Unifying Linguistic, Musical and Visual Processing

Rens Bod

ILLC, University of Amsterdam
School of Computing, University of Leeds

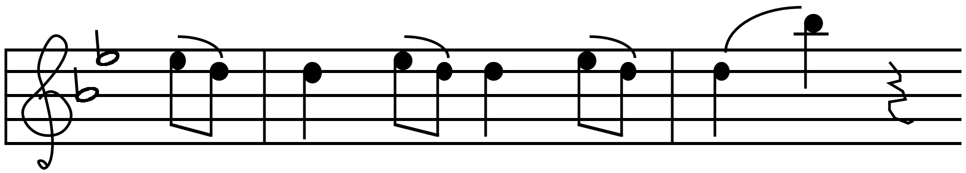
What do Language, Music and Image have in common?

E.g.:

Language:

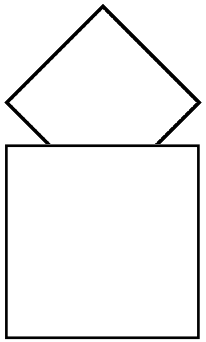
"List the sales of products in 2003"

Music:



...

Image:



At first sight very little...

How do we perceive Language, Music and Image?

Inherent to all forms of perception:

A *structuring process* in *groups*, *subgroups*, *sub-subgroups*, etc.

It is virtually impossible *not* to perceive structure

(People even assign structure to noise...)

How do we perceive Language, Music and Image?

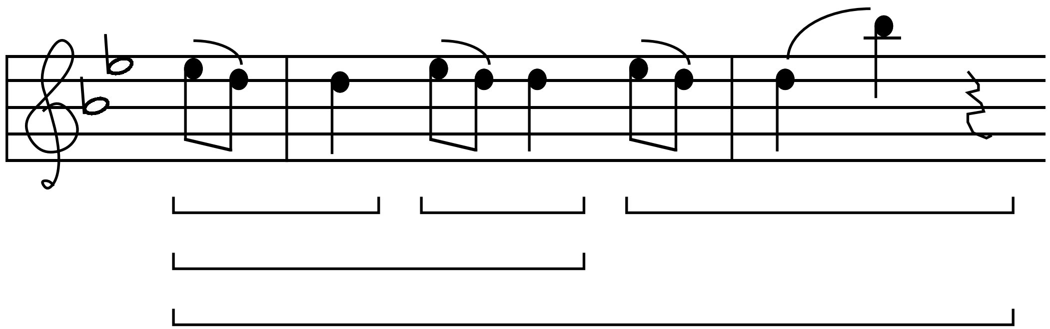
Inherent to all forms of perception:

A *structuring process* in *groups*, *subgroups*, *sub-subgroups*, etc.

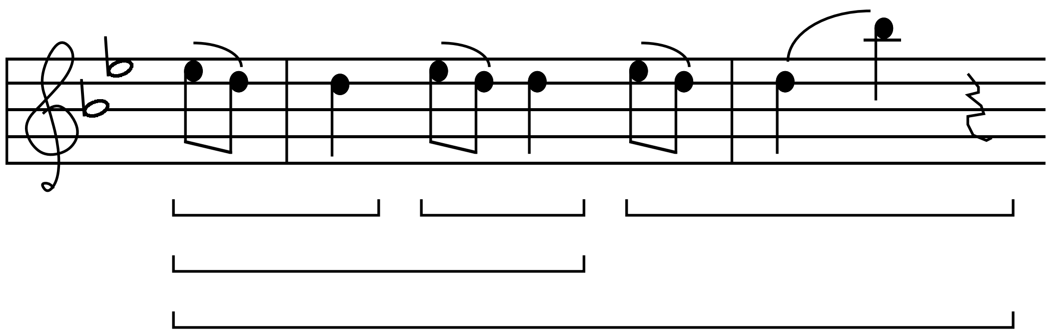
It is virtually impossible *not* to perceive structure

(People even assign structure to noise...)

In music, grouping structure is typically respresented as:



Grouping Structure in Music



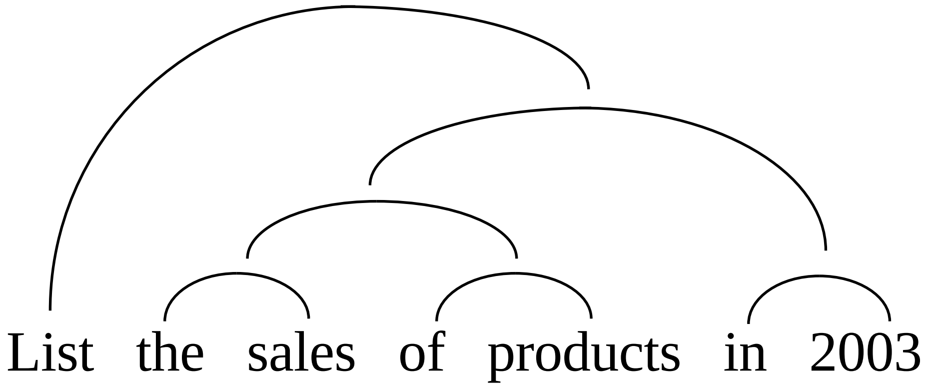
The musical piece as a whole forms a *group*

A *group* consists of *subgroups* which are recursively built up out of smaller *subgroups*, up to the smallest unit (e.g. a pitch)

Grouping structure represents how *parts* combine into a *whole*

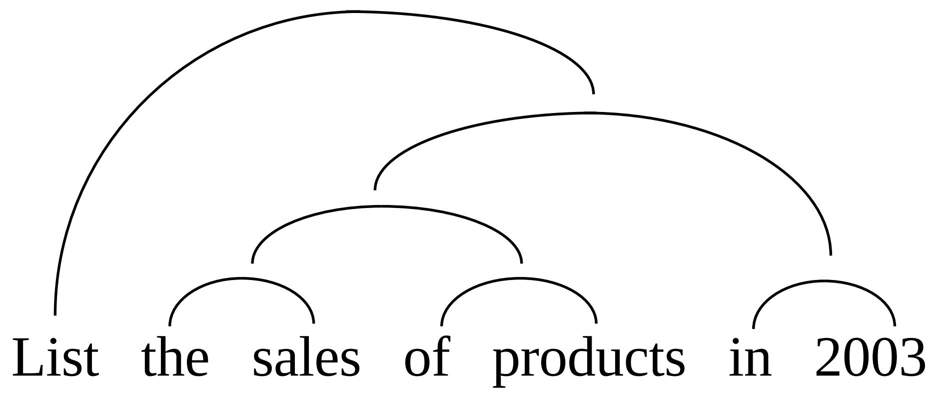
Grouping Structure in Language

Groups in language form a *tree structure* (Wundt 1880):

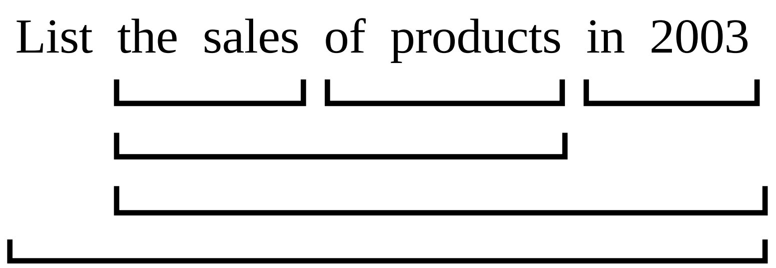


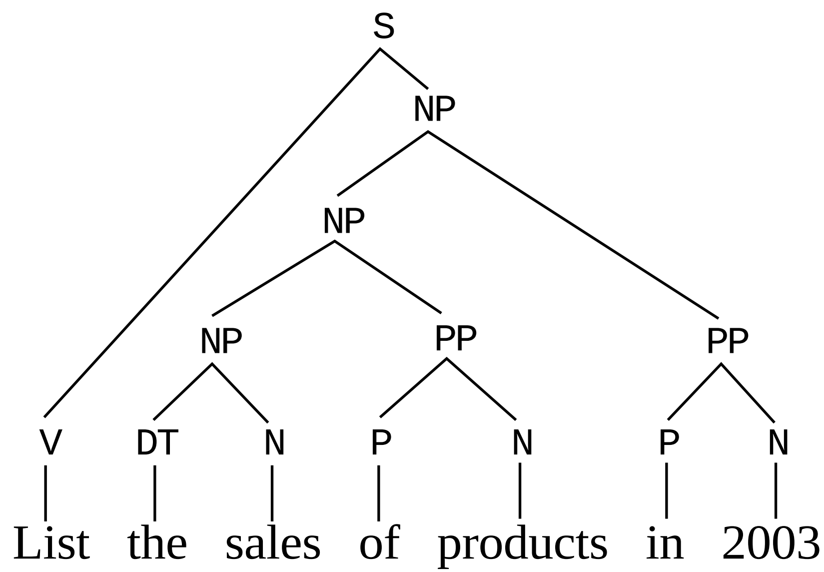
Grouping Structure in Language

Groups in language form a *tree structure* (Wundt 1880):

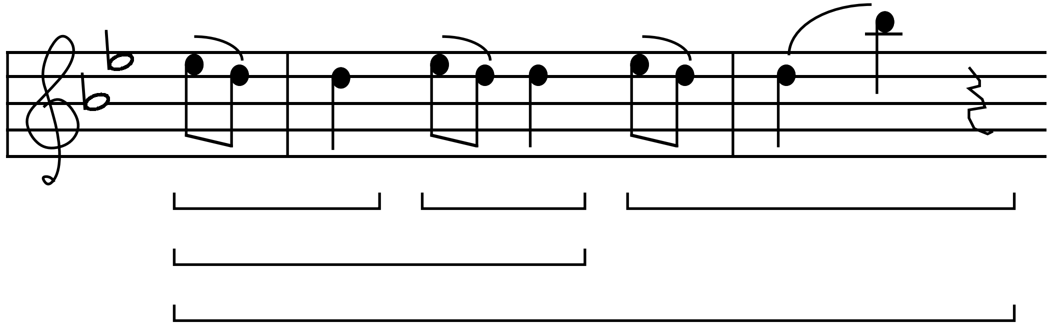


Grouping structure in different representations (Chomsky 1956):

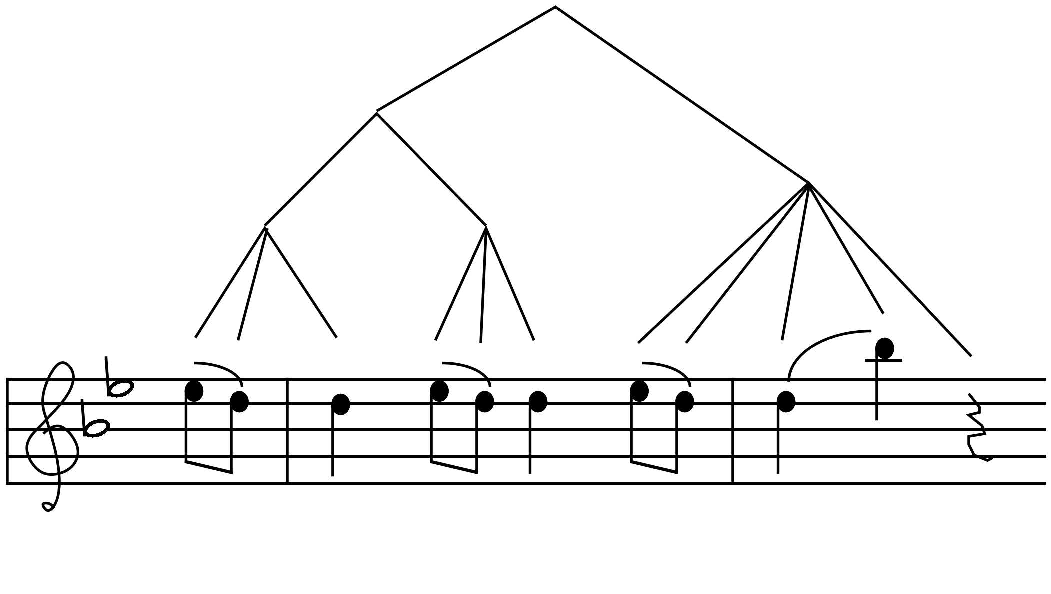




Grouping Structure = Tree Structure

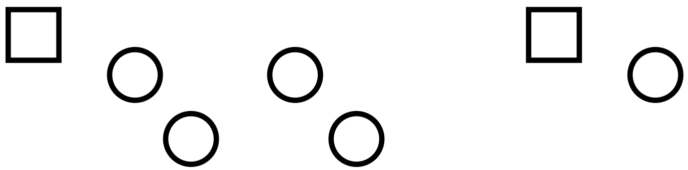


is equivalent (isomorphic) with:

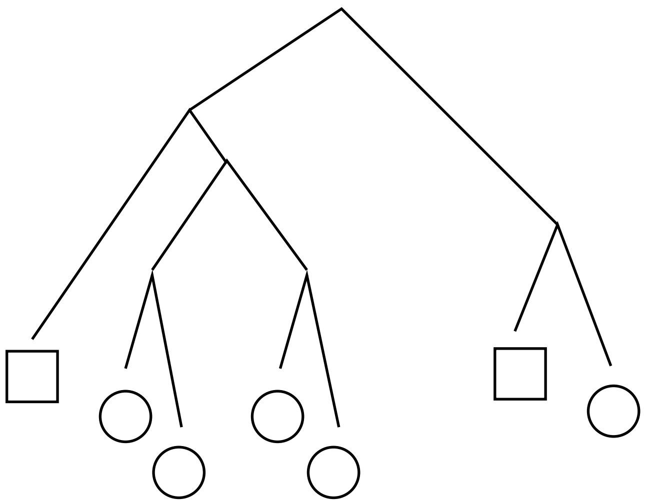


Also Visual Groups form a Tree Structure

According to Wertheimer (1923) the visual input

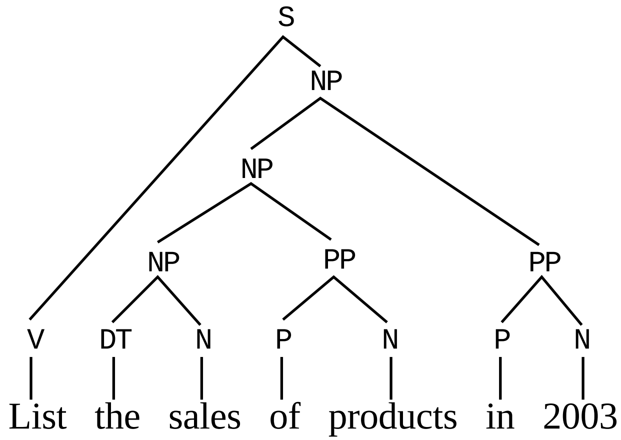


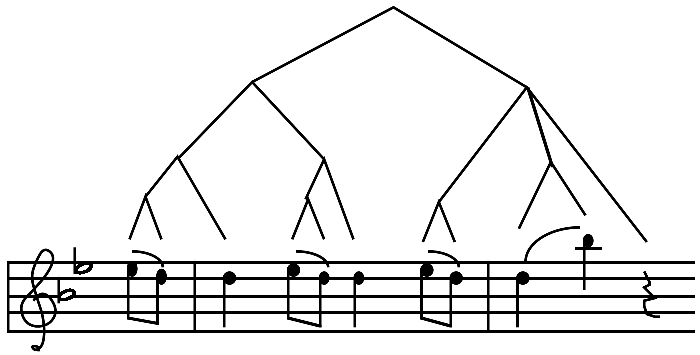
is assigned the following structure:

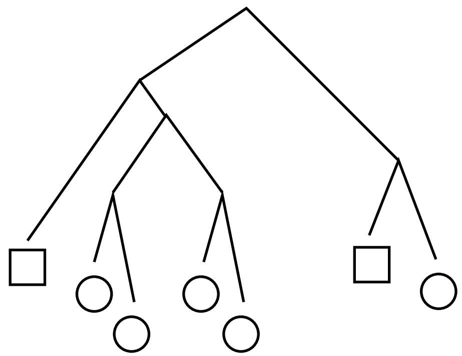


Perceptual structuring forms the link between low-level segmentation and higher-level interpretation algorithms

Perceptual Structure = Tree Structure



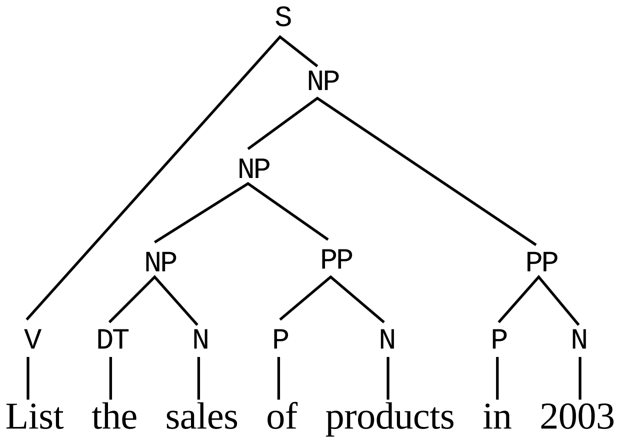


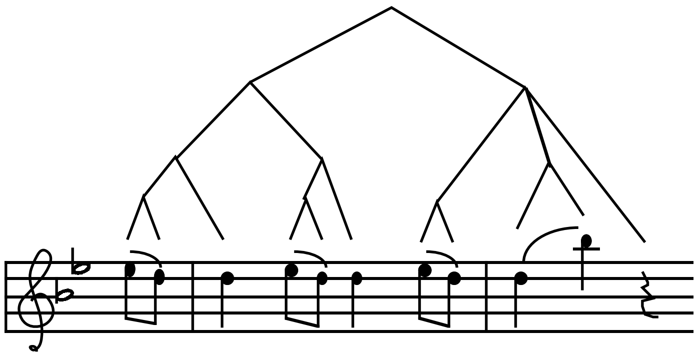


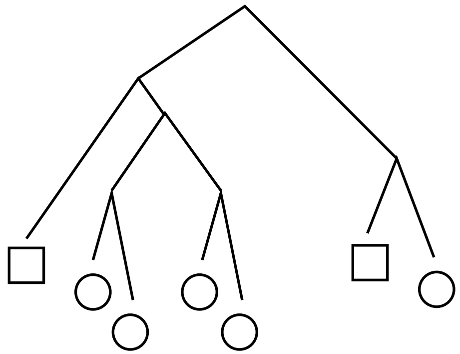
Relatively Uncontroversial:

There exists *one representation* for structural perception for all modalities

Perceptual Structure = Tree Structure







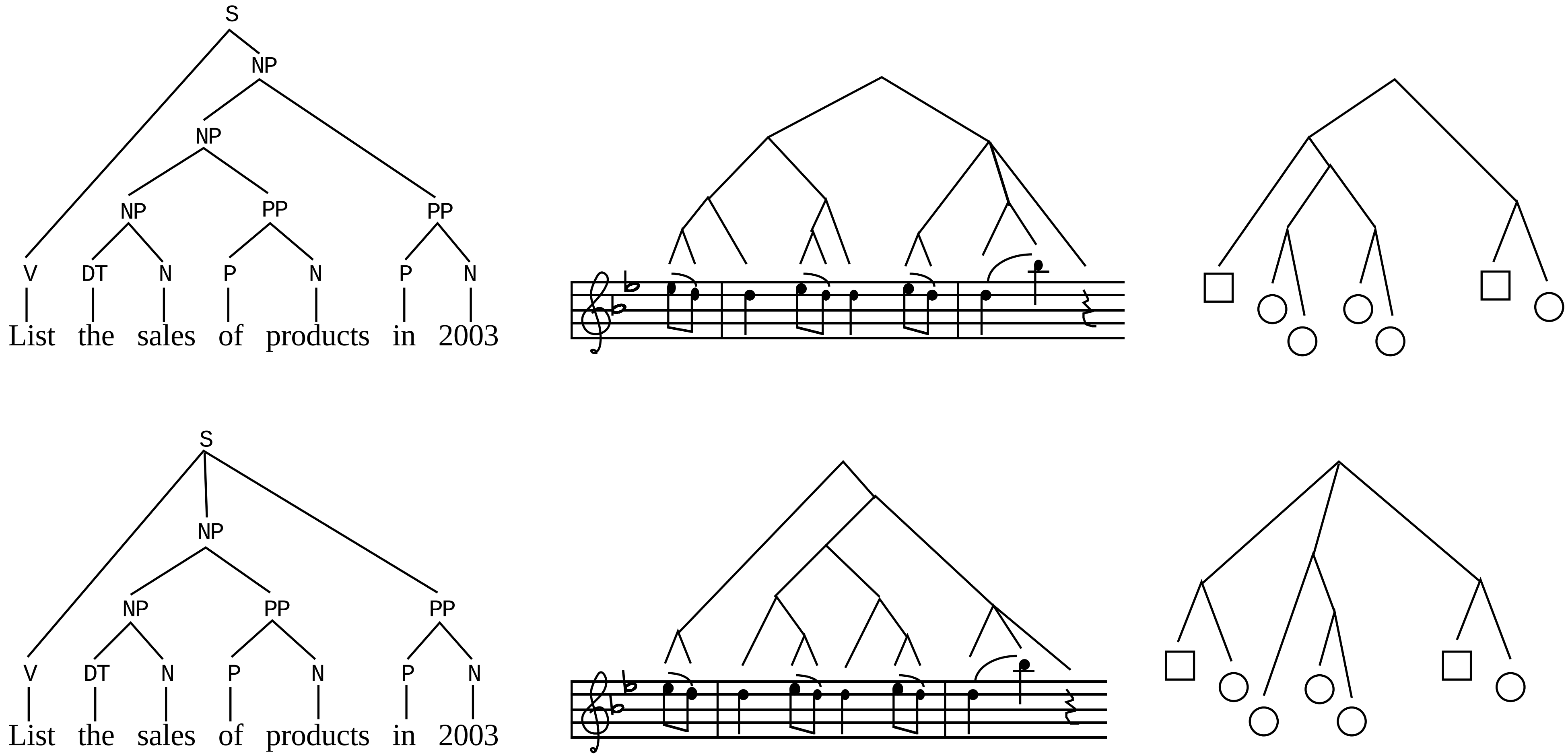
Relatively Uncontroversial:

There exists *one representation* for structural perception for all modalities

Very Controversial:

There exists *one model* that predicts the perceived structure in *language, music* en *vision*

Additional Problem: Perception is Ambiguous



The same input can be assigned several structures: ambiguity

Ambiguity is not just a problem

Average sentence from *Wall Street Journal* has more than **one million** different *possible* tree structures (Charniak 1999)

Adding semantics makes the problem even worse!

Ambiguity is not just a problem

Average sentence from *Wall Street Journal* has more than **one million** different *possible* tree structures (Charniak 1999)

Adding semantics makes the problem even worse!

"Any given sequence of notes is infinitely ambiguous, but this ambiguity is seldom apparent to the listener" (Longuet-Higgins 1987)

Humans perceive mostly just *one* grouping structure

Ambiguity is not just a problem

Average sentence from *Wall Street Journal* has more than **one million** different *possible* tree structures (Charniak 1999)

Adding semantics makes the problem even worse!

"Any given sequence of notes is infinitely ambiguous, but this ambiguity is seldom apparent to the listener" (Longuet-Higgins 1987)

Humans perceive mostly just *one* grouping structure

> 96% agreement among subjects (language users)

Language: *Penn Treebank*
Music: *Essen Folksong Collection*
Vision: *Nijmegen Visual Database*

Historically, two competing principles for solving ambiguity in perception

1. **Simplicity Principle** (Wertheimer 1923...Leeuwenberg 2001, Chater 2003)

   Preference for the *simplest* structure

2. **Likelihood Principle** (Helmholtz 1910...Suppes 1984, Charniak 2001)

   Preference for the *most likely* structure

Can these principles still inspire us?

The Dual Nature of Perception

These principles each play a *different* role in perception:

Simplicity: general preference for "economy", "least effort", "shortest derivation"

Likelihood: a memory-based bias due to previous experiences

The Dual Nature of Perception

These principles each play a *different* role in perception:

Simplicity: general preference for "economy", "least effort", "shortest derivation"

Likelihood: a memory-based bias due to previous experiences

Hypothesis: perceptual systen strives for the *simplest* structure but in doing so it is influenced by the *likelihood* of previous structures

Possible Measures for Simplicity and Likelihood

Simplicity: *number* of "steps" to generate a tree structure

Likelihood: joint *probability* of the steps to generate a tree structure

We can compute this if we have a large, representative collection of tree structures for each modality (a "corpus")

Possible Measures for Simplicity and Likelihood

Simplicity: *number* of "steps" to generate a tree structure

Likelihood: joint *probability* of the steps to generate a tree structure

We can compute this if we have a large, representative collection of tree structures for each modality (a "corpus")

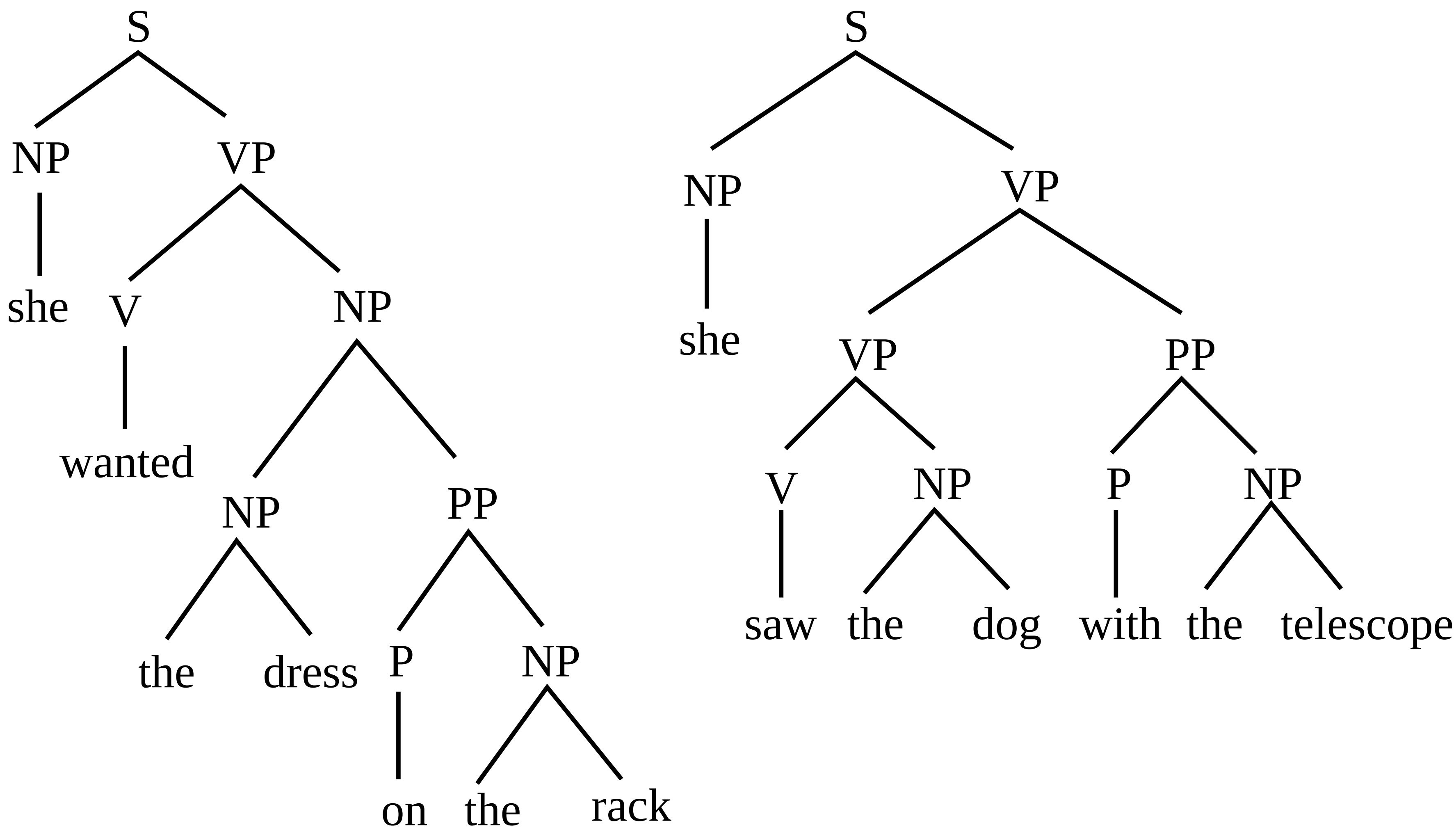
***Data-Oriented Parsing model (DOP)*:**

New input is analyzed and interpreted out of parts of previously perceived input

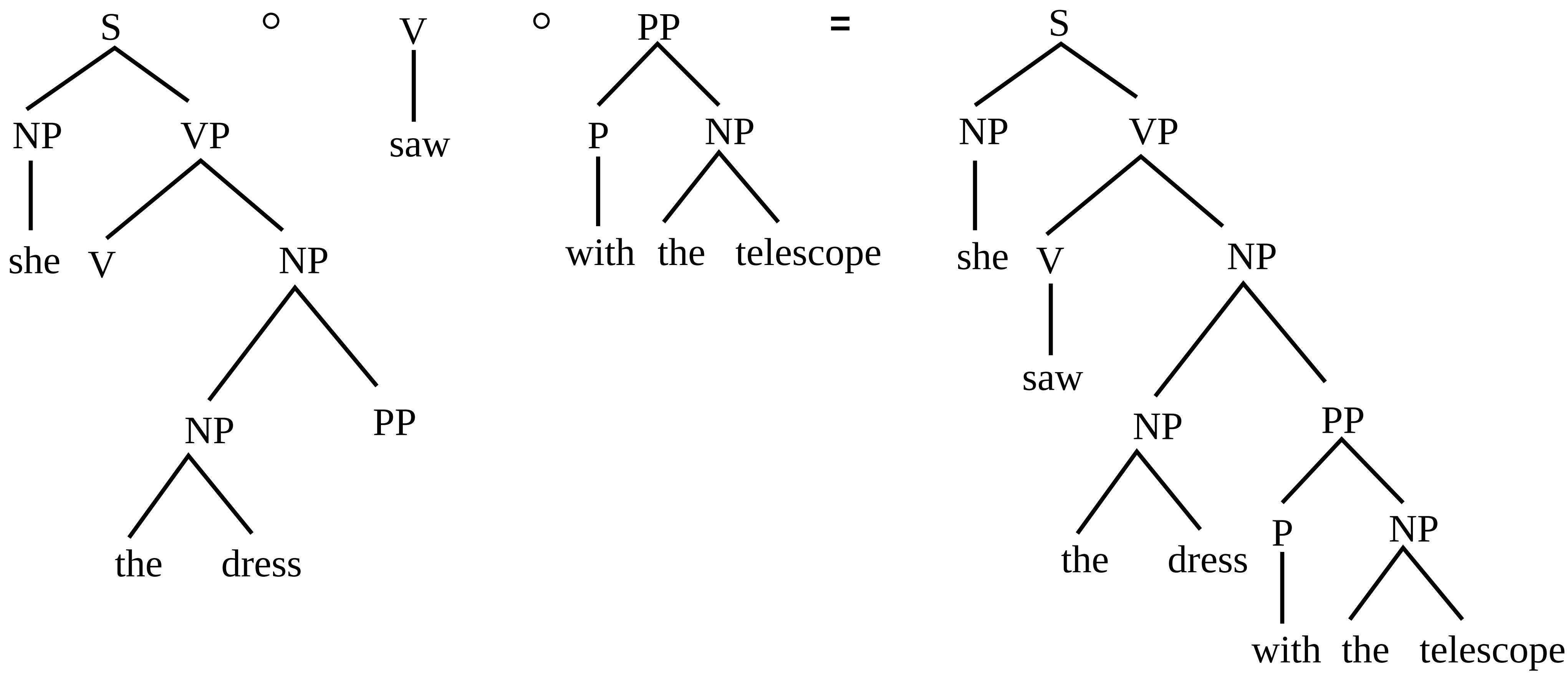
(cf. CBR, Corpus-based NLP, EBL, ...)

Example of a DOP model for Language

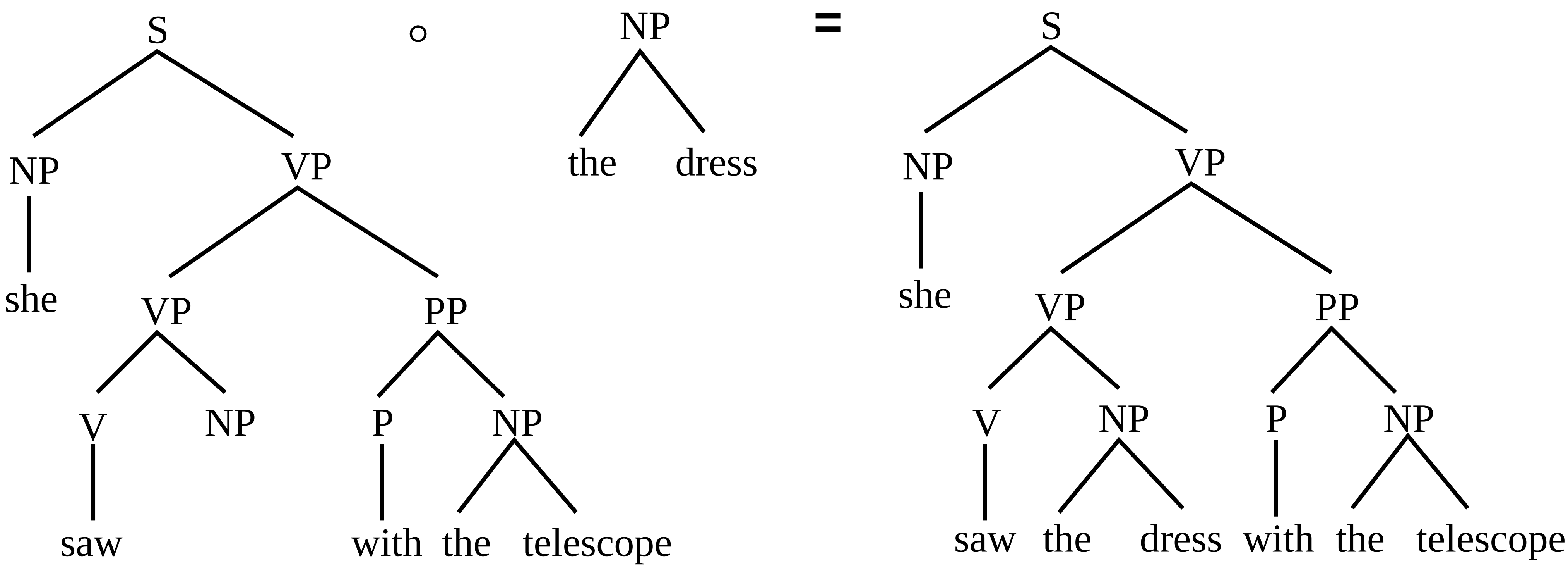
Let's start with an extremely simple corpus:



A new sentence such as "*She saw the dress with the telescope*" is analyzed by **combining subtrees from the corpus**



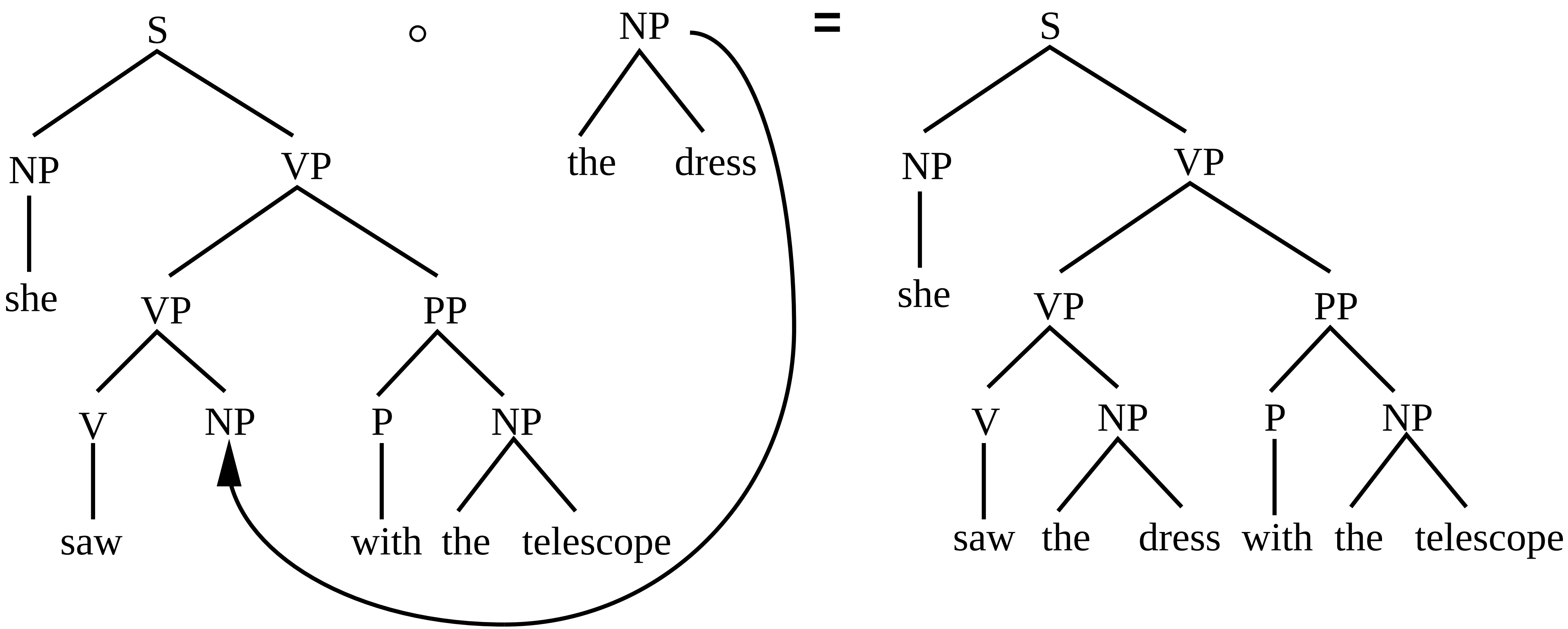
But there is also a "competing" analysis:



This analysis consists of two steps, and is therefore preferred according to the *simplicity principle*: **maximal similarity** with corpus.

But it is **not** preferred according to the *likelihood principle*!

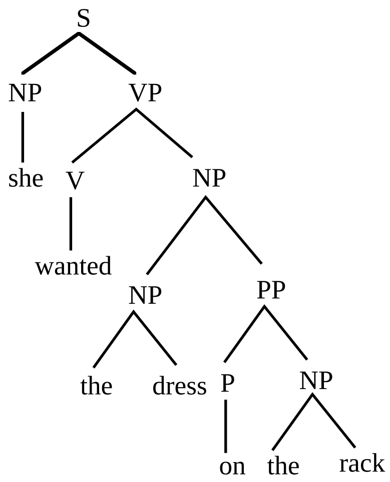
But there is also a "competing" analysis:

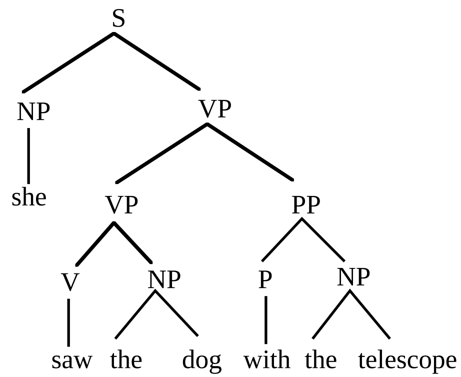


This analysis consists of two steps, and is therefore preferred according to the *simplicity principle*: **maximal similarity** with corpus.

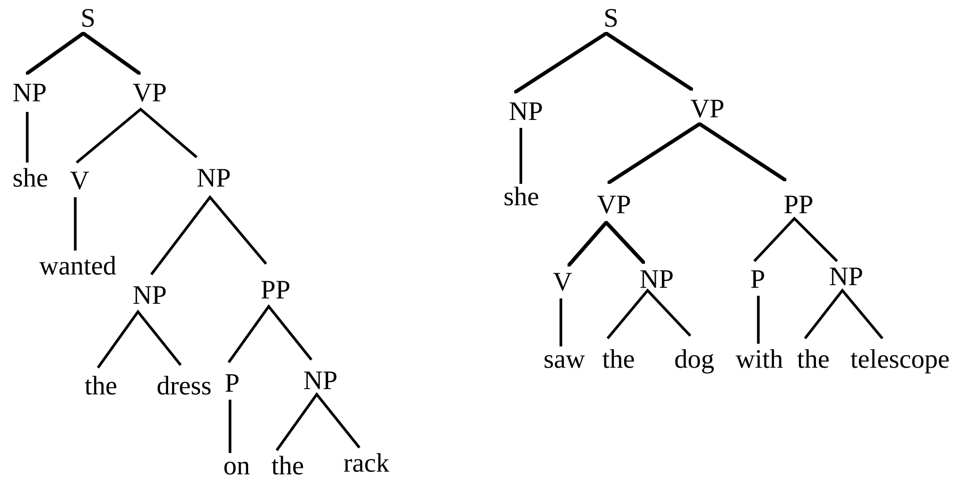
But it is **not** preferred according to the *likelihood principle*!

Corpus

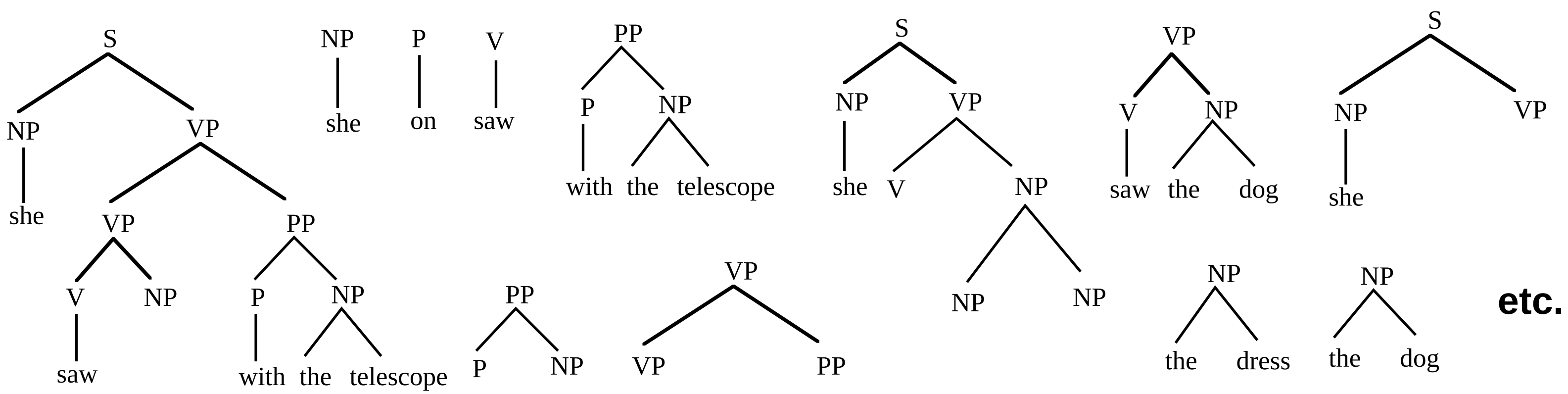

S
NP
VP
she
V
NP
wanted
NP
PP
the
dress
P
NP
on
the
rack


S
NP
VP
she
VP
PP
V
NP
P
NP
saw
the
dog
with
the
telescope

Corpus

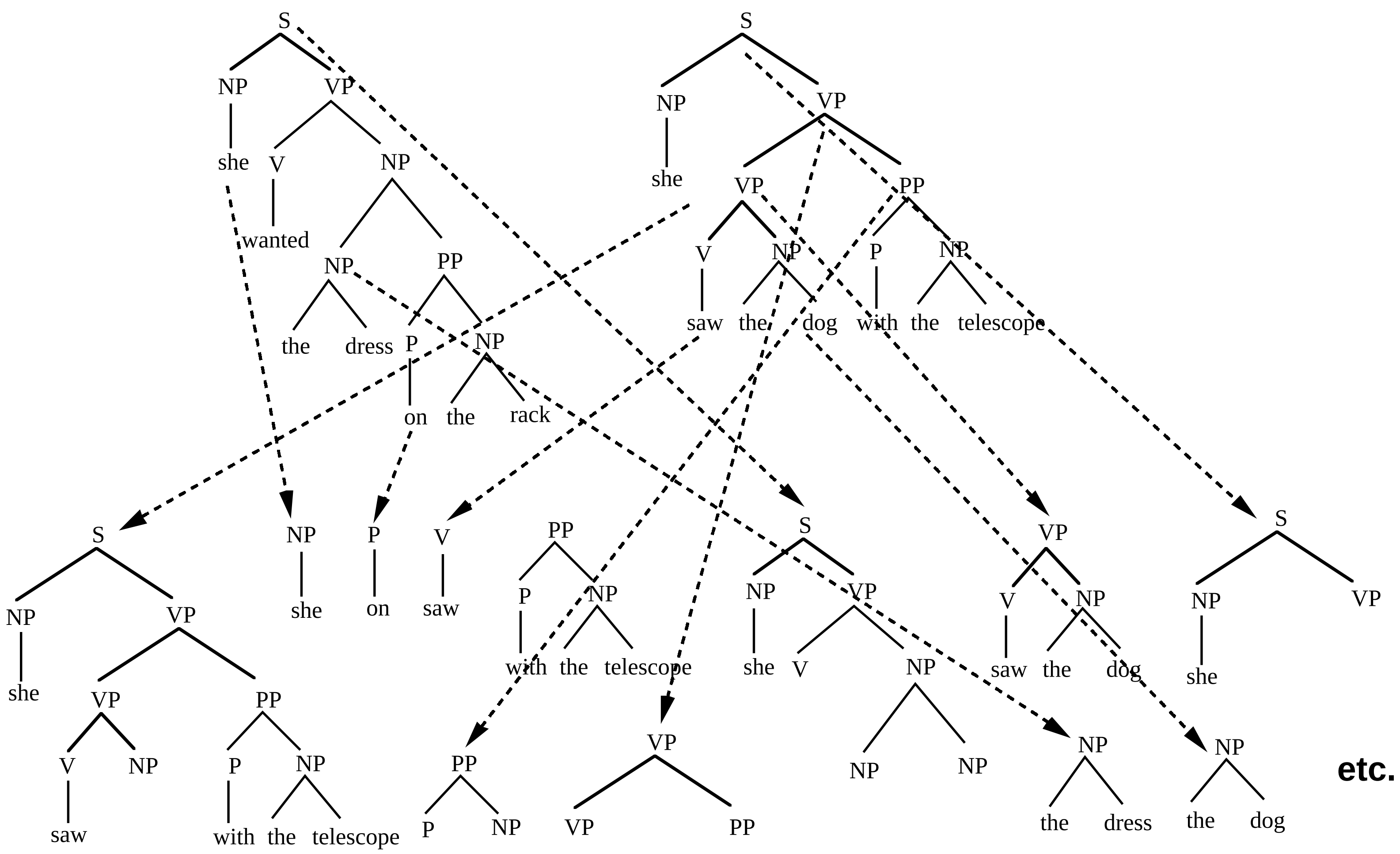


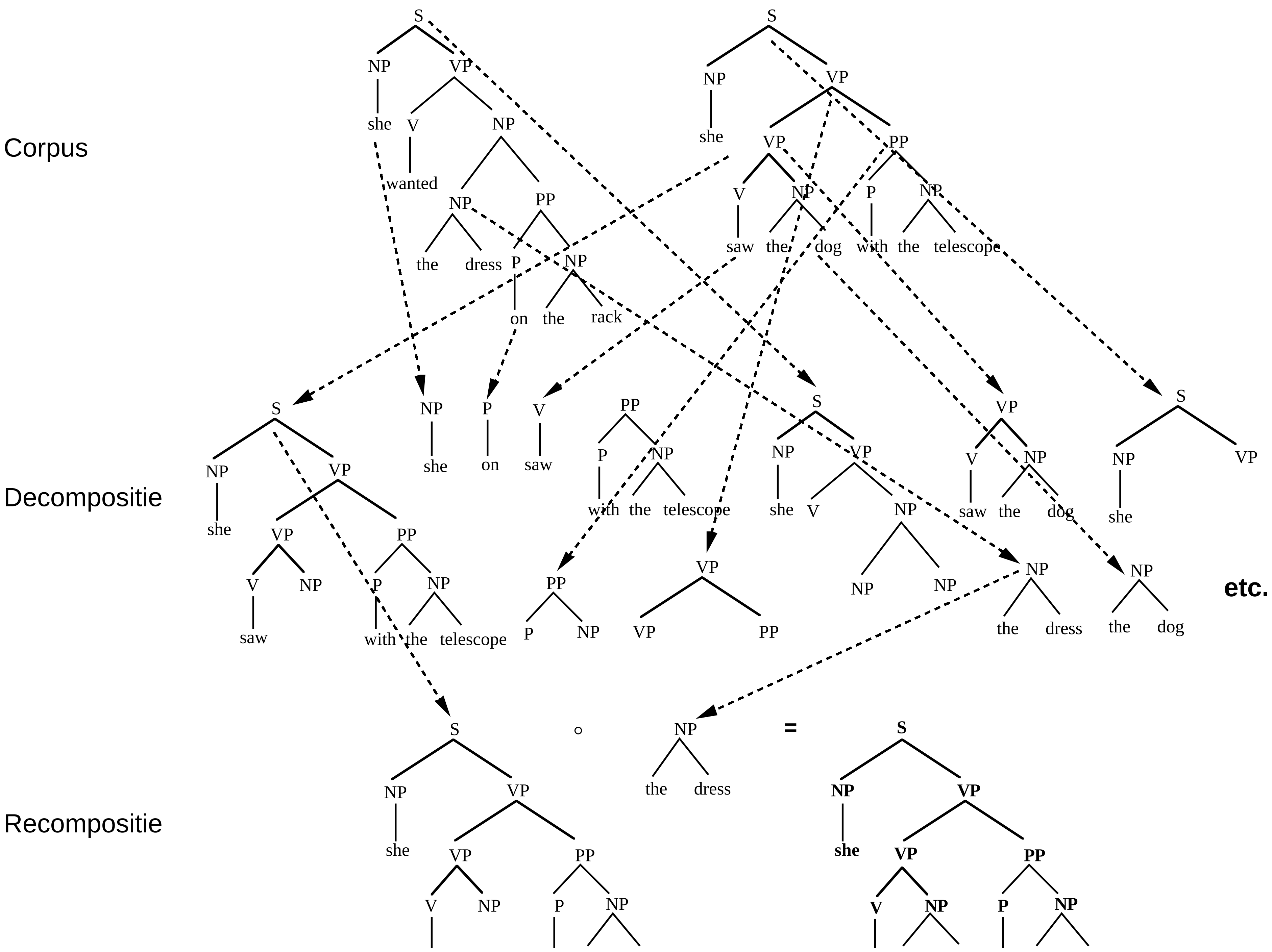
Decompositie



Corpus

Decompositie




Corpus
Decompositie
Recompositie
etc.

DOP models are Stochastic Tree Grammars

By putting various constraints on STGs, we can instantiate:

- stochastic context-free grammars
- stochastic head-lexicalized grammars
- stochastic tree-adjoining grammars
- stochastic finite-state grammars

  etc...

We will focus on STSGs (Stochastic Tree Subsitution Grammars)

DOP models are Stochastic Tree Grammars

By putting various constraints on STGs, we can instantiate:

- stochastic context-free grammars
- stochastic head-lexicalized grammars
- stochastic tree-adjoining grammars
- stochastic finite-state grammars

  etc...

We will focus on STSGs (Stochastic Tree Subsitution Grammars)

However, we have also developed DOP models for richer structures, such as *LFG*, *HPSG*, *Logical-Semantic* and *Discourse annotations*

(e.g. Bod & Kaplan 1998, 2003; Way 2003; Neumann 2003; Bod et al. 1996; Bod 1998)

Experiments with large corpora

Penn Treebank, Essen Folksong Collection:

Tens of thousands of analyzed sentences and folksongs

Experiments with large corpora

Penn Treebank, Essen Folksong Collection:

Tens of thousands of analyzed sentences and folksongs

Simplest tree structure for string *s*:

Minimize number N of corpus subtrees in tree T

$$T_{best} = \arg\min_T N(T \mid s)$$

Experiments with large corpora

Penn Treebank, Essen Folksong Collection:

Tens of thousands of analyzed sentences and folksongs

Simplest tree structure for string *s*:

Minimize number N of corpus subtrees in tree T

$$T_{best} = \arg\min_T N(T \mid s)$$

Likeliest tree structure for string *s*:

Maximize product of relative frequencies of subtrees t_i in T

$$T_{best} = \arg\max_T P(T \mid s) = \arg\max_{<t_1 \ldots t_n>} \prod_i P(t_i \mid s)$$

Experiments with large corpora

Penn Treebank, Essen Folksong Collection:

Tens of thousands of analyzed sentences and folksongs

Simplest tree structure for string *s*:

Minimize number N of corpus subtrees in tree T

$$T_{best} = \arg\min_T N(T \mid s)$$

Likeliest tree structure for string *s*:

Maximize product of relative frequencies of subtrees t_i in T

$$T_{best} = \arg\max_T P(T \mid s) = \arg\max_{<t_1 \ldots t_n>} \prod_i P(t_i \mid s)$$

Our best hypothesis so far:

The perceptual system selects the *simplest* structure from the top of the distribution of *most probable* structures

The probability of:

a *subtree* t : $$P(t) = \frac{|t|}{\sum_{t' :\ root(t')=root(t)} |t'|}$$

a *derivation* $d = t_1 \circ ... \circ t_n$: $$P(t_1 \circ ... \circ t_n) = \prod_i P(t_i)$$

a *parse tree* T : $$P(T) = \sum_d \prod_i P(t_{id})$$

where t_{id} is the i-th subtree in derivation d that produces T

Computational Aspects of DOP

Problem: exponentially many subtrees in DOP / STSG

Can be solved by reducing DOP to an isomorphic *Probabilistic Context-Free Grammar* or *PCFG*

Computational Aspects of DOP

Problem: exponentially many subtrees in DOP / STSG

Can be solved by reducing DOP to an isomorphic *Probabilistic Context-Free Grammar* or *PCFG*

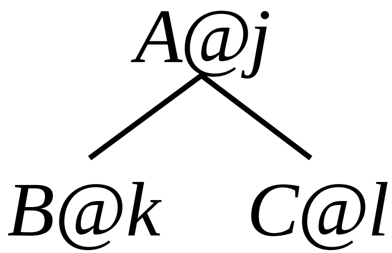
Every node in every tree in corpus is assigned a unique number:

$A@k$ denotes node at address k where A is nonterminal of that node

A new nonterminal is created for each node in the training data: A_k

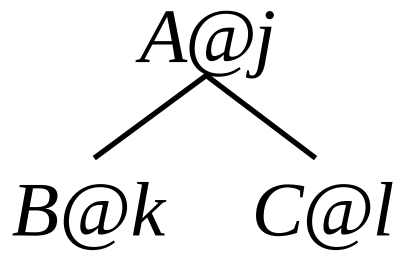
Sketch of PCFG reduction of DOP (1)

Consider a node A@j of the following form in STSG/DOP:



Sketch of PCFG reduction of DOP (1)

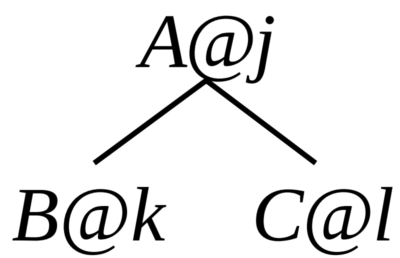
Consider a node A@j of the following form in STSG/DOP:



There are b_k non-trivial subtrees headed by *B@k* plus trivial case where left node is simply *B*.

Sketch of PCFG reduction of DOP (1)

Consider a node A@j of the following form in STSG/DOP:



There are b_k non-trivial subtrees headed by *B@k* plus trivial case where left node is simply *B*.

Thus $b_k + 1$ different possibilities on the left branch

Similarly, $c_l + 1$ possibilities on the right branch

Thus, $a_j = (b_k + 1)(c_l + 1)$ possible subtrees headed by *A@j*

Sketch of PCFG reduction of DOP (2)

There is a PCFG with the following property (Bod 2003; Goodman 2003):

for every subtree in training corpus headed by *A*, the PCFG will generate an isomorphic subderivation with probability 1/*a*

$A_j \rightarrow BC$	$(1/a_j)$	$A \rightarrow BC$	$(1/a)$
$A_j \rightarrow B_k C$	(b_k/a_j)	$A \rightarrow B_k C$	(b_k/a)
$A_j \rightarrow B C_l$	(c_l/a_j)	$A \rightarrow B C_l$	(c_l/a)
$A_j \rightarrow B_k C_l$	$(b_k c_l/a_j)$	$A \rightarrow B_k C_l$	$(b_k c_l/a)$

Sketch of PCFG reduction of DOP (2)

There is a PCFG with the following property (Bod 2003; Goodman 2003):

for every subtree in training corpus headed by *A*, the PCFG will generate an isomorphic subderivation with probability 1/*a*

$A_j \rightarrow BC$	$(1/a_j)$	$A \rightarrow BC$	$(1/a)$
$A_j \rightarrow B_kC$	(b_k/a_j)	$A \rightarrow B_kC$	(b_k/a)
$A_j \rightarrow BC_l$	(c_l/a_j)	$A \rightarrow BC_l$	(c_l/a)
$A_j \rightarrow B_kC_l$	(b_kc_l/a_j)	$A \rightarrow B_kC_l$	(b_kc_l/a)

Rather than using all subtrees, we can use a "compact" PCFG !

Sketch of PCFG reduction of DOP (3)

- Dynamic programming algorithm known as *Viterbi bottom-up search* computes *most probable derivation* for input string

- Same algorithm can be used to compute *shortest derivation* (i.e. simplest tree) by assigning each subtree equal probability

Sketch of PCFG reduction of DOP (3)

- Dynamic programming algorithm known as *Viterbi bottom-up search* computes *most probable derivation* for input string

- Same algorithm can be used to compute *shortest derivation* (i.e. simplest tree) by assigning each subtree equal probability

  Thus both *likeliest* and *simplest* tree are efficiently computed

  Next, we can compute the simplest among the *n* likeliest trees also by *Viterbi n best search*

Sketch of PCFG reduction of DOP (3)

- Dynamic programming algorithm known as *Viterbi bottom-up search* computes *most probable derivation* for input string

- Same algorithm can be used to compute *shortest derivation* (i.e. simplest tree) by assigning each subtree equal probability

  Thus both *likeliest* and *simplest* tree are efficiently computed

  Next, we can compute the simplest among the *n* likeliest trees also by *Viterbi n best search*

- Other work has proposed different computational solutions:

  Voted Perceptron (Collins), *Tree Kernels* (Bod, Duffy), *MaxEnt* (Sima'an), *MDL* (Bonnema), *E-M* (Prescher)...

Test Domains

- ***Linguistic test domain:***

  Wall Street Journal (WSJ) corpus in the Penn Treebank:
  50.000 manually analyzed sentences

Test Domains

- ***Linguistic test domain:***

  Wall Street Journal (WSJ) corpus in the Penn Treebank: 50.000 manually analyzed sentences

- ***Musical test domain:***

  Essen Folksong Collection (EFC): 20.150 melodically analyzed western folksongs:

  - *Pitches*: numbers from 1 to 7
  - *Duration indicators*: underscore (_) or a period (.) *after* the numbers
  - *Octave position*: plus and minus signs (+,-) *before* the numbers
  - *Chromatic alterations*: "#" or "b" *after* the numbers
  - *Pauses*: 0, possibly followed by duration indicators

Test Domains

- ***Linguistic test domain:***

  Wall Street Journal (WSJ) corpus in the Penn Treebank: 50.000 manually analyzed sentences : *the* benchmark in NLP

- ***Musical test domain:***

  Essen Folksong Collection (EFC): 20.150 melodically analyzed western folksongs:

  - *Pitches*: numbers from 1 to 7
  - *Duration indicators*: underscore (_) or a period (.) *after* the numbers
  - *Octave position*: plus and minus signs (+,-) *before* the numbers
  - *Chromatic alterations*: "#" or "b" *after* the numbers
  - *Pauses*: 0, possibly followed by duration indicators

- ***Visual test domain*: see later**

Example from Essen Folksong Collection

#4551: *Schneckhaus Schneckhaus stecke deine Hörner aus*

(German children song)

5_3_5_3_1234553_1234553_12345_3_12345_3_553_553_553_65432_1_

Example from Essen Folksong Collection

#4551: *Schneckhaus Schneckhaus stecke deine Hörner aus*

(German children song)

5_3_5_3_1234553_1234553_12345_3_12345_3_553_553_553_65432_1_

Grouping structure according to Essen Folksong collection:

((5_3_5_3_) (1234553_) (1234553_) (12345_3_) (12345_3_) (553_553_)
(553_65432_1_))

NB: linguistic phrase structure does not predict musical phrase structure!

Preprocessing the Essen Folksong Annotations

- We automatically added three basic labels to the phrase structures:

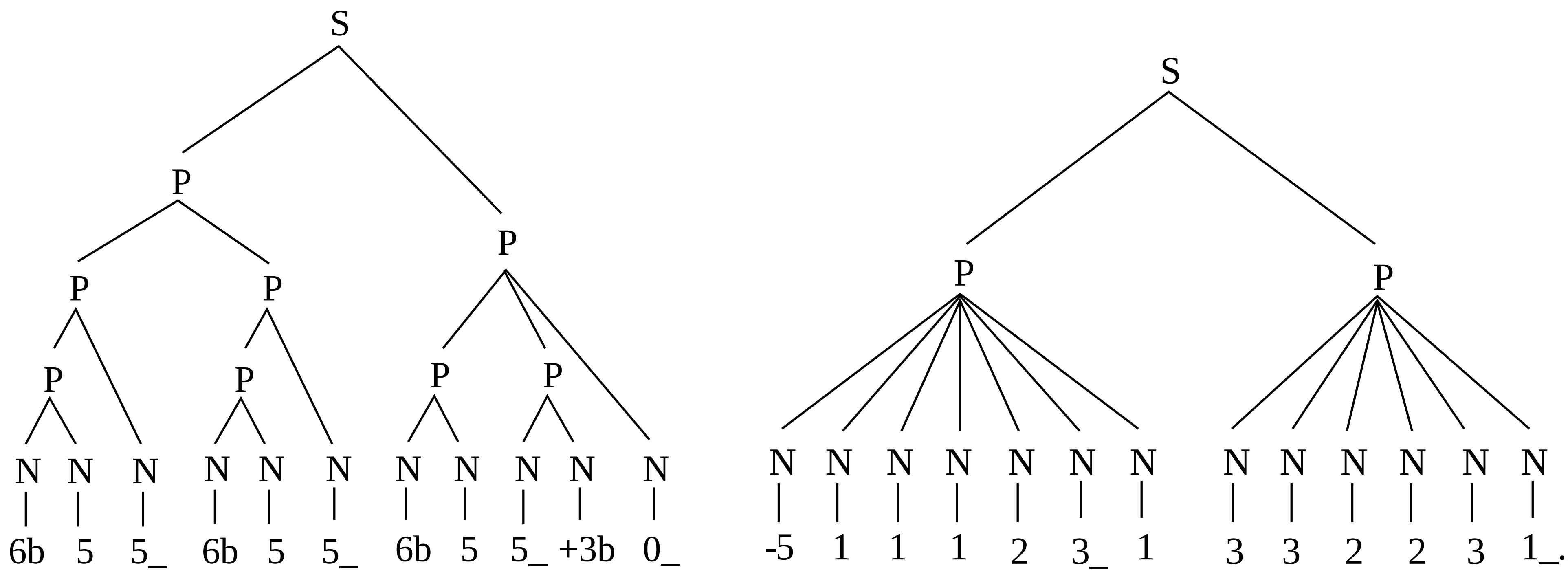
  "S" to each whole song

  "P" to each phrase

  "N" to each note

- In this way, we obtain conventional tree structures that can be used by DOP/STSG, or its isomorphic PCFG

Examples of some simple musical trees



Experimental Evaluation

Corpora are randomly divided into 10 *training/test set splits*

Experimental Evaluation

Corpora are randomly divided into 10 *training/test set splits*

Test 1: **Simplicity-Likelihood-DOP** (**SL-DOP**)

Selects simplest structure from among
n *likeliest structures*

Experimental Evaluation

Corpora are randomly divided into 10 *training/test set splits*

Test 1: **Simplicity-Likelihood-DOP** (**SL-DOP**)

Selects <u>simplest</u> structure from among
n *likeliest structures*

Test 2: **Likelihood-Simplicity-DOP (LS-DOP)**

Selects <u>likeliest</u> structure from among
n *simplest structures*

Scores of SL-DOP & LS-DOP

	SL-DOP (simplest among *n* likeliest)		LS-DOP (likeliest among *n* simplest)	
n	Language	Music	Language	Music
1	87.9%	86.0%	85.6%	84.3%
5	89.3%	86.8%	86.1%	85.5%
10	90.2%	87.2%	87.0%	85.7%
11	**90.2%**	**87.3%**	87.0%	85.7%
12	**90.2%**	**87.3%**	87.0%	85.7%
13	**90.2%**	**87.3%**	87.0%	85.7%
14	90.2%	87.2%	87.0%	85.7%
15	90.2%	87.2%	87.0%	85.7%
20	90.0%	86.9%	87.1%	85.7%
50	88.7%	85.6%	87.4%	86.0%
100	86.8%	84.3%	87.9%	86.0%
1,000	85.6%	84.3%	87.9%	86.0%

Scores of SL-DOP & LS-DOP

	SL-DOP (simplest among *n* likeliest)		LS-DOP (likeliest among *n* simplest)	
n	Language	Music	Language	Music
1	87.9%	86.0%	85.6%	84.3%
5	89.3%	86.8%	86.1%	85.5%
10	90.2%	87.2%	87.0%	85.7%
11	**90.2%**	**87.3%**	87.0%	85.7%
12	**90.2%**	**87.3%**	87.0%	85.7%
13	**90.2%**	**87.3%**	87.0%	85.7%
14	90.2%	87.2%	87.0%	85.7%
15	90.2%	87.2%	87.0%	85.7%
20	90.0%	86.9%	87.1%	85.7%
50	88.7%	85.6%	87.4%	86.0%
100	86.8%	84.3%	87.9%	86.0%
1,000	85.6%	84.3%	87.9%	86.0%

Same model obtains **maximal** scores for **both** language and music

Perceptual system strives for **simplest** analysis, but "searches" only among the **most likely** analyses (see Schaefer et al. 2004 for psychological experiments)

Comparison with other work

Language: DOP outperforms Collins, Charniak, Ratnaparkhi on WSJ

$\rightarrow$ non-head dependencies can only be covered by subtrees without (lexical) restrictions

Comparison with other work

Language: DOP outperforms Collins, Charniak, Ratnaparkhi on WSJ

$\rightarrow$ non-head dependencies can only be covered by subtrees without (lexical) restrictions

Music: DOP outperforms Temperley, Thom, Chang on EFC

$\rightarrow$ many phrases that include large intervals are not captured by harmonic, metrical or melodic "rules"

Comparison with other work

Language: DOP outperforms Collins, Charniak, Ratnaparkhi on WSJ

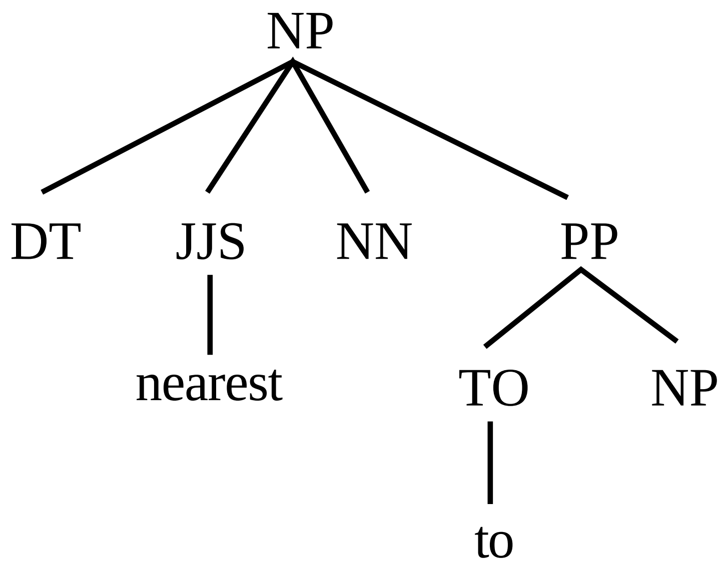
$\rightarrow$ non-head dependencies can only be covered by subtrees without (lexical) restrictions

Music: DOP outperforms Temperley, Thom, Chang on EFC

$\rightarrow$ many phrases that include large intervals are not captured by harmonic, metrical or melodic "rules"

By using largest possible subtrees (simplest analysis) which occur most frequently, DOP takes into account more dependencies

Example of non-headword dependency (ATIS corpus)



E.g.: *Show the nearest airport to Denver*

- non-head modifier *nearest* predicts the correct PP-attachment
- Example from WSJ: *BA carried <u>more</u> people <u>than</u> cargo in 1988*

Example of "jump phrase" (EFC)

Folksong K0690:

(3_2_11-5) (-5332211-5) (-512314_2) (...

- Gestalt principles predict "wrong" phrases on large intervals:

(3_2_11-5-5) (332211-5-5) (12314_2) (...

- Parallelism, meter & harmony reinforce same "wrong" predictions!

- Many phrases reflect idiom-dependent pitch contours which cannot be predicted by rules, but only by "patterns" (Cf. Huron 1996)

The importance of large subtrees

- Large subtrees may be statistically significant though they are linguistically and musically redundant

- Continuum between "regular phrases" (*rules*) and "idiomatic phrases" (*patterns*) both in language and music

- DOP can capture the full gradience between *rules* and *patterns*

How can we apply this to Visual Structures?

Structured visual databases are still too small (<300) to get statistically significant results

How can we apply this to Visual Structures?

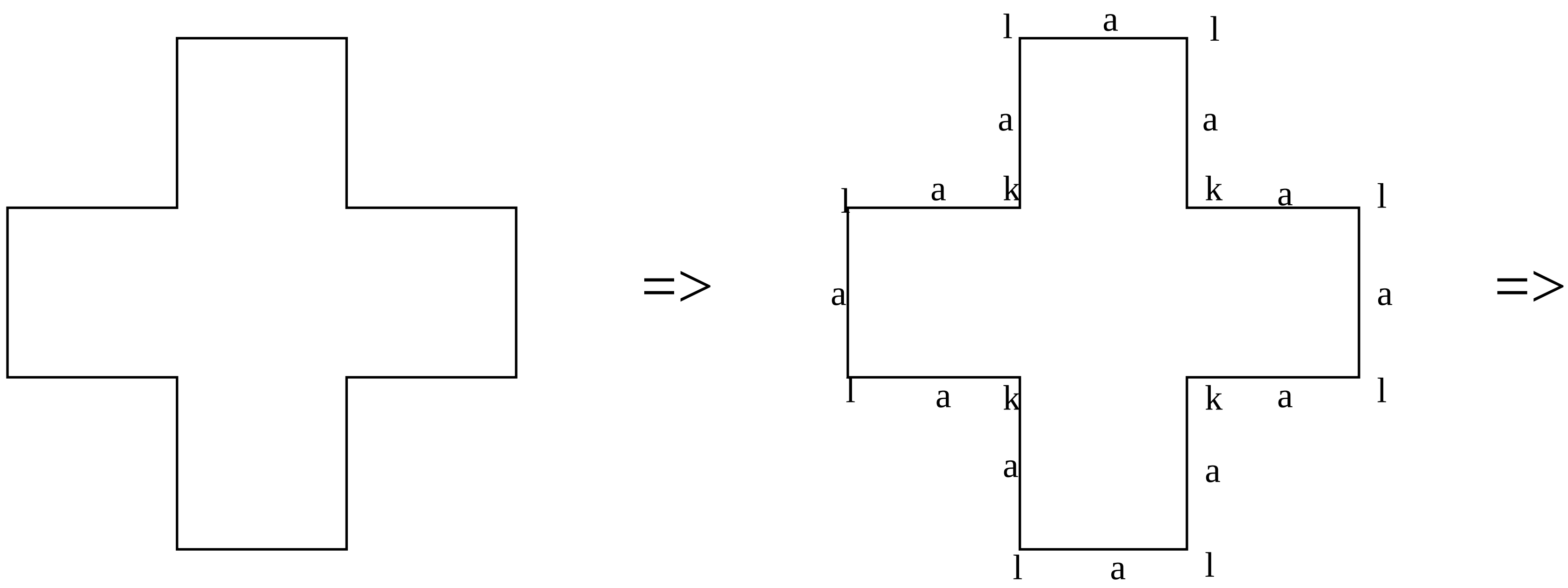
Structured visual databases are still too small (<300) to get statistically significant results

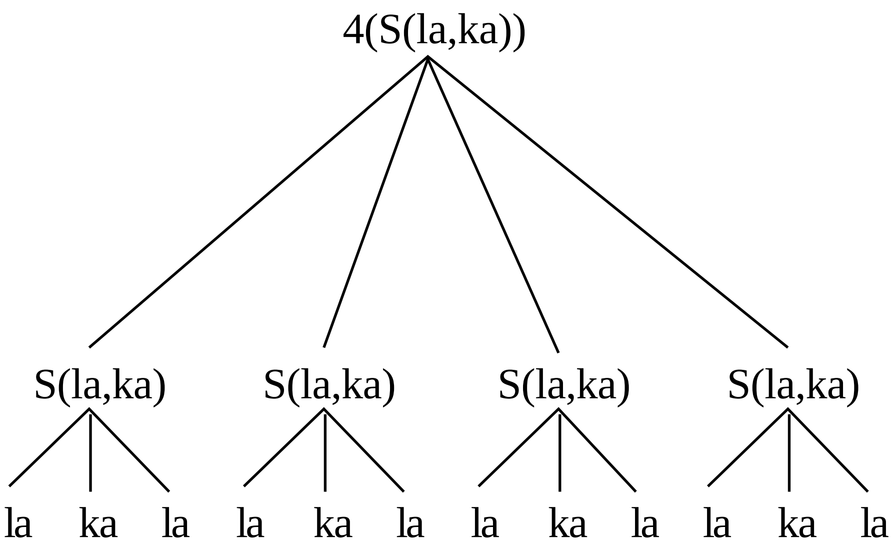
What are the primitive elements in visual perception?

In <u>Nijmegen Visual Database</u>: *line segments*, *angles*, *a.o.*

"Syntactic" categories: *symmetry* (*S*), *alternation* (*A*), *iteration* (*I*).

Of course, we only deal with medium-level computer vision in this way





Experiments support SL-DOP, but not statistically significant

DOP is used in various AI applications

- Structural language models for speech (Bod 1998, 2000; Chelba 1998)

  $\text{argmax}_W\ P(W \mid A) = \text{argmax}_W \sum_T P(W, T \mid A)$

- Statistical machine translation (Hearne & Way 2004; Poutsma & Bod 2003)

  argmax P(Translated sentence | Source sentence)

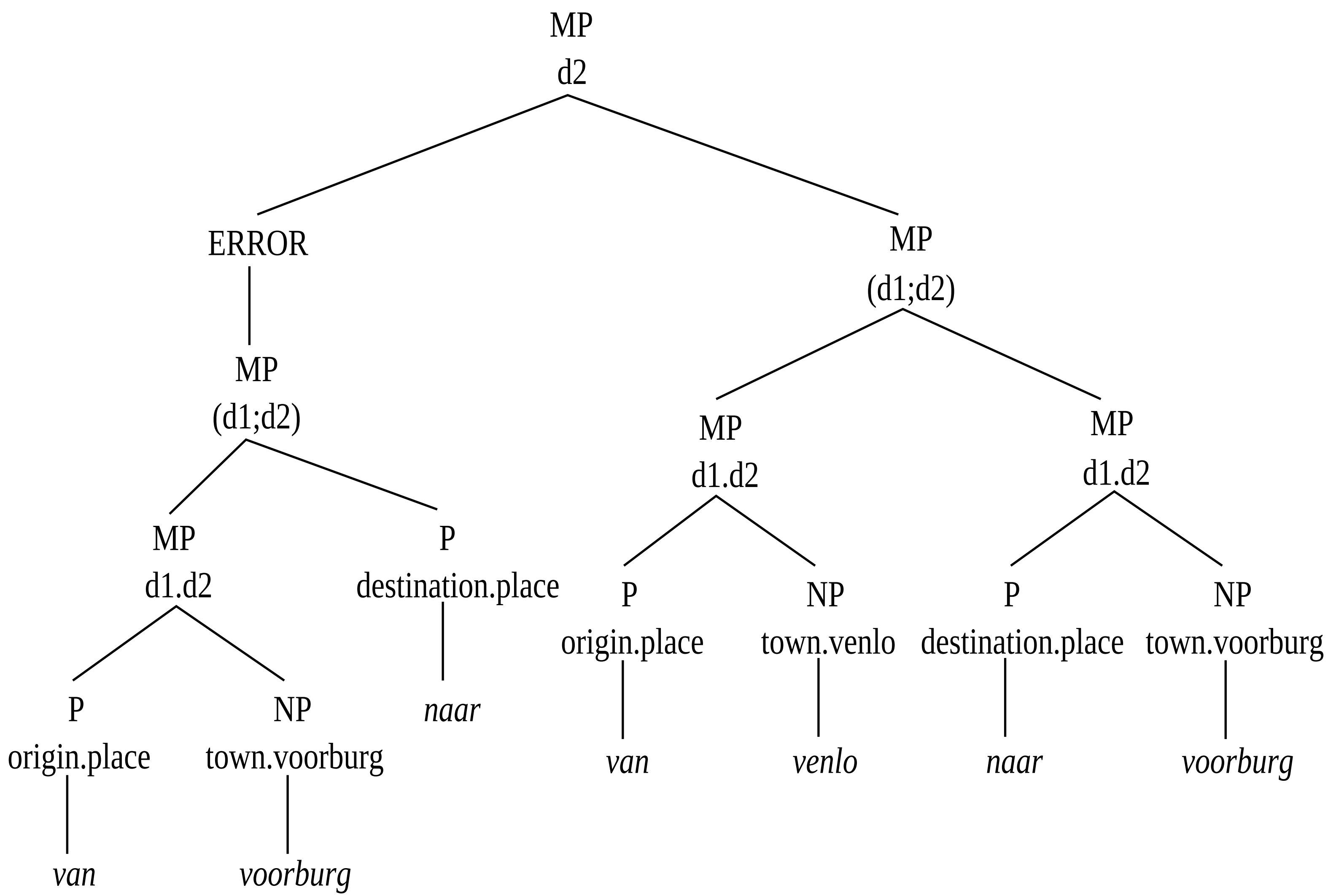
- Musical tempo tracking systems (Zaanen, Honing & Bod 2004)

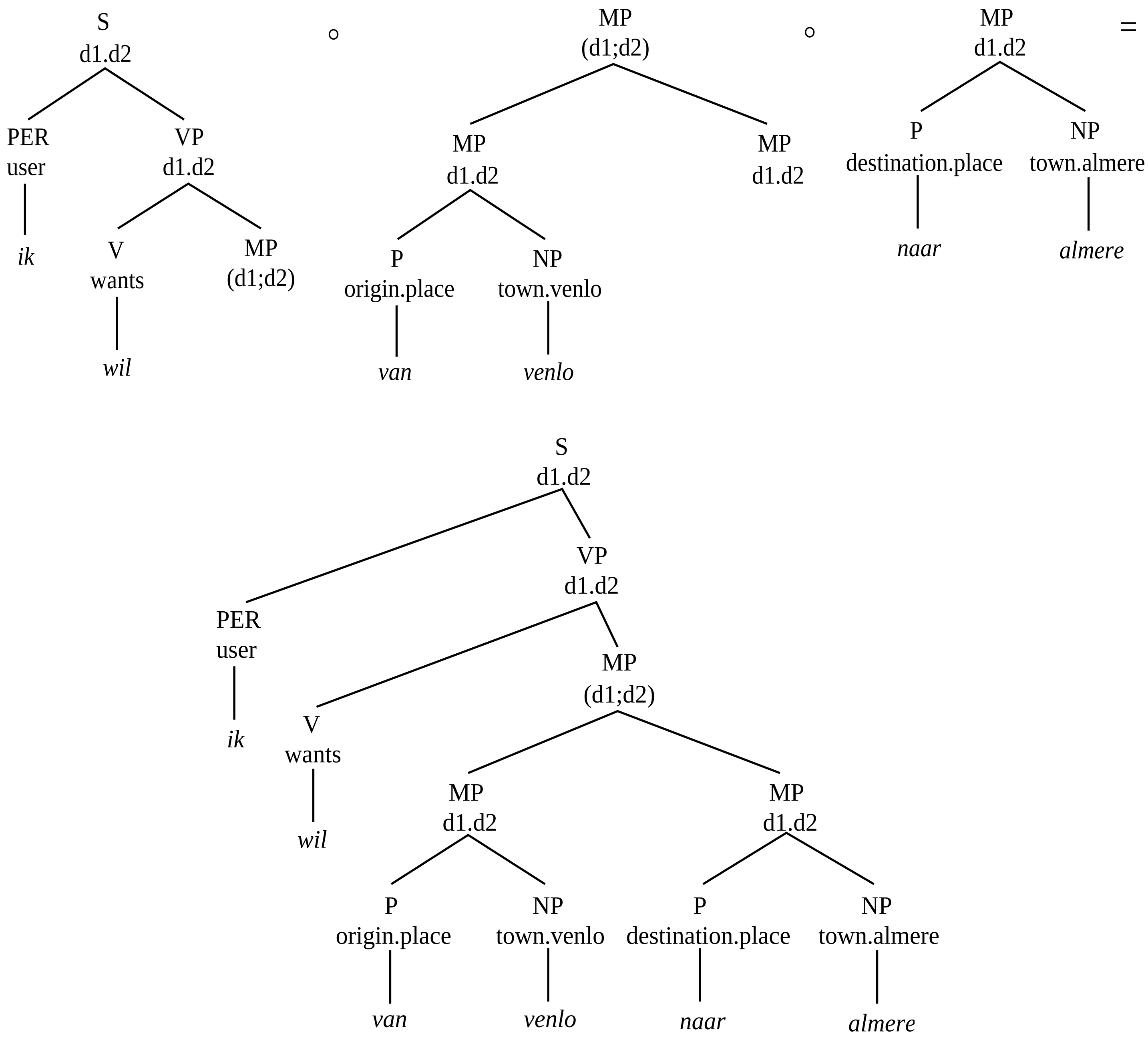
  argmax P(Temporal structure | Acoustic input)

- Interactive spoken dialog systems (Bod 1999; Scha et al. 1999), used by OVIS

  argmax P(Interpretation, Word string | Acoustic signal)

Example of OVIS annotation used in spoken dialog




S
d1.d2
PER
user
ik
VP
d1.d2
V
wants
wil
MP
(d1;d2)
∘
MP
(d1;d2)
MP
d1.d2
P
origin.place
van
NP
town.venlo
venlo
MP
d1.d2
∘
MP
d1.d2
P
destination.place
naar
NP
town.almere
almere
=
S
d1.d2
PER
user
ik
VP
d1.d2
V
wants
wil
MP
(d1;d2)
MP
d1.d2
P
origin.place
van
NP
town.venlo
venlo
MP
d1.d2
P
destination.place
naar
NP
town.almere
almere

How far do exemplar-based models stretch?

- Problem solving with exemplar-based model such as DOP/STSG?

How far do exemplar-based models stretch?

- Problem solving with exemplar-based model such as DOP/STSG?

- Exemplar-based reasoning has been proposed as early as Thomas Kuhn in his account on *normal science* (in his "*Structure of* ...")

  "Scientists solve problems by modeling them on previous problem-solutions" (Kuhn 1962)

How far do exemplar-based models stretch?

- Problem solving with exemplar-based model such as DOP/STSG?

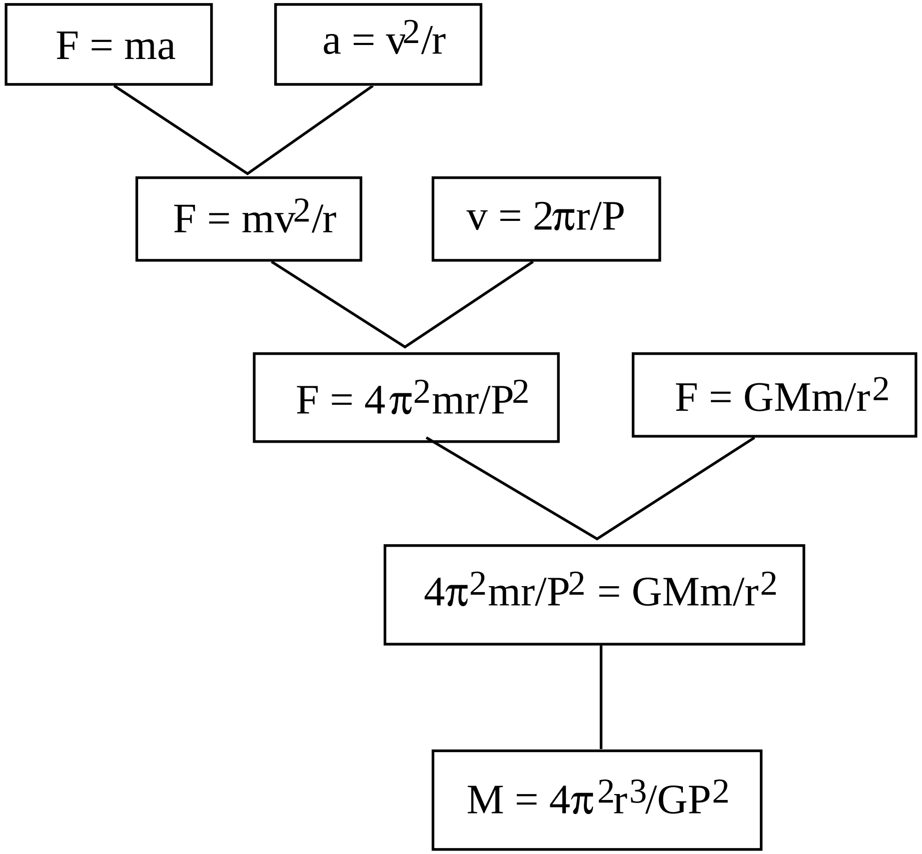
- Exemplar-based reasoning has been proposed as early as Thomas Kuhn in his account on *normal science* (in his "*Structure of* ...")

  "Scientists solve problems by modeling them on previous problem-solutions" (Kuhn 1962)

- Problem-solutions in physics can be represented by derivation trees
  -- though they do not represent grouping structure

Example of derivation tree in classical mechanics

Derivation of planet's mass from a satellite's orbit using Newton's laws



A tree describes the steps from higher-level laws to the solution (formula)

Subtrees can be reused to solve new problems

$F = ma$ $\qquad$ $a = v^2/r$

$F = mv^2/r$ $\qquad$ $v = 2\pi r/P$

$F = 4\pi^2 mr/P^2$ $\qquad$ $F = GMm/r^2$

$4\pi^2 mr/P^2 = GMm/r^2$

Deriving Kepler's third law by this subtree

$$F = ma \qquad a = v^2/r$$

$$F = mv^2/r \qquad v = 2\pi r/P$$

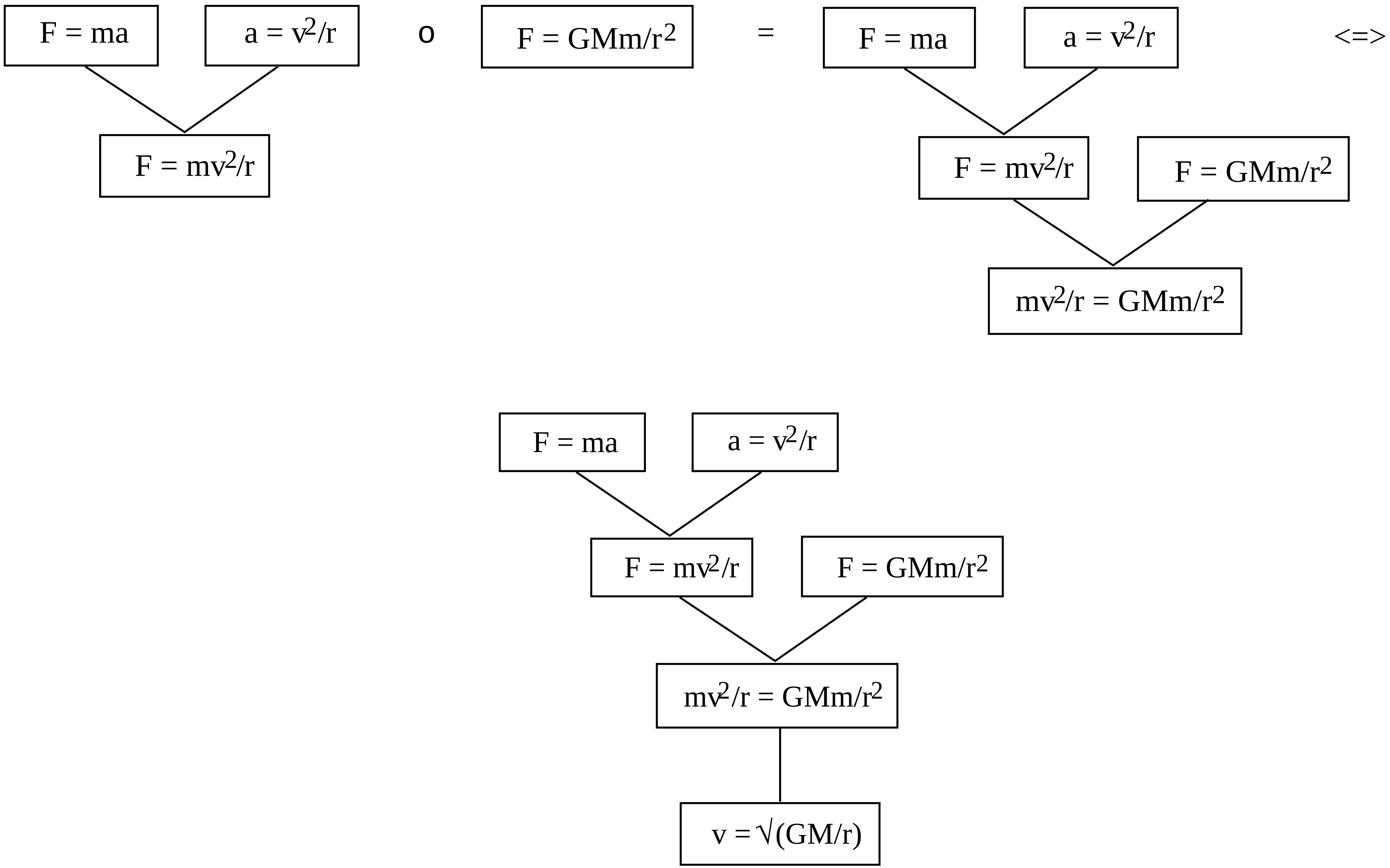
$$F = 4\pi^2 mr/P^2 \qquad F = GMm/r^2$$

$$4\pi^2 mr/P^2 = GMm/r^2$$

$$r^3/P^2 = GM/4\pi^2$$

We only need to solve the last equation of the previous subtree for r^3/P^2

Often we need to combine two or more subtrees (by term rewriting)



Derivation trees in fluid mechanics

E.g. derivation of *orifice system* from Bernoulli involves an ad hoc correction coefficient (C_d)

$$\Sigma E = \text{constant}$$

$$\rho g z_1 + \rho v_1^2/2 + p_1 = \rho g z_2 + \rho v_2^2/2 + p_2$$

$$p_1 = p_2$$
$$v_1 = 0$$
$$z_1 - z_2 = h$$

$$v = \sqrt{(2gh)}$$

$$Q(\text{theoretical}) = vA$$

$$Q(\text{theoretical}) = A\sqrt{(2gh)}$$

$$Q(\text{actual}) = C_d Q(\text{theoretical})$$

$$Q(\text{actual}) = C_d A\sqrt{(2gh)}$$

Derivation tree for a *weir* (*dam*) can still be derived by subtrees from orifice system that include the ad hoc correction

$$\Sigma E = \text{constant}$$

$$\rho g z_1 + \rho v_1^2/2 + p_1 = \rho g z_2 + \rho v_2^2/2 + p_2$$

$$p_1 = p_2$$
$$v_1 = 0$$
$$z_1 - z_2 = h$$

$$v = \sqrt{(2gh)}$$

$$Q(\text{theoretical}) = vA$$

$$Q(\text{theoretical}) = \int v\,dA$$

$$dA = B\,dh$$

$$Q(\text{theoretical}) = \int vB\,dh$$

$$Q(\text{theoretical}) = B\sqrt{(2g)} \int \sqrt{h}\,dh$$

$$Q(\text{theoretical}) = (2/3) B\sqrt{(2g)}\, h^{3/2}$$

$$Q(\text{actual}) = C_d Q(\text{theoretical})$$

$$Q(\text{actual}) = (2/3)\, C_d B\sqrt{(2g)}\, h^{3/2}$$

Simplicity and Likelihood also in problem solving

- Prefer largest possible derivational chunks, such that minimal recourse to additional derivational steps is needed

Simplicity and Likelihood also in problem solving

- Prefer largest possible derivational chunks, such that minimal recourse to additional derivational steps is needed

- Prefer more frequently occurring chunks: reflects usefulness

Simplicity and Likelihood also in problem solving

- Prefer largest possible derivational chunks, such that minimal recourse to additional derivational steps is needed

- Prefer more frequently occurring chunks: reflects usefulness

- *P*(*Derivation-tree* | *Phenomenon*) can be computed in a Bayesian way as in language and music, given a corpus of "exemplars"

We have just received an NWO grant for "Exemplar-Based Explanation" (one postdoc and one phd student)

Conclusions

- DOP provides a general framework for stochastic grammar models

Conclusions

- DOP provides a general framework for stochastic grammar models
- Same model achieves highest accuracy for both music and language on resp. EFC and WSJ

Conclusions

- DOP provides a general framework for stochastic grammar models
- Same model achieves highest accuracy for both music and language on resp. EFC and WSJ
- The model can also be used for vision, problem solving and reasoning -- as long as we can create a corpus of prior structures

Conclusions

- DOP provides a general framework for stochastic grammar models

- Same model achieves highest accuracy for both music and language on resp. EFC and WSJ

- The model can also be used for vision, problem solving and reasoning -- as long as we can create a corpus of prior structures

- **AI** should aim at developing general models for (each level of) cognition rather than particularist models for each cognitive task separately

  there are autonomous levels of explanation, but without striving for underlying models AI becomes a plethora of disparate algorithms