NPFL075 Practical Class 01

1) phrase-structure vs. dependency trees, non-projectivity
2) word- and morphological layer of PDT (homework)

Jiří Mírovský

Charles University

Institute of Formal and Applied Linguistics

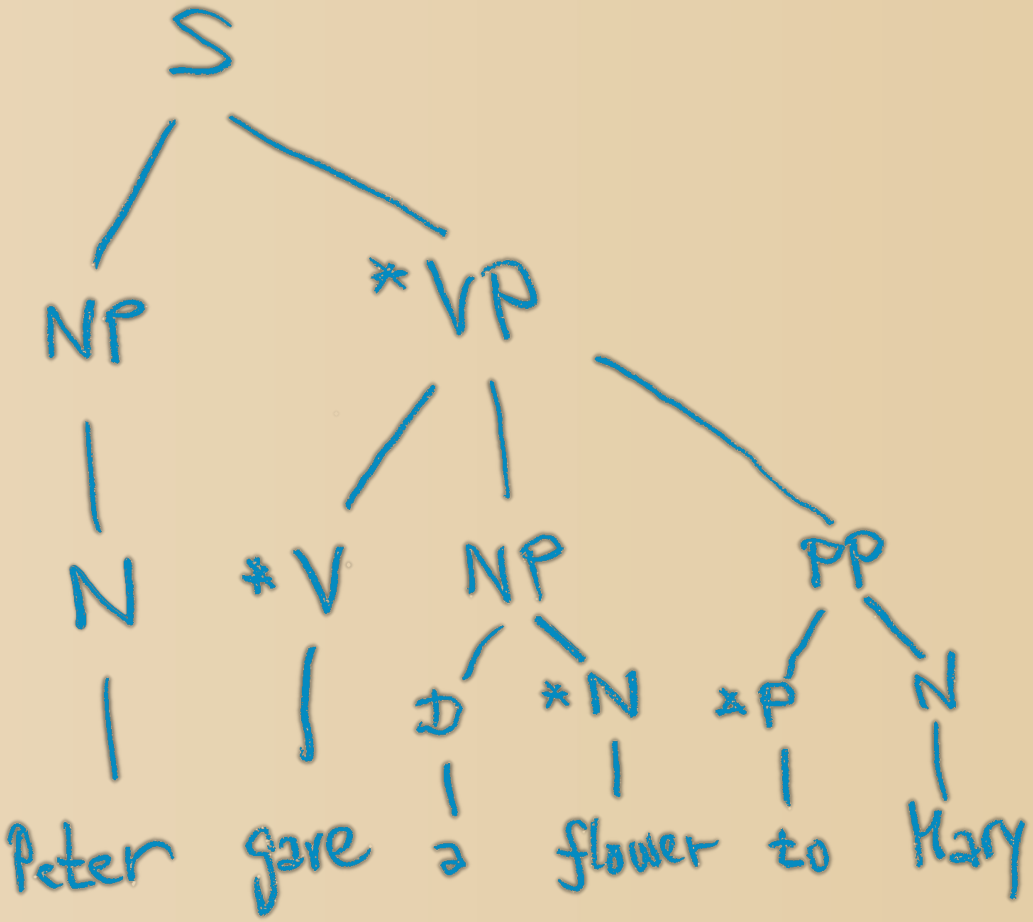
Phrase Structure Trees

```
S (
    NP ( N ( 'Peter' ) )
  * VP ( * V ( 'gave' )
            NP ( D ( 'a' )
                * N ( 'flower' ) )
            PP ( * P ( 'to' )
                    N   ( 'Mary' ) )
        )
   )
```

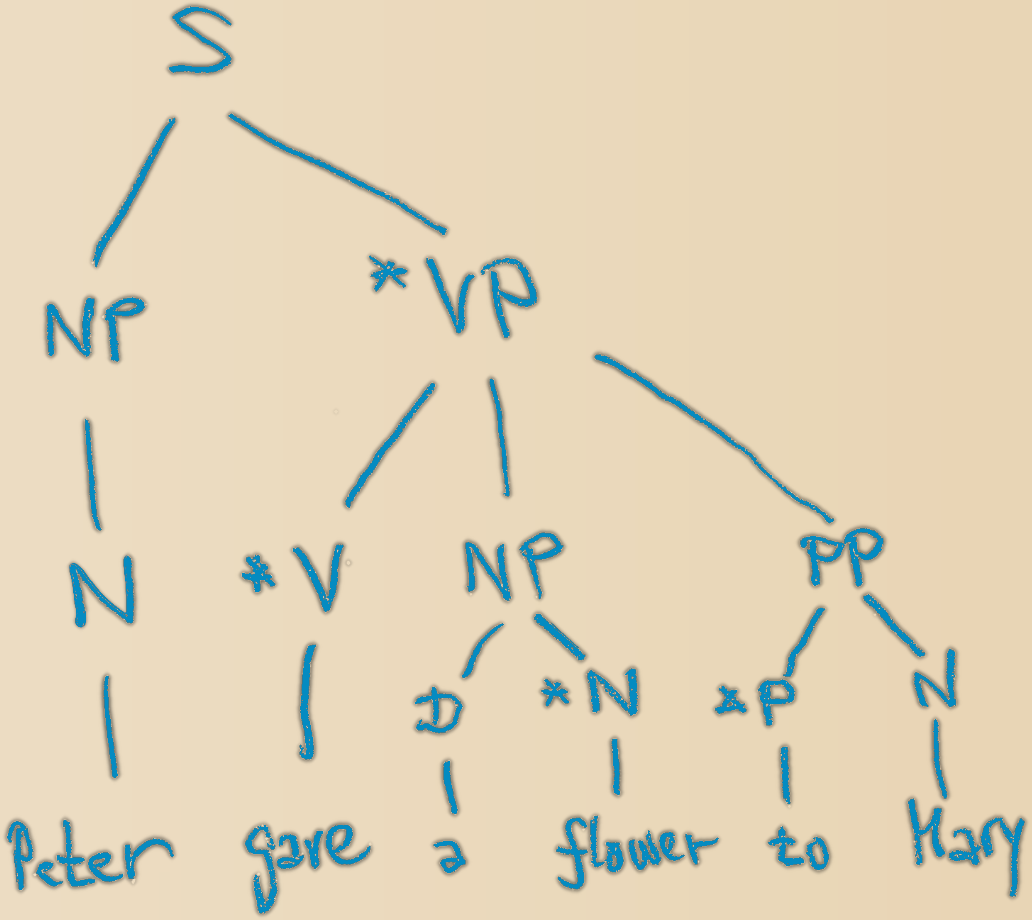
→ draw the tree

Phrase Structure Trees

```
S (
    NP ( N ( 'Peter' ) )
  * VP ( * V ( 'gave' )
            NP ( D ( 'a' )
                * N ( 'flower' ) )
            PP ( * P ( 'to' )
                   N   ( 'Mary' ) )
         )
   )
```

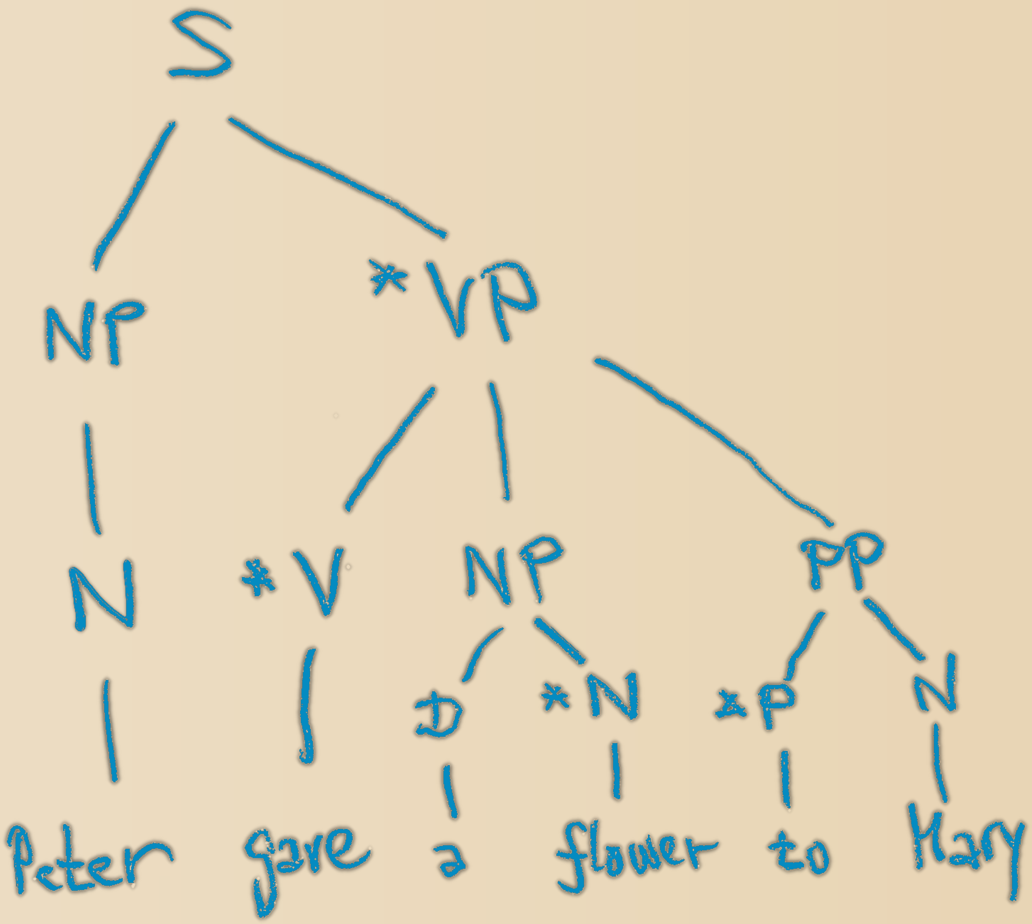


Phrase Structure and Dependency Trees



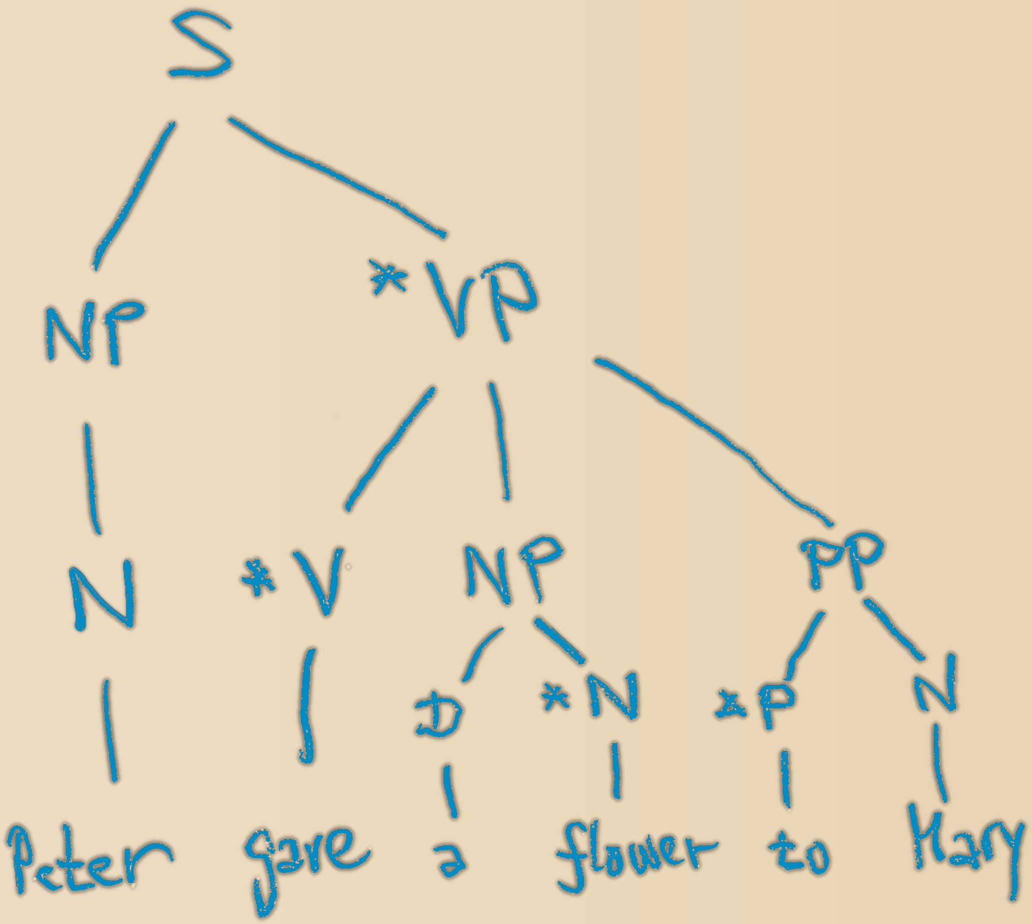
- properties?

Phrase Structure and Dependency Trees



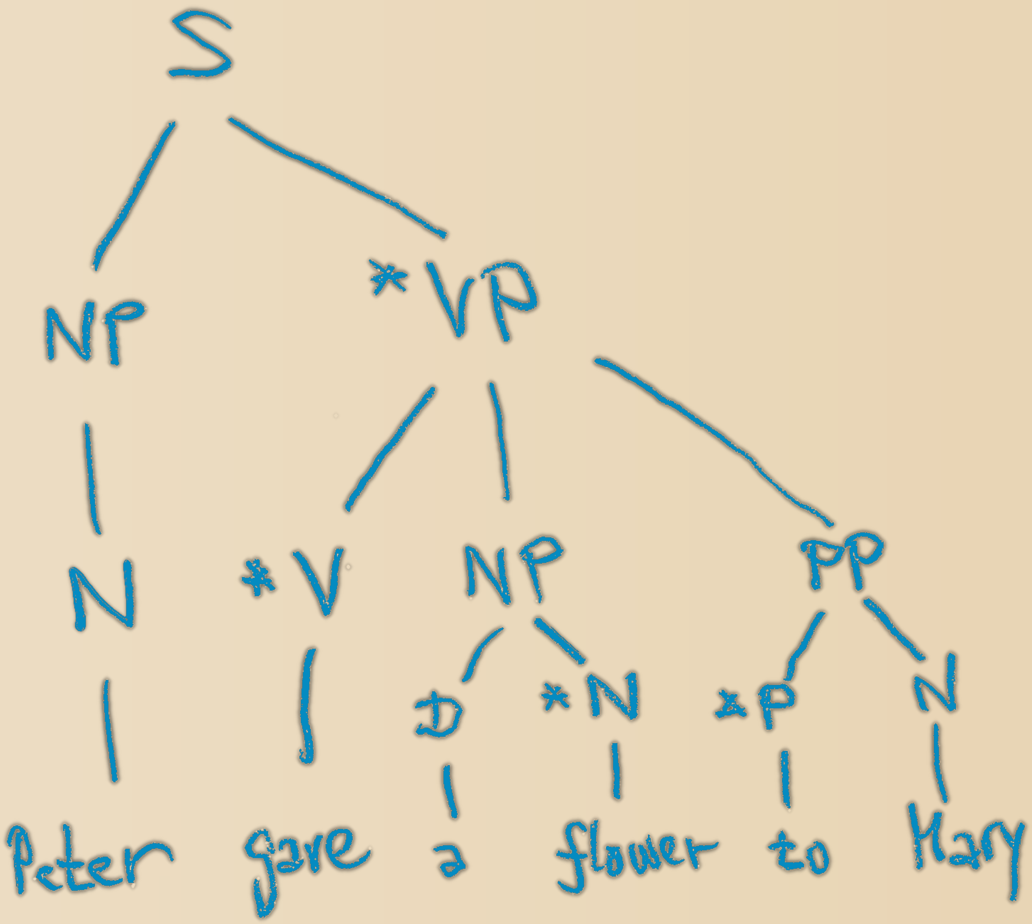
- oriented edges
- a single root
- each node has a single parent (except for the root)
- linear order

Phrase Structure Trees



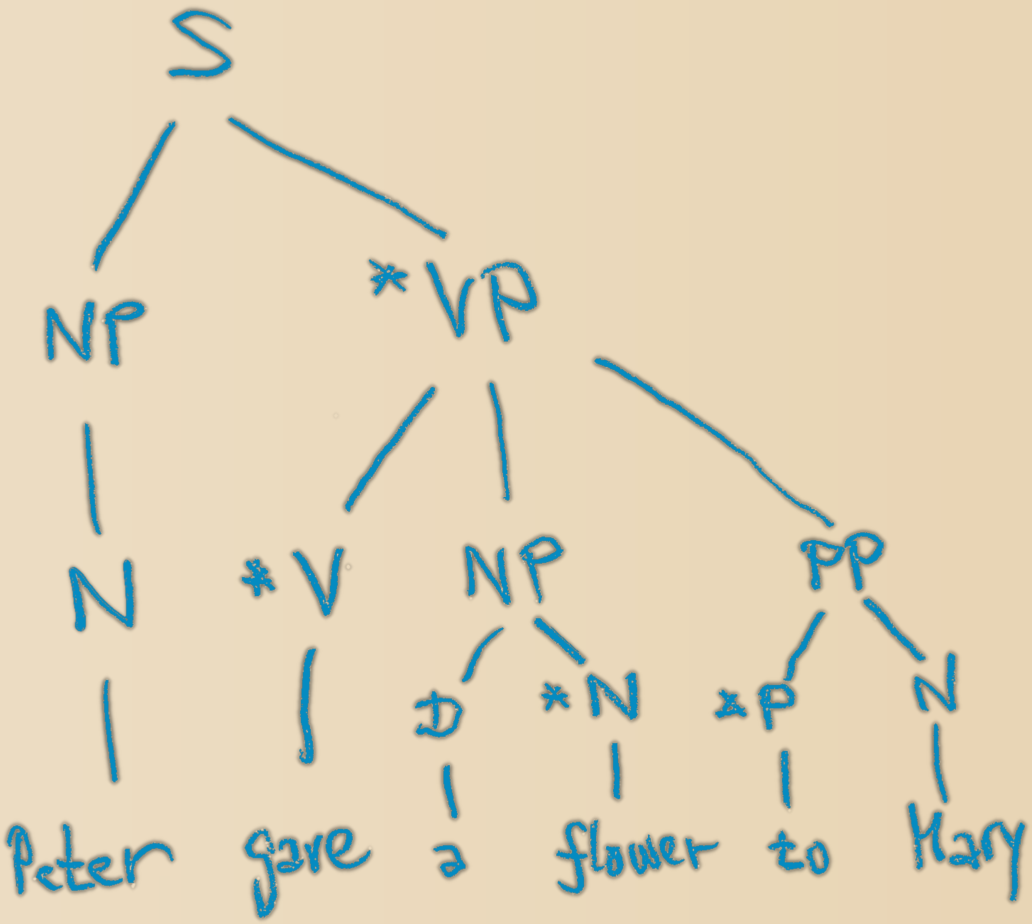
- additional properties?

Phrase Structure Trees



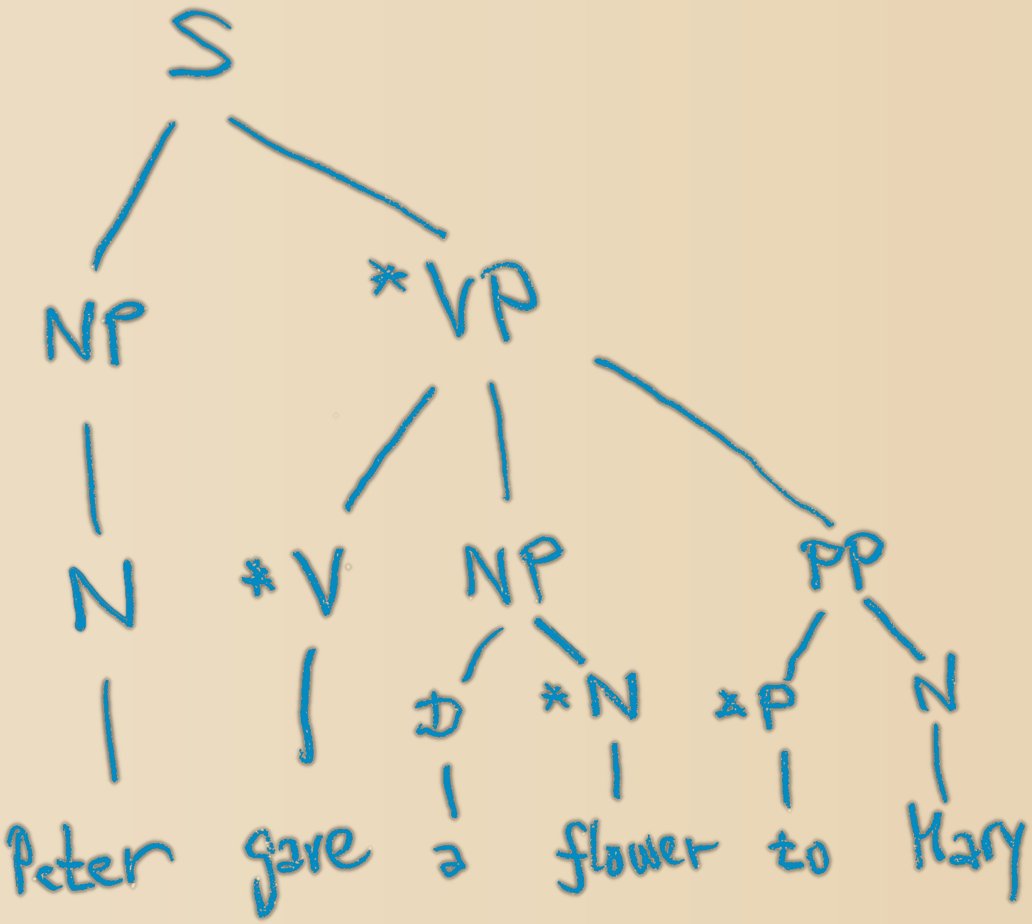
- projectivity (no crossing edges)
- context-free grammar

Phrase Structure Trees



write a context-free grammar from the tree

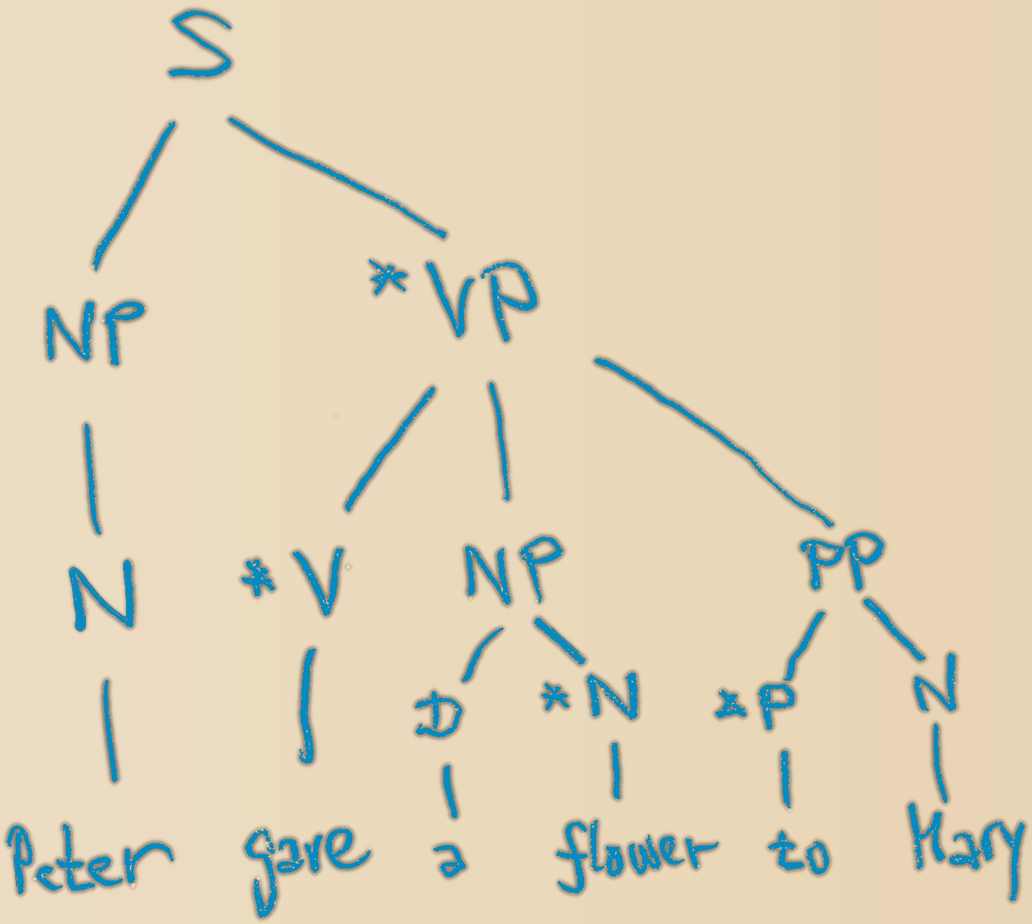
Phrase Structure Trees



S → NP *VP
NP → N
VP → *V NP PP
NP → D *N
PP → *P N

N → ‘Peter‘
…

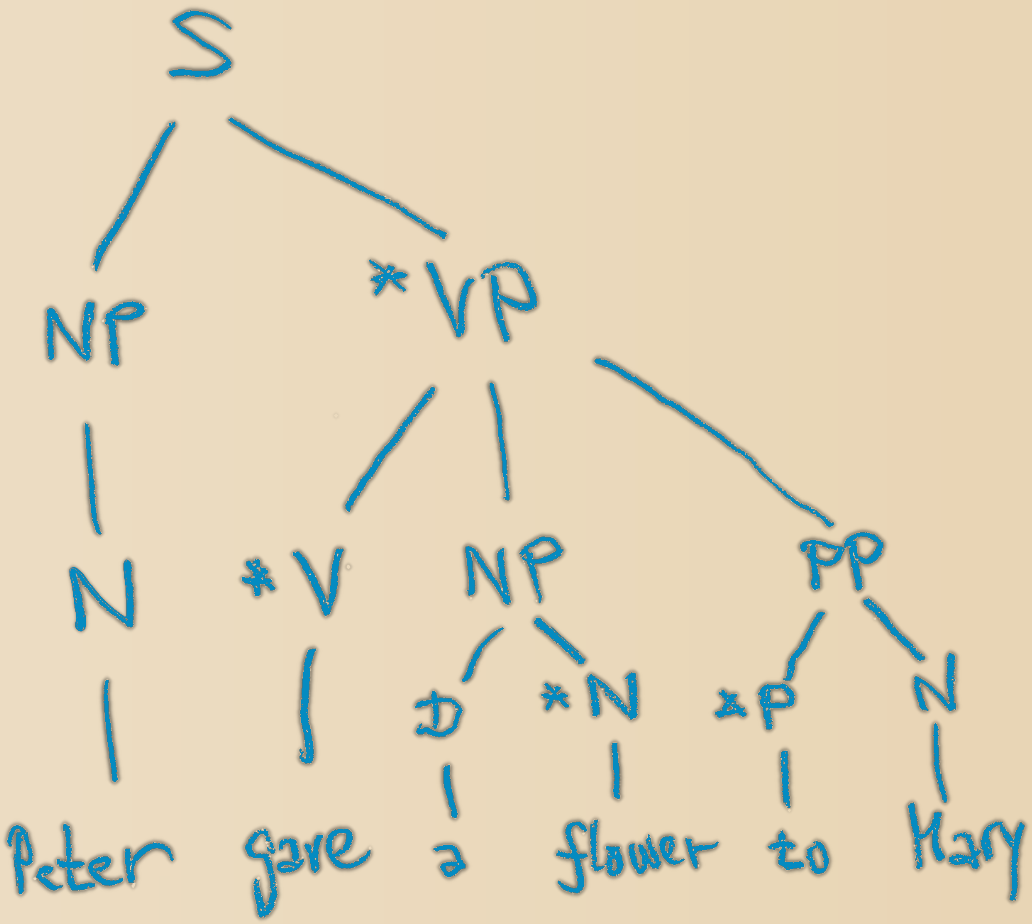
Phrase Structure Trees → Dependency Trees



bottom-up

- for each terminal node find its governor (head)
  - Peter → gave
  - gave = root
  - a → flower
  - …

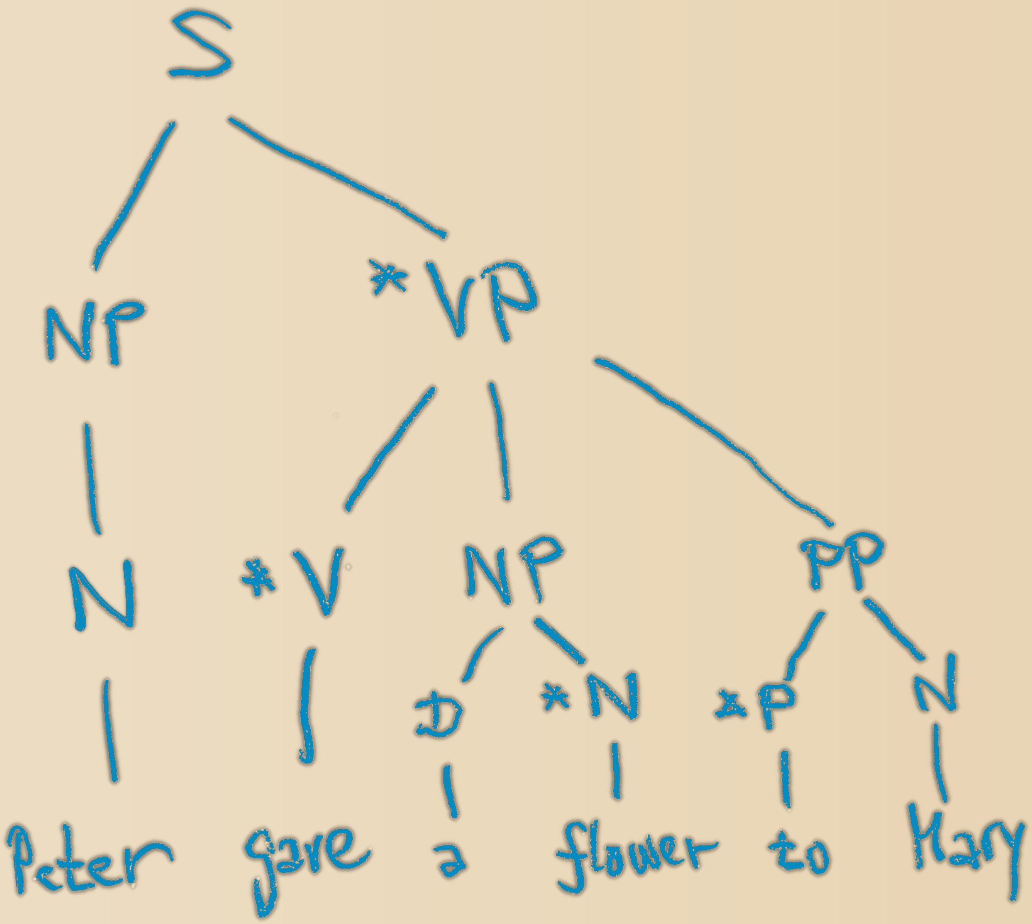
Phrase Structure Trees → Dependency Trees



from top to down

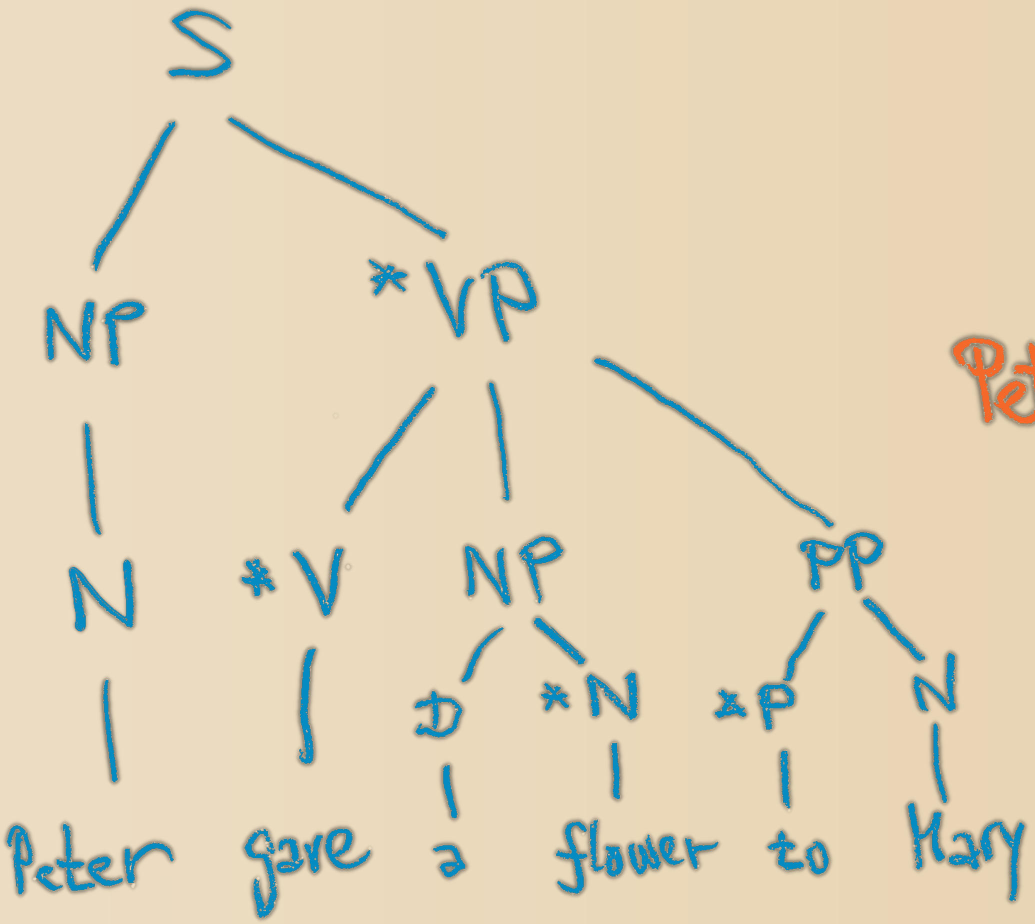
- start with S and go recursively:
  - process the head branch → the root of the subtree
  - process other branches and add them as sons of the subtree root

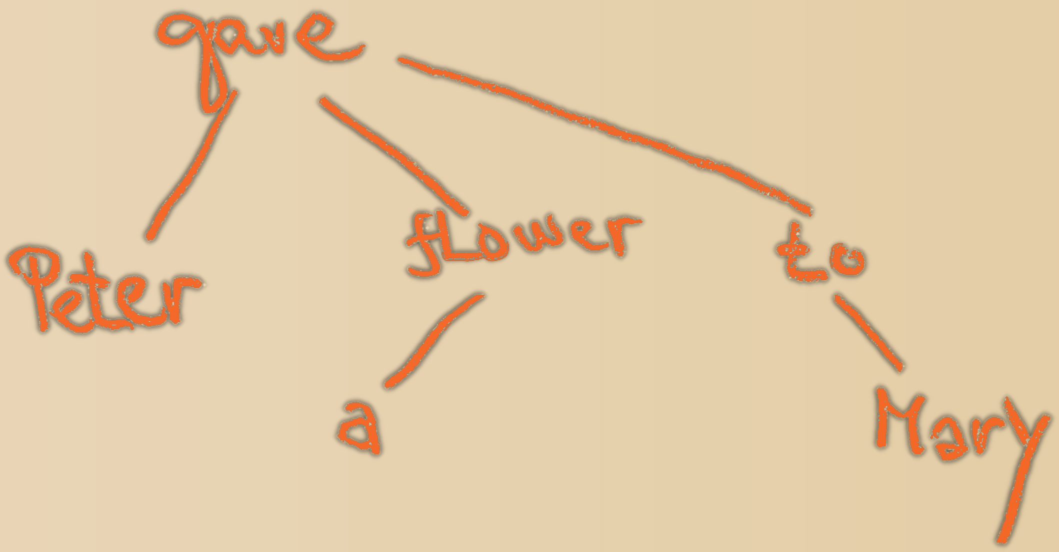
Phrase Structure Trees → Dependency Trees



Draw the dependency tree from the phrase-structure tree.

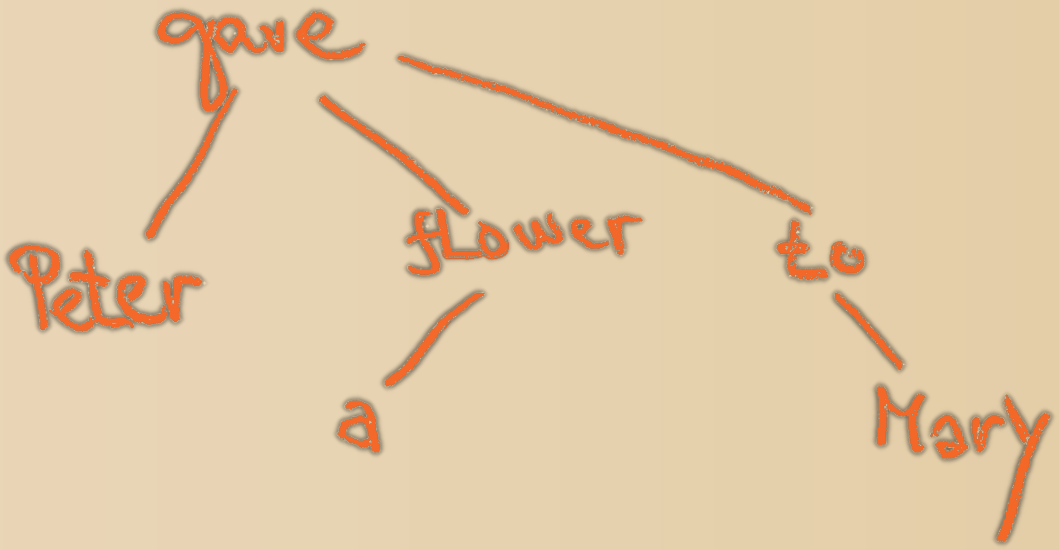
Phrase Structure Trees
→ Dependency Trees





Data Representation of Trees
1) a parent for each node

(**note:** for now, we do not care about linear order)

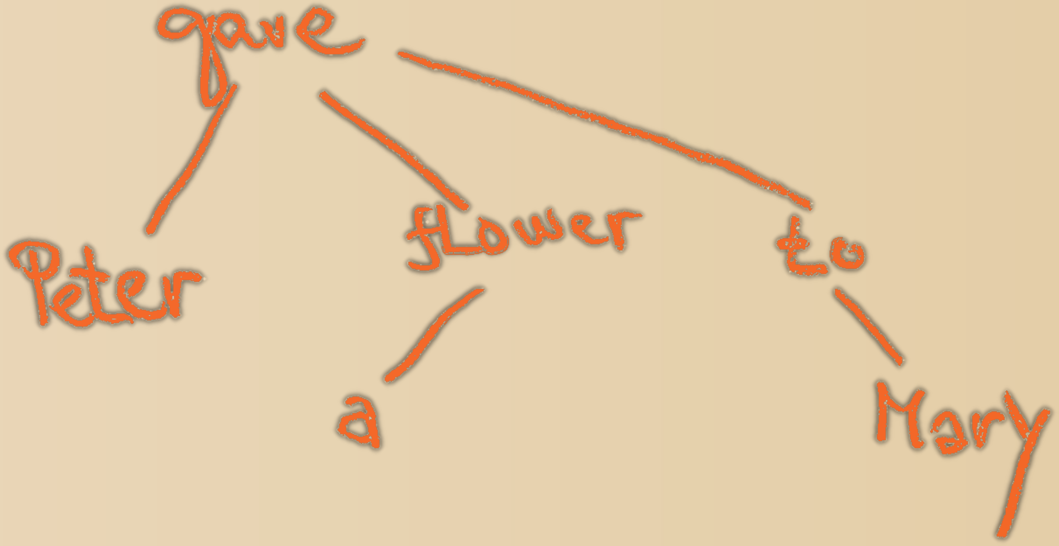


properties: simple, constant size for each node

Data Representation of Trees
1) a parent for each node

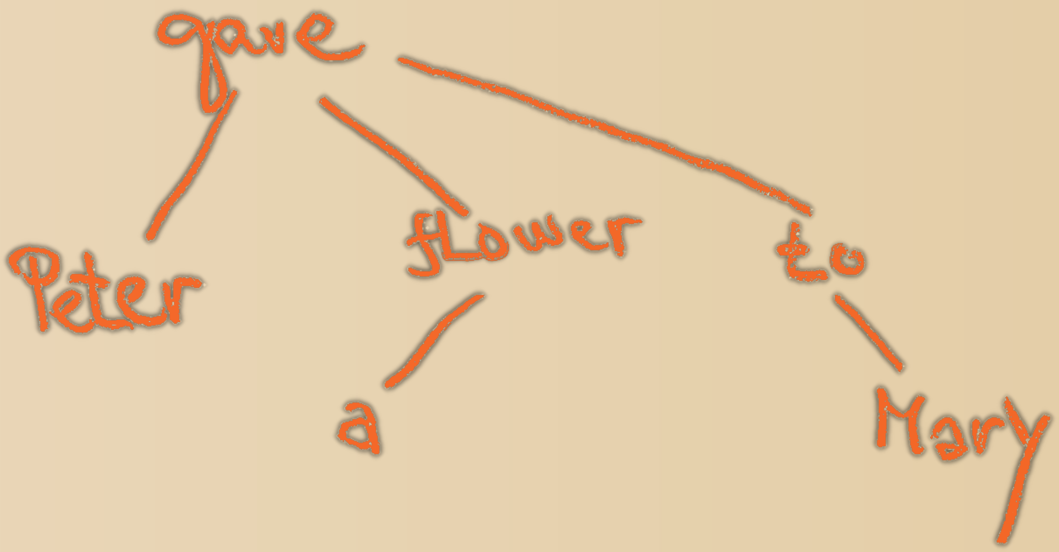
(**note:** for now, we do not care about linear order)

gave.
Peter.gave
flower.gave
a.flower
to.gave
Mary.to



properties: simple, constant size for each node

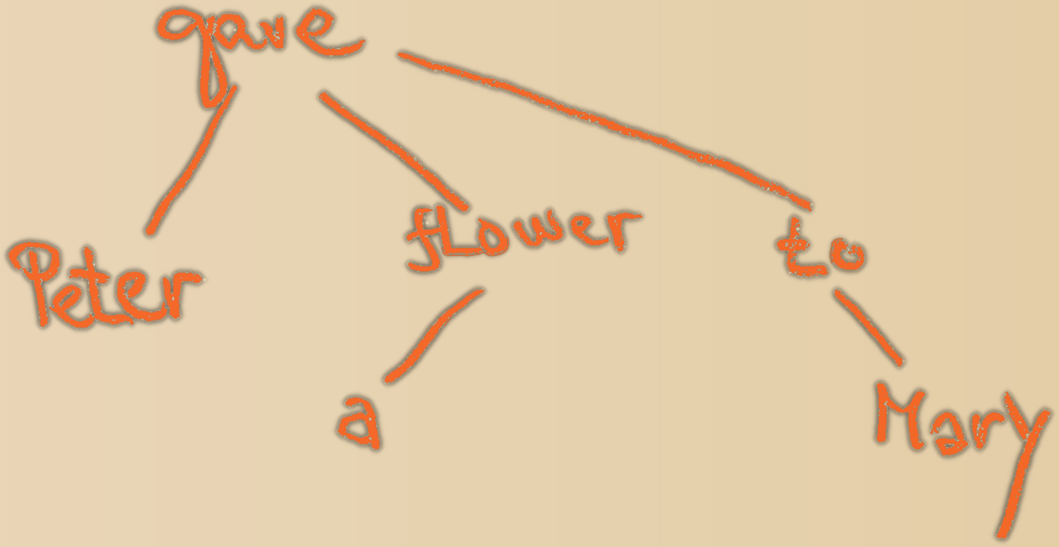
Data Representation of Trees
2) list of sons for each node



properties: simple, variable size for each node, good for top-down approach

Data Representation of Trees
2) list of sons for each node

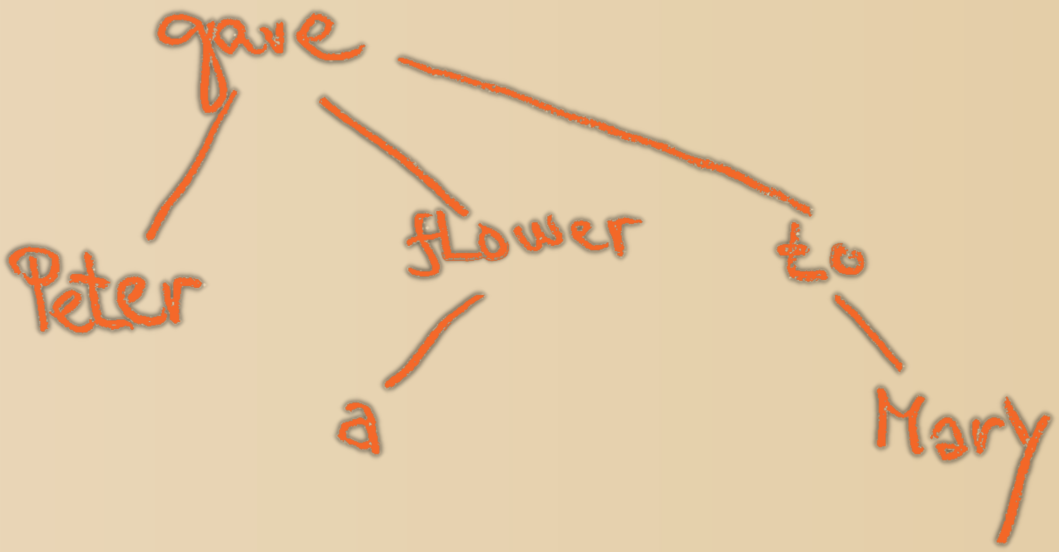
gave: Peter, flower, to
flower: a
to: Mary



properties: simple, variable size for each node, good for top-down approach

Data Representation of Trees

3) first son and first brother for each node

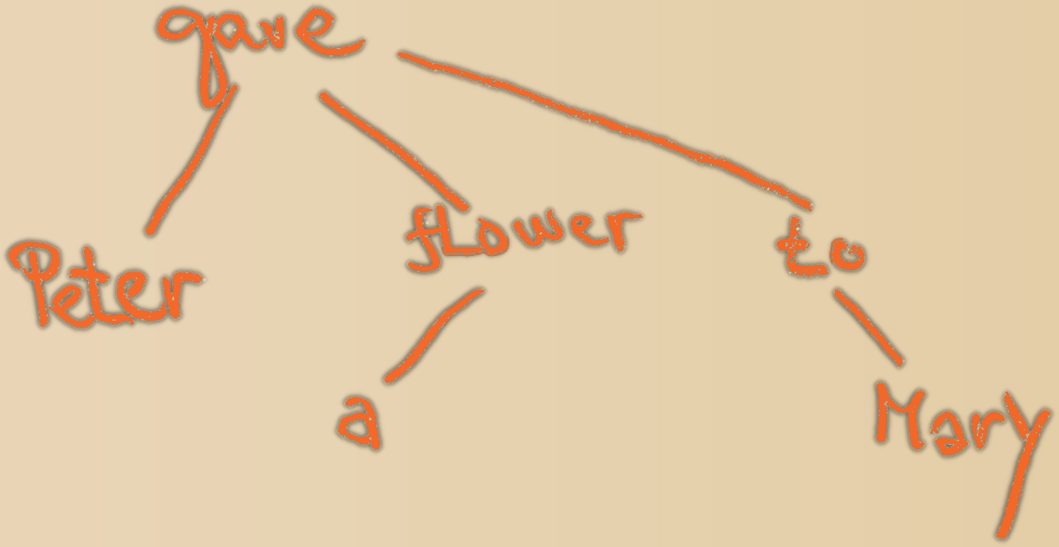


properties: unintuitive, fixed size for each node, good for top-down approach

Data Representation of Trees

3) first son and first brother for each node

gave: Peter/-
Peter: -/flower
flower: a/to
a: -/-
to: Mary/-
Mary: -/-



properties: unintuitive, fixed size for each node, good for top-down approach

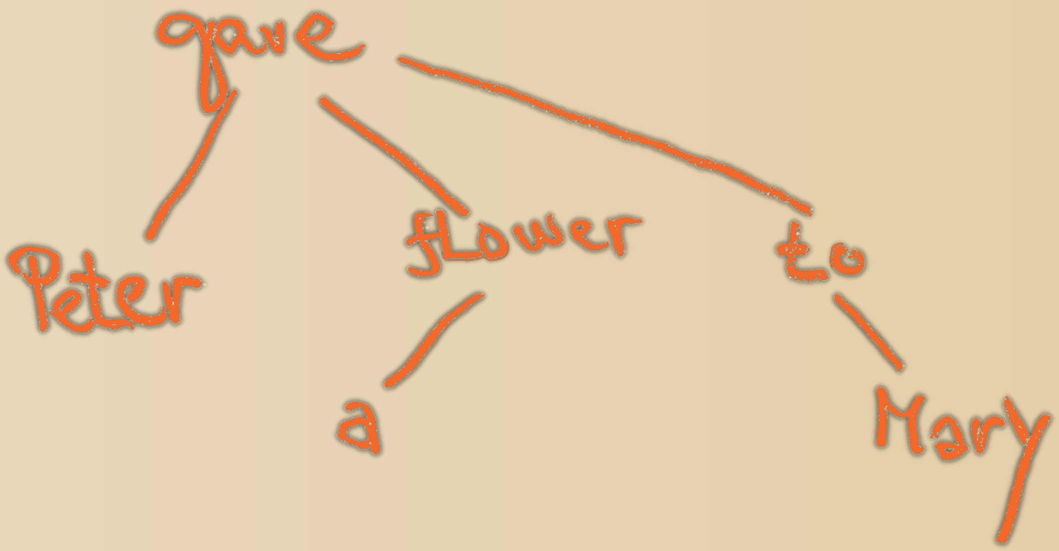
Two File Formats for Trees

1) reference to parent

gave.
Peter.gave
flower.gave
a.flower
to.gave
Mary.to

2) recursive list of sons

gave(Peter,flower(a),to(Mary))



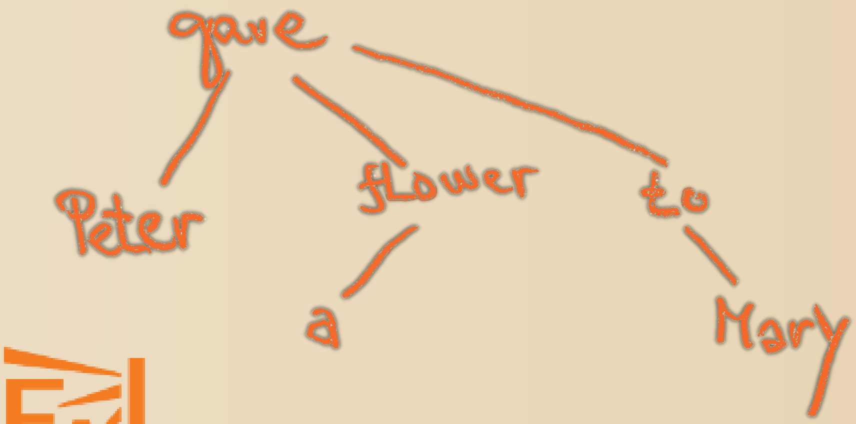
(**note:** we still do not care about linear order)

Two File Formats for Trees

1) reference to parent

gave.
Peter.gave
flower.gave
a.flower
to.gave
Mary.to

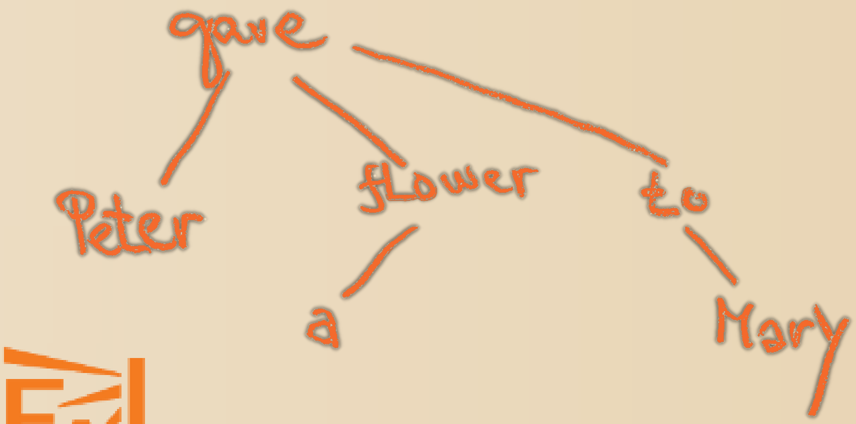
Possible errors:



Two File Formats for Trees

1) reference to parent

gave.
Peter.gave
flower.gave
a.flower
to.gave
Mary.to



Possible errors:

errors in syntax:

- no dot or more dots at a line
- forbidden characters

errors in semantics:

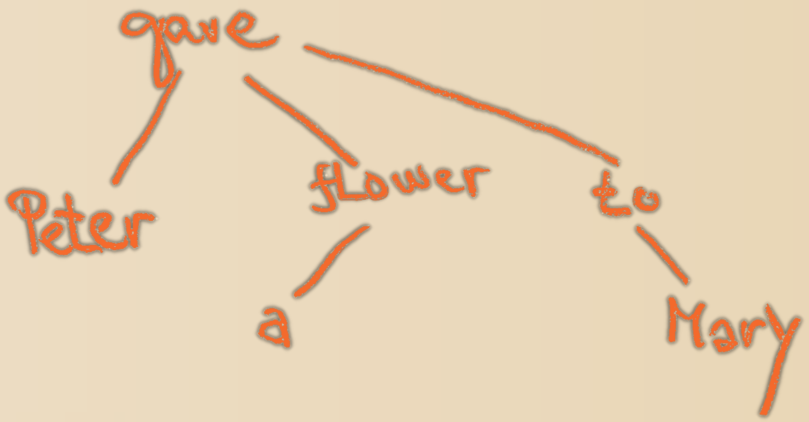
- missing root or more than one root
- several parents for a single node
- self-reference
- cycle in the references
- discontinuous tree
- empty file (maybe not an error)

Two File Formats for Trees

2) recursive list of sons

gave (Peter,flower (a),to(Mary))

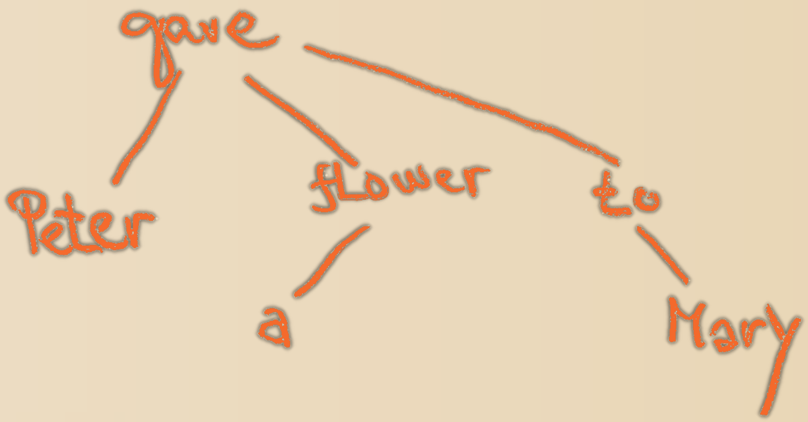
Possible errors:



Two File Formats for Trees

2) recursive list of sons

gave (Peter,flower (a),to(Mary))



Possible errors:

errors in syntax:

- many possible errors, e.g. mismatching parentheses

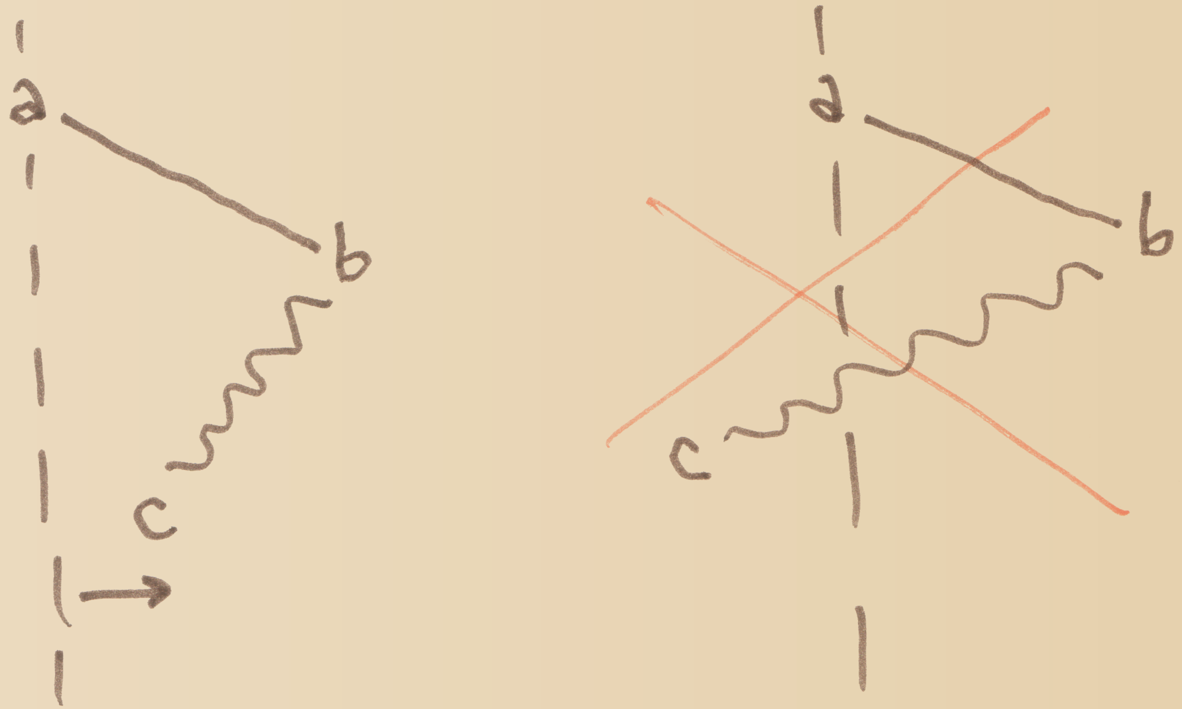
errors in sytax/semantics:

- repeated node
- multiple roots: a (b c) d (e f)
- no root: (b c)

Non-Projective Trees

Definition 1:

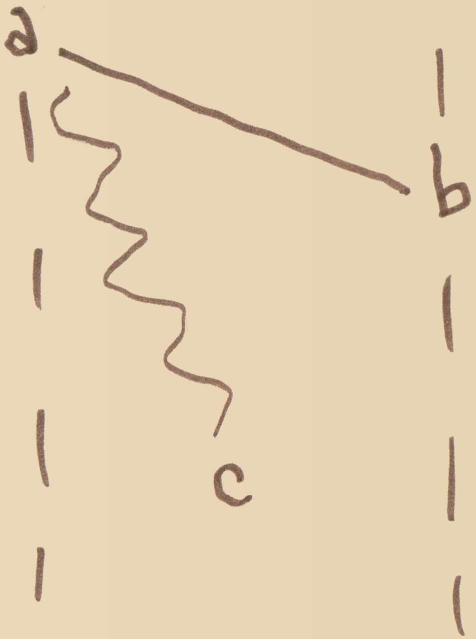
a is father of **b**, **b** is on the right from **a**, **c** is a transitive child of **b** => **c** is on the right from **a** (and vice versa)



Non-Projective Trees

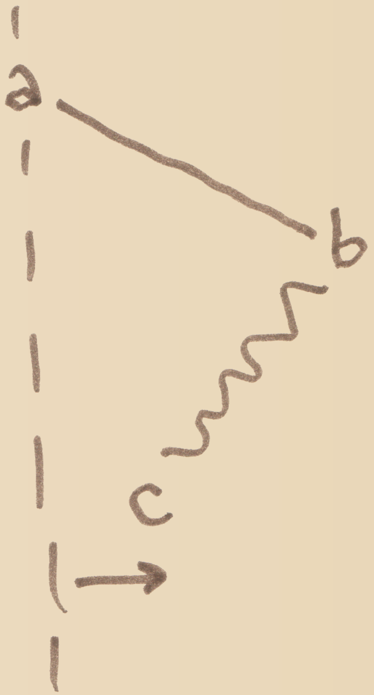
Definition 2:

a is father of **b** => any node inbetween them has to be a transitive child of **a**

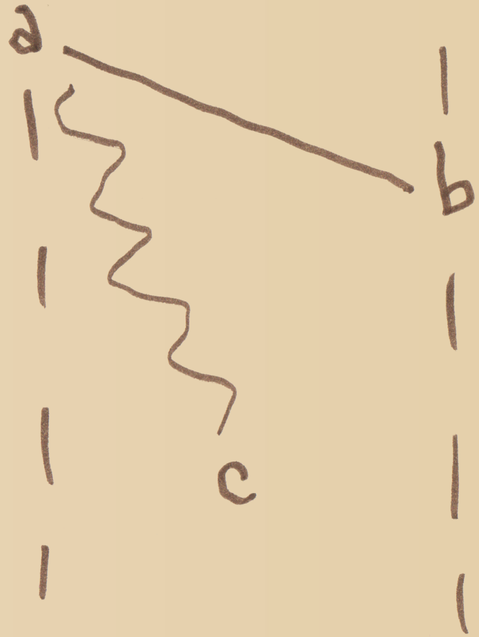


Non-Projective Trees

Are the definitions equivalent?

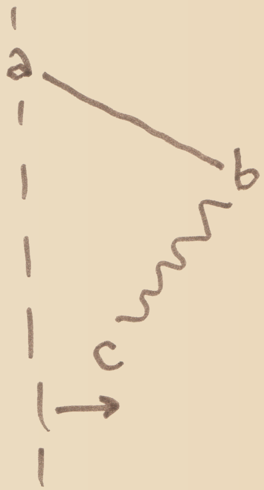


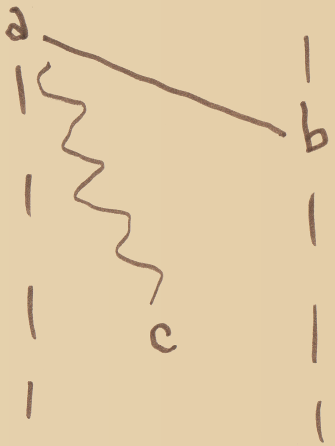
vs.

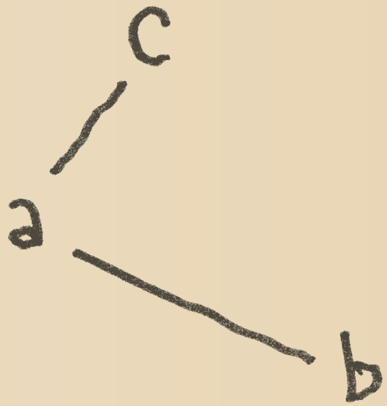


Non-Projective Trees

They are not equivalent:

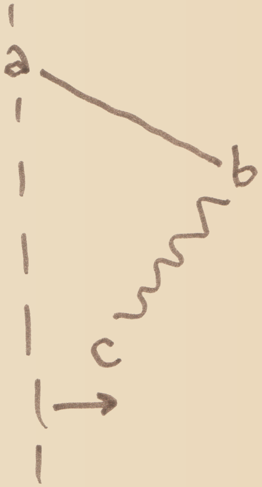


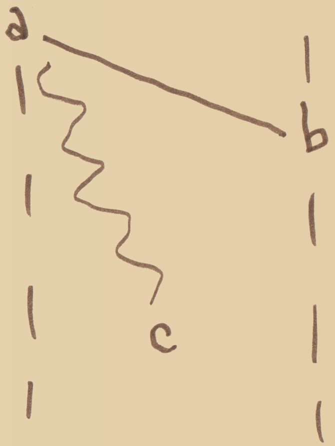


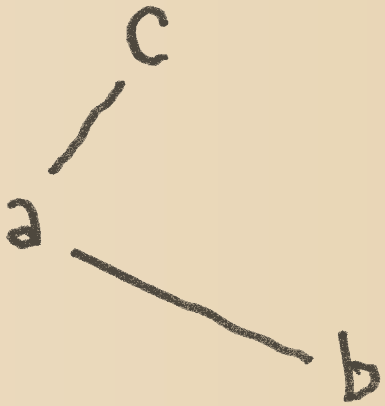


Non-Projective Trees

They are not equivalent:



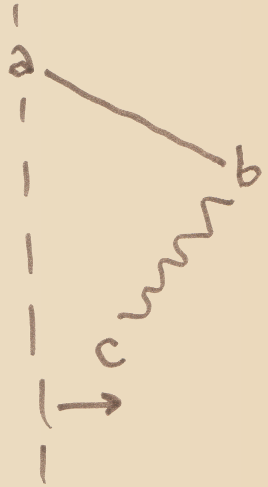


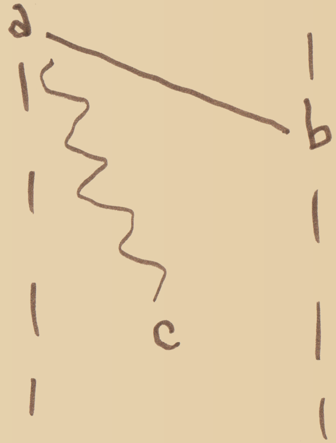


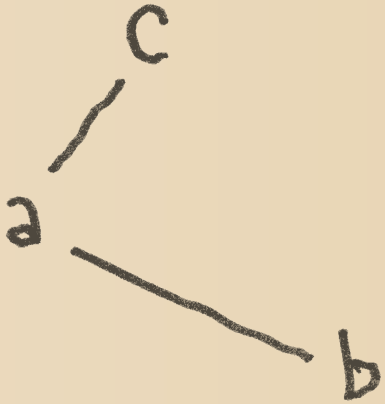
both definitions

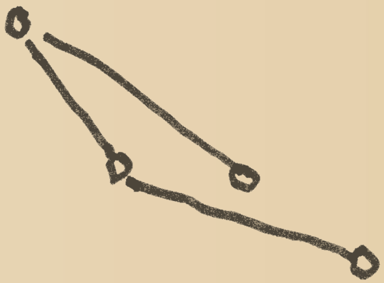
Non-Projective Trees

They are not equivalent:





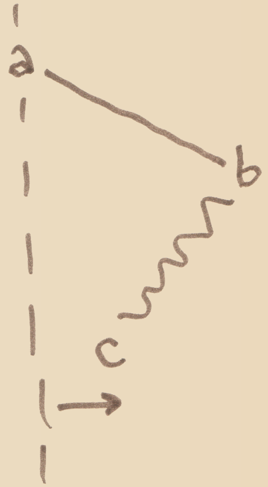


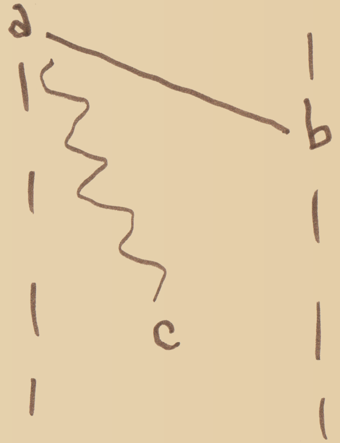


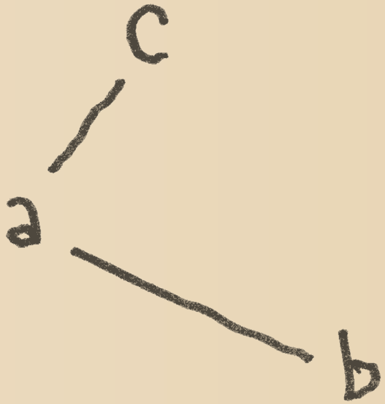
both definitions

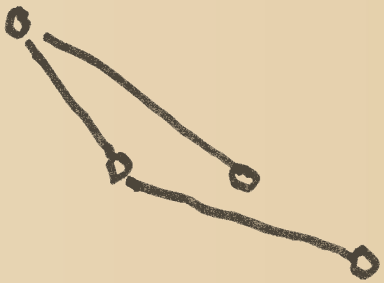
Non-Projective Trees

They are not equivalent:







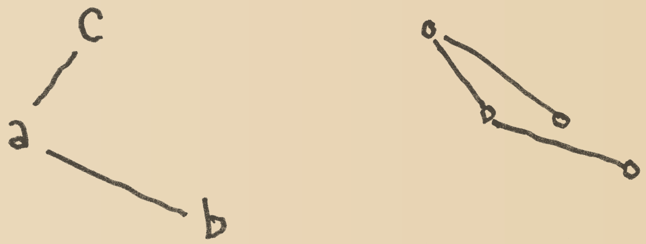


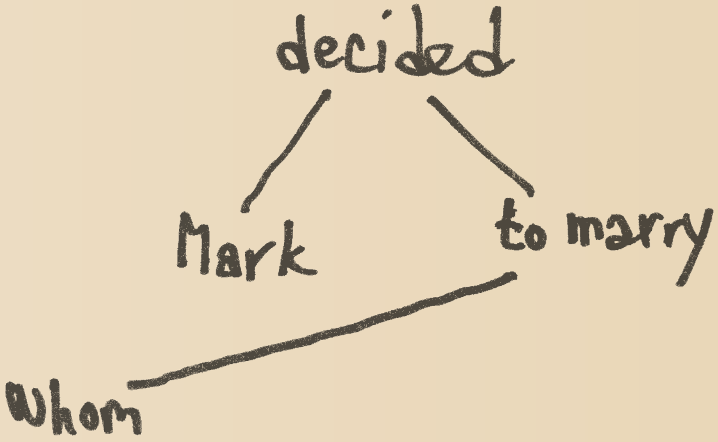
both definitions

only def. 2

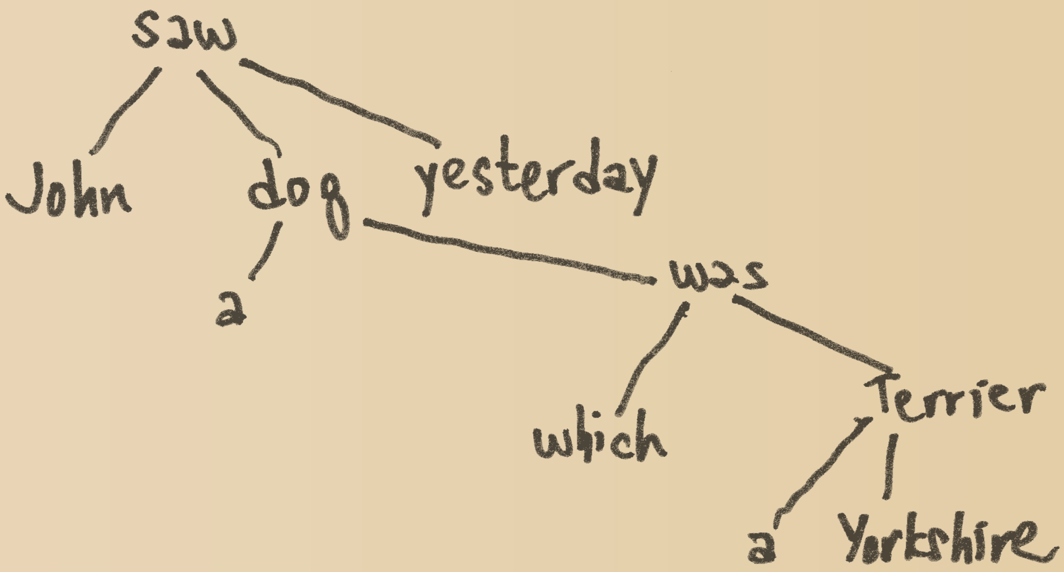
Non-Projective Trees

Real examples:





Whom Mark decided to marry?



John saw a dog yesterday which was a Yorkshire Terrier.

Course Outline

- various formats for phrase-structure and dependency trees, transformations
- mining information from the word layer and morphological layer of the Prague Dependency Treebank or the Prague English Dependency Treebank (bash, Perl or another programming language)
- mining information from all layers of the PDT/PEDT (btred, Perl)
- mining information from UD data (Udapi, Python)
- searching in treebanks with PML-Tree Query (PML-TQ)

Morphology

Jen osel se živí vlastní prací.

Morphology

Jen osel se živí vlastní prací.

Each word without context morphologically ambiguous, e.g.:

Jen – “only”

Jen – the Japanese currency

Morphology

Time flies like an arrow.

(source: Wikipedia)

Morphology

Time flies like an arrow.

As a generalisation, time passes in the same way that an arrow generally flies (i.e. quickly).

(source: Wikipedia)

Morphology

Time flies like an arrow.

(as an imperative) Measure the speed of flies like you would measure that of an arrow - i.e. (You should) time flies as you would time an arrow.

(imperative) Measure the speed of flies in the way an arrow would - i.e. (You should) time flies in the same manner that an arrow would time them.

(imperative) Measure the speed of flies with qualities resembling those of arrows - i.e. (You should) time those flies that are like an arrow.

(declarative) Time moves in a way an arrow would.

(declarative, i.e. neutrally stating a proposition) Certain flying insects, called "time flies," enjoy an arrow.

(source: Wikipedia)

Morphology

Time flies like an arrow.

(as an imperative) Measure the speed of flies like you would measure that of an arrow - i.e. (You should) time flies as you would time an arrow.

(imperative) Measure the speed of flies in the way an arrow would - i.e. (You should) time flies in the same manner that an arrow would time them.

(imperative) Measure the speed of flies with qualities resembling those of arrows - i.e. (You should) time those flies that are like an arrow.

(declarative) Time moves in a way an arrow would.

(declarative, i.e. neutrally stating a proposition) Certain flying insects, called "time flies," enjoy an arrow.

(source: Wikipedia)

Morphology

Time flies like an arrow.

(as an imperative) Measure the speed of flies like you would measure that of an arrow - i.e. (You should) time flies as you would time an arrow.

(imperative) Measure the speed of flies in the way an arrow would - i.e. (You should) time flies in the same manner that an arrow would time them.

(imperative) Measure the speed of flies with qualities resembling those of arrows - i.e. (You should) time those flies that are like an arrow.

(declarative) Time moves in a way an arrow would.

(declarative, i.e. neutrally stating a proposition) Certain flying insects, called "time flies," enjoy an arrow.

(source: Wikipedia)

Morphology

Time flies like an arrow.

(as an imperative) Measure the speed of flies like you would measure that of an arrow - i.e. (You should) time flies as you would time an arrow.

(imperative) Measure the speed of flies in the way an arrow would - i.e. (You should) time flies in the same manner that an arrow would time them.

(imperative) Measure the speed of flies with qualities resembling those of arrows - i.e. (You should) time those flies that are like an arrow.

(declarative) Time moves in a way an arrow would.

(declarative, i.e. neutrally stating a proposition) Certain flying insects, called "time flies," enjoy an arrow.

(source: Wikipedia)

Morphology

Time flies like an arrow.

(as an imperative) Measure the speed of flies like you would measure that of an arrow - i.e. (You should) time flies as you would time an arrow.

(imperative) Measure the speed of flies in the way an arrow would - i.e. (You should) time flies in the same manner that an arrow would time them.

(imperative) Measure the speed of flies with qualities resembling those of arrows - i.e. (You should) time those flies that are like an arrow.

(declarative) Time moves in a way an arrow would.

(declarative, i.e. neutrally stating a proposition) Certain flying insects, called "time flies," enjoy an arrow.

(source: Wikipedia)

Homographs

In English, there are also homographs:

- bow, tear, lead, wind, …

In Czech very rare:

- proudit [to flow] vs. pro~udit [to smoke meat thoroughly]

and possibly:

- neurologický vs. ne~urologický
- pře~dražený vs. před~ražený (Mám všechny jízdenky předražené.)

Layers of Language Description

Word Layer

Máma mlemaso.

Mom is mncingmeat.

Layers of Language Description

Word Layer

Máma mlemaso.

3 tokens:

- Máma
- mlemaso
- .

Mom is mncingmeat.

4 tokens:

- Mom
- is
- mncingmeat
- .

Layers of Language Description

Morphological Layer

Máma mlemaso.

Máma mele maso.
spell spell

Mom is mncingmeat.

Mom is mincing meat.
spell spell

Layers of Language Description

Morphological Layer

Máma mlemaso.

Máma mele maso.

spell spell

máma mlít maso .

Mom is mncingmeat.

Mom is mincing meat.

spell spell

mom to_be to_mince meat .

Layers of Language Description

Morphological Layer

Máma mlemaso.

Máma	mele	maso.	
	spell	spell	
máma	mlít	maso	.
noun	verb	noun	punct

Mom is mncingmeat.

Mom	is	mincing	meat.	
		spell	spell	
mom	to_be	to_mince	meat	.
noun	verb	verb	noun	punct

Layers of Language Description

Morphological Layer

Máma mlemaso.

Máma	mele	maso	.
	spell	spell	
máma	mlít	maso	.
noun	verb	noun	punct
NNFS1…	VB-S---3…	NNNS4…	Z:…

Mom is mncingmeat.

Mom	is	mincing	meat	.
		spell	spell	
mom	to_be	to_mince	meat	.
noun	verb	verb	noun	punct
NN	VBZ	VBG	NN	.

Layers of Language Description

Morphological Layer

MorphoDiTa – a morphological analyzer and tagger
https://ufal.mff.cuni.cz/morphodita
(morphological analysis vs. tagging)

You may try individual words:

- stát [several meanings]
- bank, state, fine

or whole sentences:

- Jen osel se živí vlastní prací.
- Time flies like an arrow.

NPFL075 Practical Class 01 Homework

https://ufal.mff.cuni.cz/~mirovsky/vyuka/NPFL075/2026/01_homework.html

- submit your homeworks via svn to your personal directories
- do not add data to the svn repository; if you want to automatize things, use wget
  - e.g. wget https://ufal.mff.cuni.cz/~mirovsky/vyuka/NPFL075/2026/01/train-1.zip
- submit the homework by March 16th (Monday)

NPFL075 Practical Class 01
SVN checkout

Initial checkout of your personal directory:

```
svn co --username $USERNAME
https://svn.ms.mff.cuni.cz/svn/undergrads/students/$USERNAME .
```

In your checkouted directory, there is the subdirectory NPFL075, please put your hw there.

NPFL075 Practical Class 01
SVN work cycle

svn work cycle (after the initial checkout):

- svn update
- svn add [your new files/directories]
- svn commit -m "a short descriptive message"

NPFL075 Practical Class 01
Homework

Let us inspect the data for the homework

https://ufal.mff.cuni.cz/~mirovsky/vyuka/NPFL075/2026/01/train-1.zip