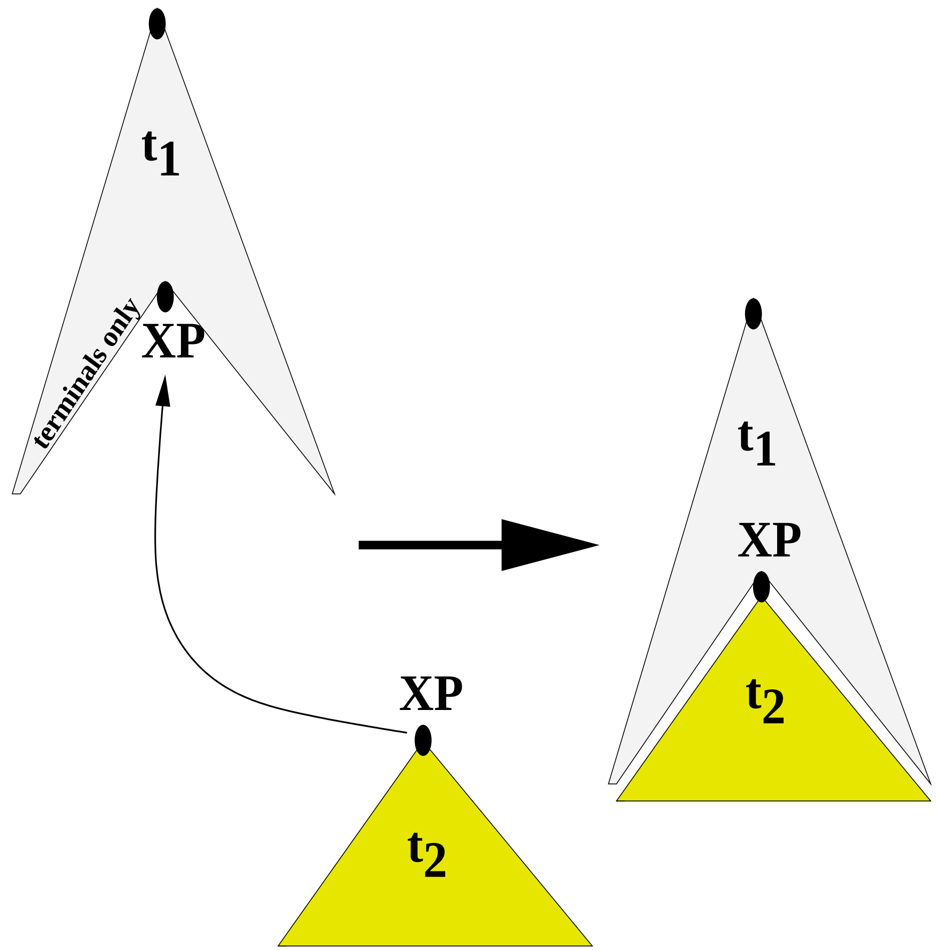
Substitution in STSG

Substition ($\circ$):

$t_1 \circ t_2$ is defined iff:

- t_2 is a subtree
- t_1 is either a subtree or the parse resulting from earlier substitions
- if the root of t_2 is labeled XP, then the left-most nonterminal leaf node in t_1 must be labeled with a nonterminal XP

The result of $t_1 \circ t_2$ is a parse obtained by substituting t_2 for XP.



Derivations and parse-trees in STSG

Definitions:

Derivation: a sequence of one or more substitutions

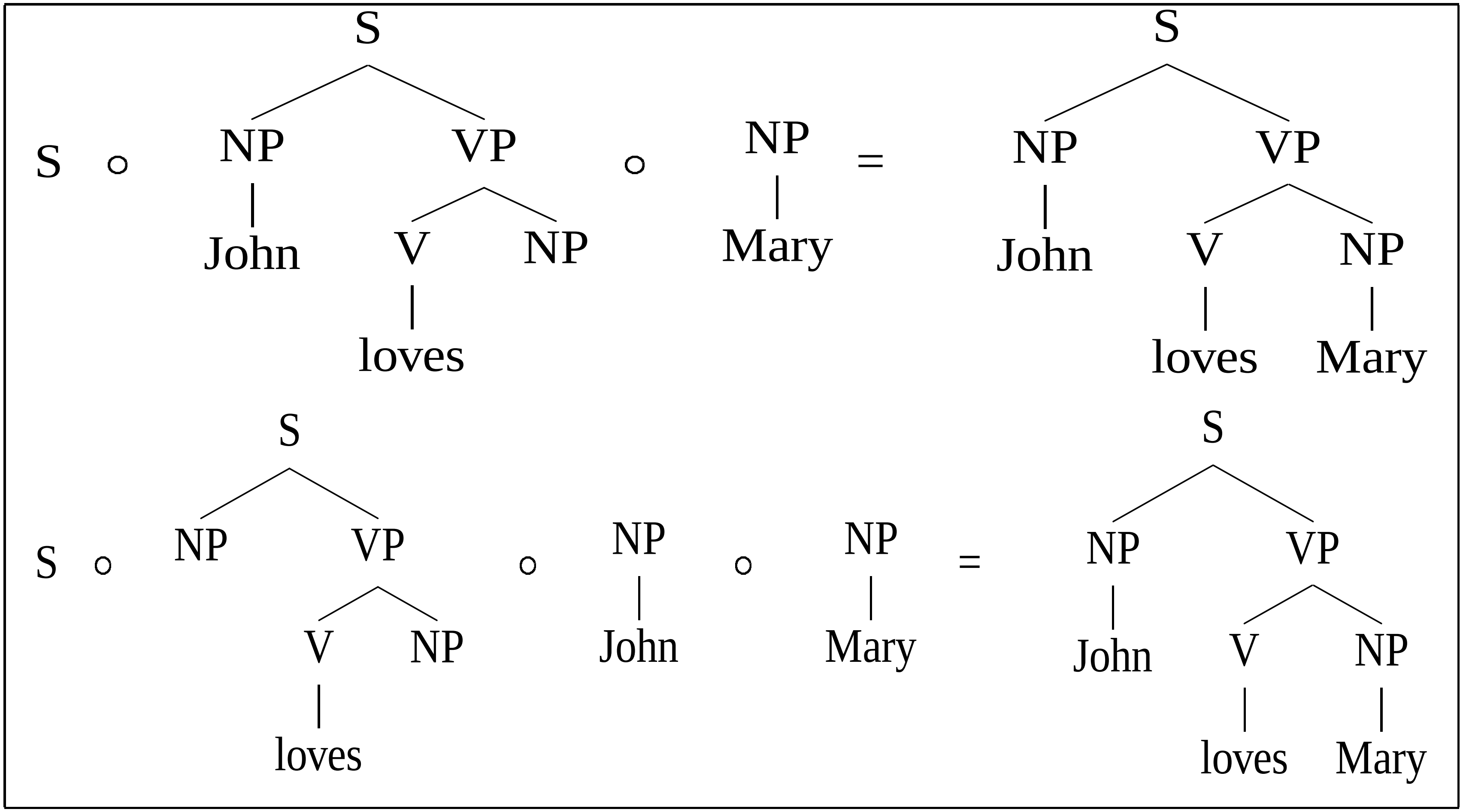
$t_1 \circ t_2 \circ \cdots \circ t_n$ stands for $(\ldots(t_1 \circ t_2) \circ t_3) \cdots \circ t_{i-1}) \ldots)$

Parse-tree: a parse-tree is the tree structure resulting from a derivation

NOTE: unlike PCFG, in DOP a parse can be generated via different derivations!

Intuitively: every derivation stands for a different way for collecting evidence from the tree-bank for the resulting parse-tree

Example: multiple derivations, same parse



Probability calculation: derivations, parses

Given DOP1 model M.

Let $Der_M(T)$ represent the set of derivations that generate parse T in M:

$$P(T|S) = \sum_{der_i \in M} P(der_i, T \mid S) = \sum_{der_i \in Der_M(T)} P(der_i \mid S)$$

$$Suppose\ \forall i : der_i = t_1^i \circ \ldots \circ t_m^i$$

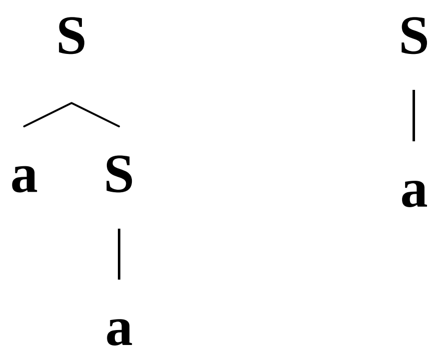
$$P(der_i|S) = \prod_{j=1}^{m} P(t_j^i \mid root(t_j^i))$$

Note similarity to/difference with PCFG!

Example probability calculation

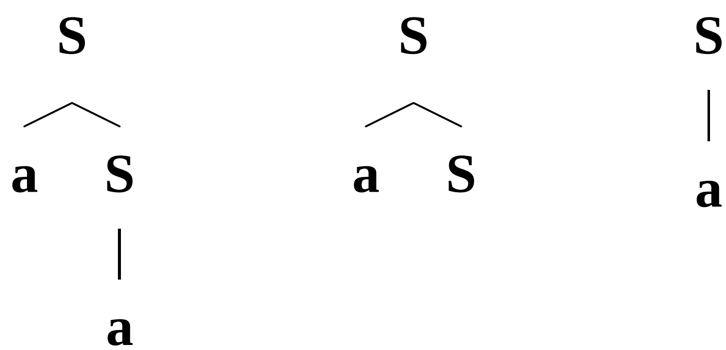
Tree-bank

$Freq(T_1) = 3 \quad Freq(T_2) = 7$



subtrees and probabilities

$P(t_1) = \frac{3}{16} \quad P(t_2) = \frac{3}{16} \quad P(t_3) = \frac{10}{16}$



$$P(T_1) = P(t_1) + P(t_2) * P(t_3)$$

$$P(T_2) = P(t_3)$$

What is in a subtree?

Given subtree t, what does $P(t \mid root(t))$ stand for?

Suppose t consist of the sequence of productions $t = R_0, \ldots, R_m$:

$$\begin{aligned} P(t \mid root(t)) &= P(R_0, \ldots R_m \mid lhs(R_0)) \\ &= P(R_0 \mid lhs(R_0)) \times \prod_{i=1}^{m} P(R_i \mid R_0, \ldots, R_{i-1}) \end{aligned}$$

- Subtree probability stands for exact joint probability of its rules
- A derivation consists of a sequence of subtrees assumed independent from one another

Every derivation of parse-tree T stands for
a different set of independence assumptions in generating T

Parsing under DOP1

Think of probabilistic parsing in two steps:

(1) Parse-forest generation: generate all parses for a DOP model and pack them in a packed parse-forest

PCFG parser: for any input, a DOP1 model obtained from a tree-bank spans the same space of parses as the PCFG obtained from that tree-bank!

(2) Parse selection: compute the probabilities of the parses and select the Most Probable Parse (MPP)

There is no deterministic polynomial-time algorithm for computing the MPP (Sima'an 1996): MPP is NP-Complete.

Algorithms for parse-selection under DOP1

Various algorithms:

Approximate MPP: Monte Carlo sampling from the space of derivations
Stop condition for sampling dependent on expected error.

Other criteria: select the parse

- **MPD:** generated by the Most Probable Derivation (Like PCFG, n^3 time) (Sima'an 1995)
- **LRR:** that maximizes the expected score on Labled Brackets Recall rate (Goodman 1996)

Goodman: adapt selection method to maximize score on evaluation metric!

Why is DOP interesting? Pros

From different point of views:

Theory: Mind provoking as it goes beyond competence models!
A new research agenda, with its own theoretical problems and challenges.

Formal power: more powerfull than PCFGs
There exist STSGs that capture distributions that PCFGs cannot capture!

Engineering: feature selection is not the essence, rather an optimization tool for DOP.

Problems of DOP1 and extensions

Problems of DOP1:

- Hard estimation of subtree probabilities: Subtree relative frequency is not Maximum-Likelihood (see Buratto and Sima'an 2003)!
- MPP is too expensive, MPD too weak!
- **DOP1 uses only weak lexicalization** (Sima'an 2000)

More robust DOP models: Tree-gram model (Sima'an 2000)

Further: incorporation of dependency probabilities.

Intermezzo: A Commercial

Data-Oriented Parsing

R. Bod, R. Scha and K. Sima'an (eds.)
CSLI Publishers, 2003.

Consists of 21 papers (by 24 researchers)
Covers a wide range of work on treebank parsing and DOP.