# Dependency Grammars and Treebanks:

## Intro – trees, word order, projectivity

Markéta Lopatková, Daniel Zeman, Jiří Mírovský

Institute of Formal and Applied Linguistics, MFF UK

lopatkova@ufal.mff.cuni.cz

# Dependency Grammars and Treebanks (NPFL075)

Lectures: Wednesday, room S1, 15:40-17:10

Markéta Lopatková, Daniel Zeman

Practical sessions:

Jiří Mírovský, Daniel Zeman

http://ufal.mff.cuni.cz/course/npfl075

Requirements:

- Homework (40%)
- Activity (10%)
- Final test (50%)

Assessment:

- excellent (= 1) ≥ 90%
- very good (= 2) ≥ 70%
- good (= 3) ≥ 50%

# Dependency Grammars and Treebanks

- Family of Prague Dependency Treebanks (PDT, PCEDT)
- Universal Dependencies
- HamleDT, PropBank, ???

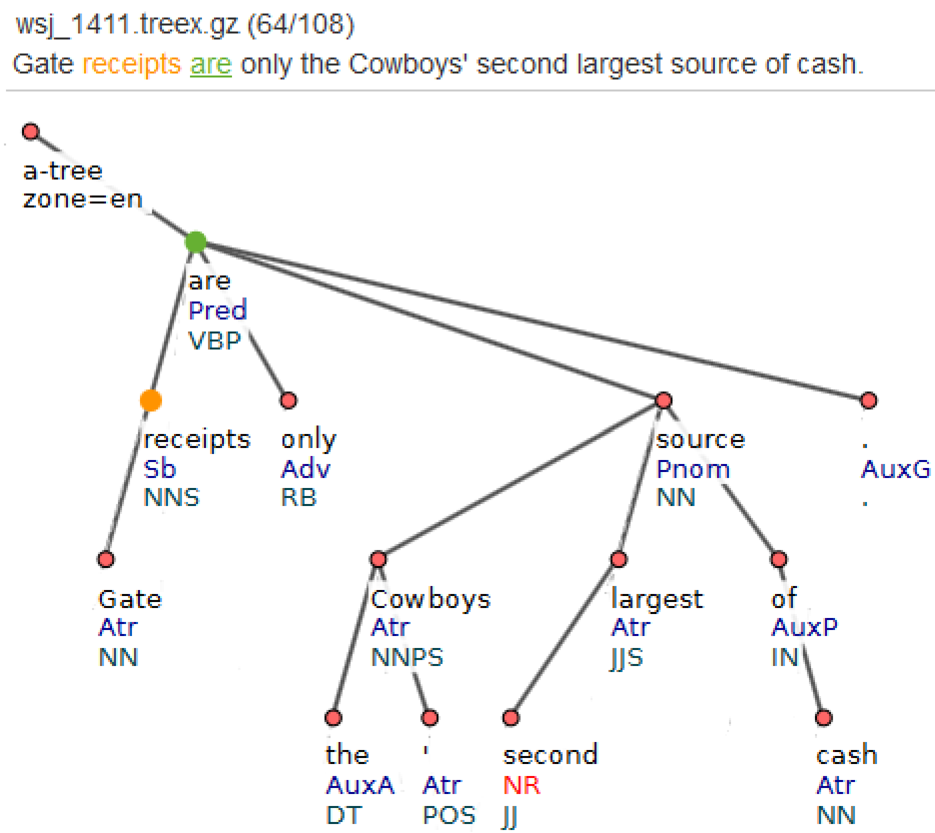
## Collection of:

- linguistically annotated data
- tools and data format(s)
- documentation

Another point of view:

- annotation scheme
- framework for annotation of different languages
- underlying linguistic theory

# How to capture sentence structure?



# Graph theory: tree

*tree* (graph theory):
definiftion:

- finite graph $\langle N, E \rangle$, N ~ nodes/vertices, E ~ edges $\{n_1,n_2\}$
- connected
- no cycles, no loops
- no more than 1 edge between any two different nodes

$\Leftrightarrow$ (undirected) graph
any two nodes are connected by exactly one simple path

*rooted tree*

- rooted $\Rightarrow$ orientation (i.e., edges ordered pairs $[n_1,n_2]$)

*directed tree* … directed graph

- which would be tree
  - if the directions on the edges were ignored, or
  - all edges are directed towards a particular node ~ the ***root***

# Data structure: tree

*tree as a data structure*:

- rooted tree (as in graph theory)
- all edges are directed from a particular node ~ the ***root***

**+**

- (linear) ordering of nodes:
  the children of each node have a specific order

# Data structure: tree (properties)

*tree as a data structure*:

- "tree-ordering" D … partial ordering on nodes
  $u \leq v \;\Leftrightarrow_{\text{def}}$ the unique path from the root to *v* passes through *u*
  (weak ordering ~ reflexive, antisymmetric, transitive)

- "linear ordering" … (partial) ordering on nodes
  (strong ordering ~ antireflexive, asymmetric, transitive)

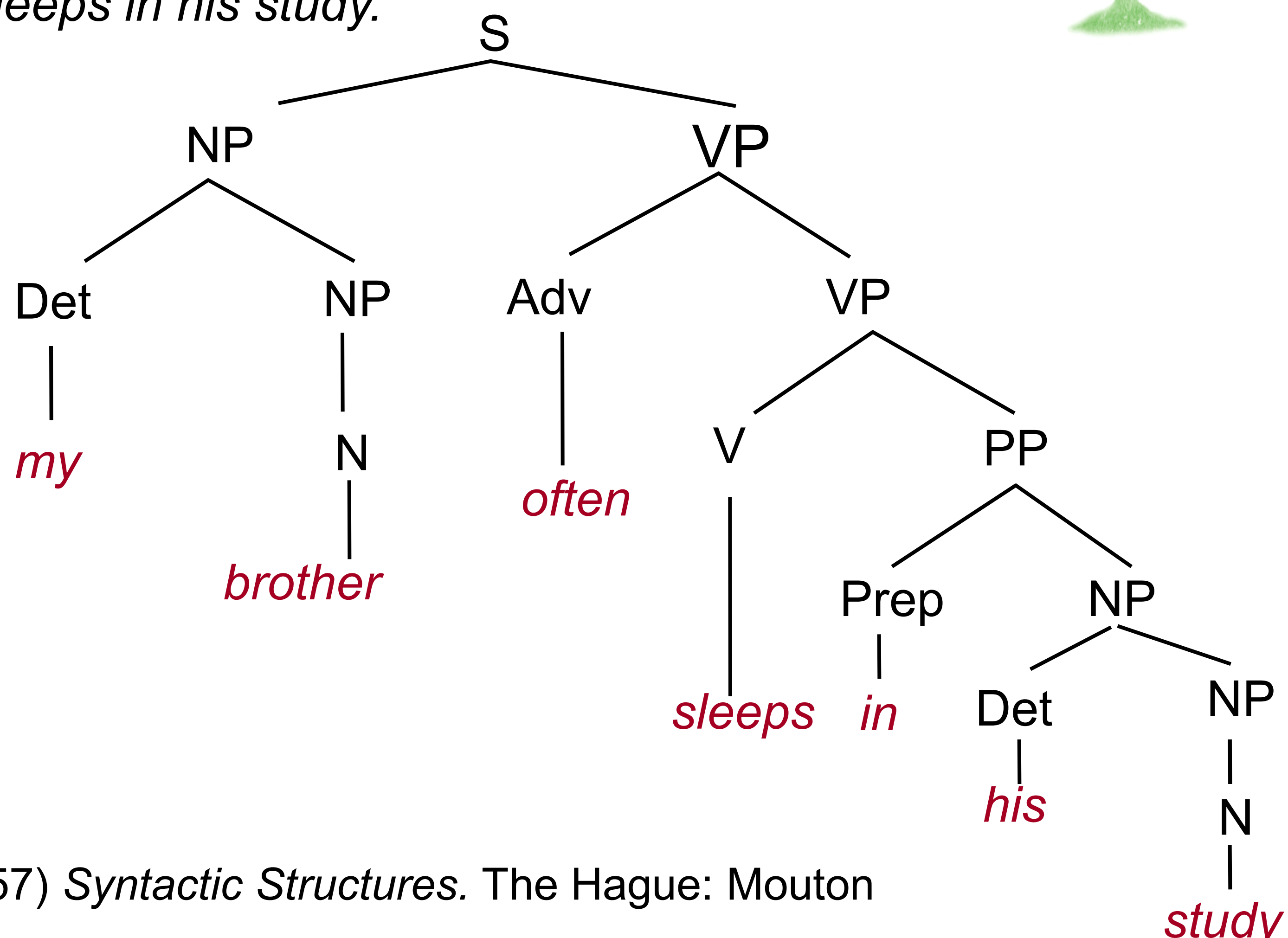
# Tree-based structures in CL

two types of tree-based structures in CL:

- phrase structure tree / constituent structure tree
- dependency tree

# Phrase structure tree

*My brother often sleeps in his study.*




Noam Chomsky (1957) *Syntactic Structures.* The Hague: Mouton

# Phrase structure tree (definition)

*$T = \langle N, D, Q, P, L \rangle$*

$\langle$N, D$\rangle$ … ***rooted tree, directed***

Q ... lexical and grammatical categories

L … labeling function N $\rightarrow$ Q

D … oriented edges (branches)

~ relation on lex. and gram. categories

*dominance relation*

**+**

P ... relation on N ~ (partial strong linear ordering)

relation of *precedence*

# Phrase structure tree (definition)

*$T = \langle N, D, Q, P, L \rangle$*

$\langle N, D \rangle$ … ***rooted tree, directed***

Q ... lexical and grammatical categories

L … labeling function $N \rightarrow Q$

D … oriented edges (branches)

~ relation on lex. and gram. categories

*dominance relation*

**+**

P ... relation on N ~ (partial strong linear ordering)

relation of *precedence*

**+**

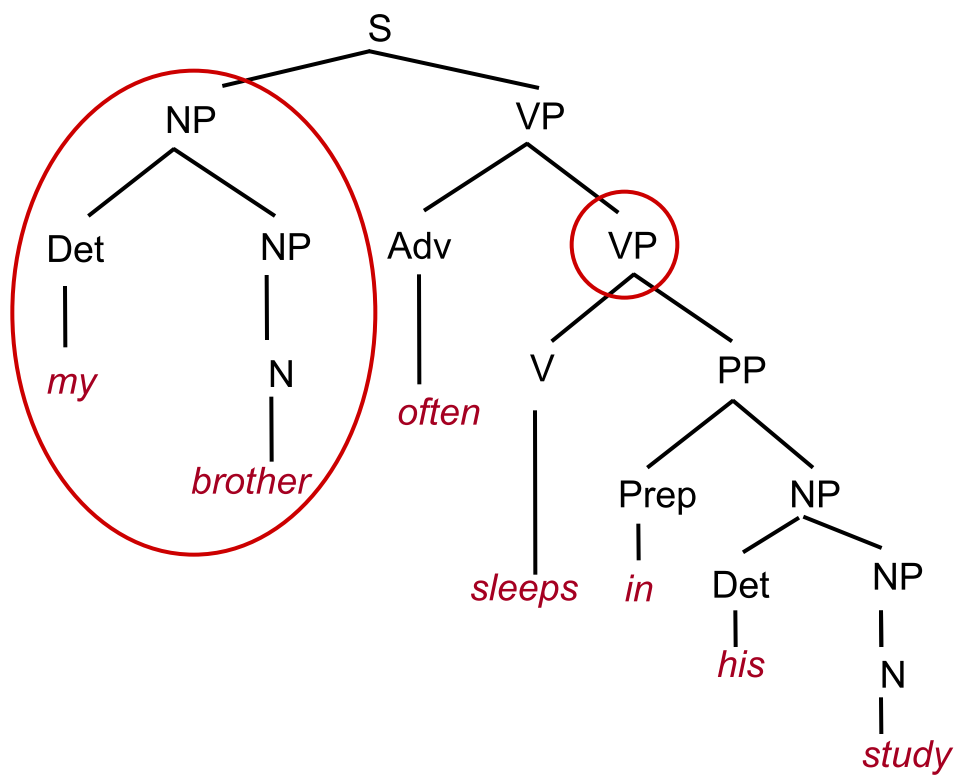
Relating dominance and precedence relations:

- *exclusivity* condition for D and P relations
- *'nontangling'* condition

# Phrase structure tree (relation P)

- *exclusivity* condition for D and P relations

  $\forall\, x,y \in N$ holds: $(\, [x,y] \in P \vee [y,x] \in P\,) \Leftrightarrow (\, [x,y] \notin D \;\&\; [y,x] \notin D)$



# Phrase structure tree (relation P)

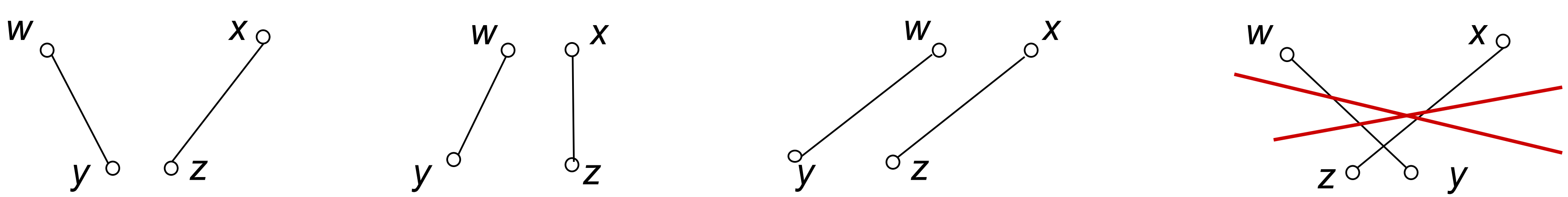
- *exclusivity* condition for D and P relations

  $\forall\ x,y \in N$ holds: $(\ [x,y] \in \mathrm{P} \vee [y,x] \in \mathrm{P}\ ) \Leftrightarrow (\ [x,y] \notin \mathrm{D}\ \&\ [y,x] \notin \mathrm{D})$

- *'nontangling'* condition

  $\forall\ w,x,y,z \in N$ holds: $(\ [w,x] \in \mathrm{P}\ \&\ [w,y] \in \mathrm{D}\ \&\ [x,z] \in \mathrm{D}\ )$

  $\Rightarrow (\ [y,z] \in \mathrm{P}\ )$



# Phrase structure tree (relation P)

- *exclusivity* condition for D and P relations

  $\forall\ x,y \in N$ holds: $(\ [x,y] \in \mathrm{P} \vee [y,x] \in \mathrm{P}\ ) \Leftrightarrow (\ [x,y] \notin \mathrm{D}\ \&\ [y,x] \notin \mathrm{D})$

- *'nontangling'* condition

  $\forall\ w,x,y,z \in N$ holds: $(\ [w,x] \in \mathrm{P}\ \&\ [w,y] \in \mathrm{D}\ \&\ [x,z] \in \mathrm{D}\ )$

  $\Rightarrow (\ [y,z] \in \mathrm{P}\ )$

$\Rightarrow$

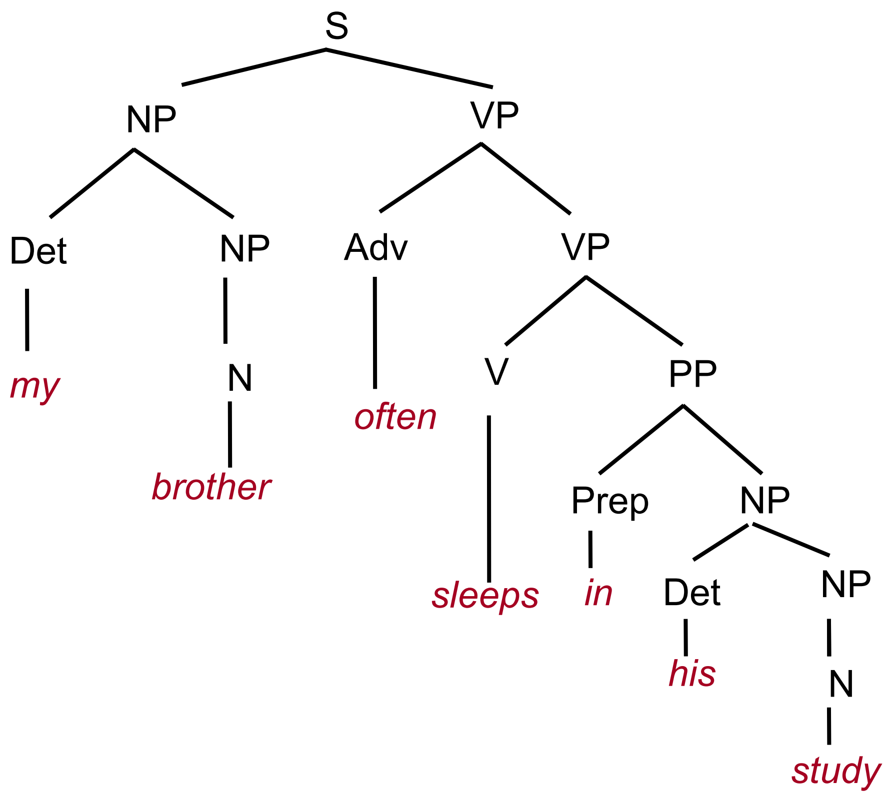
$\mathrm{T} = \langle\ \mathrm{N,D,Q,\ P,L}\ \rangle$ phrase structure tree

- $\forall\ \ x,y \in \mathrm{N}$ siblings $\Rightarrow [x,y\,] \in \mathrm{P}$
- the set of its leaves is totally ordered by P

# Phrase structure tree

## Pros

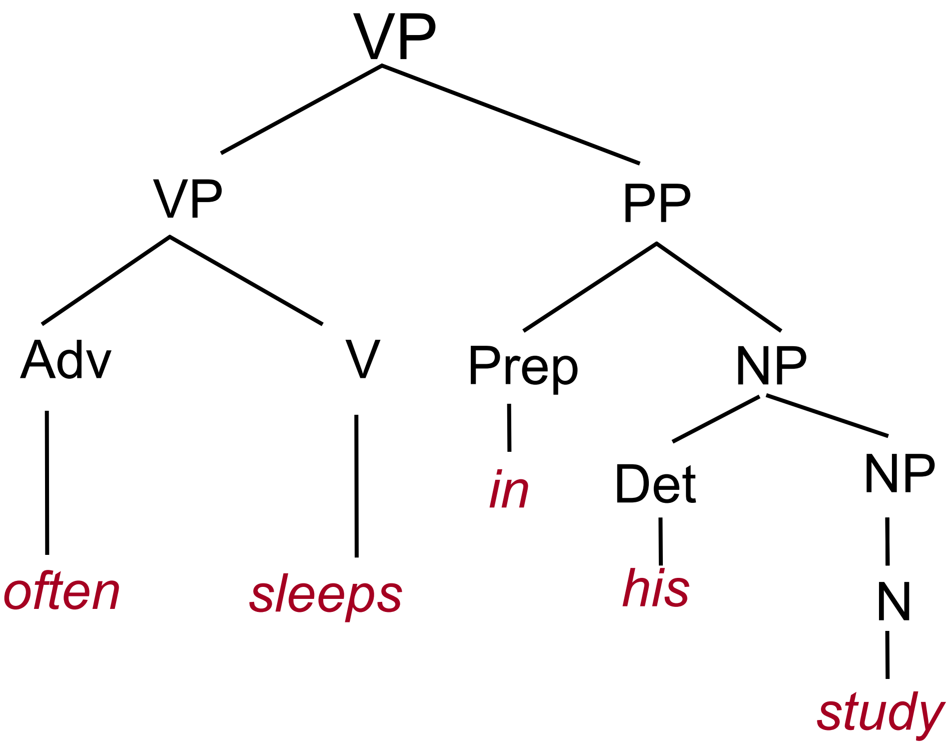
- derivation history / 'closeness' of a complementation
- coordination, apposition
- CFG-like
- derivation of a grammar



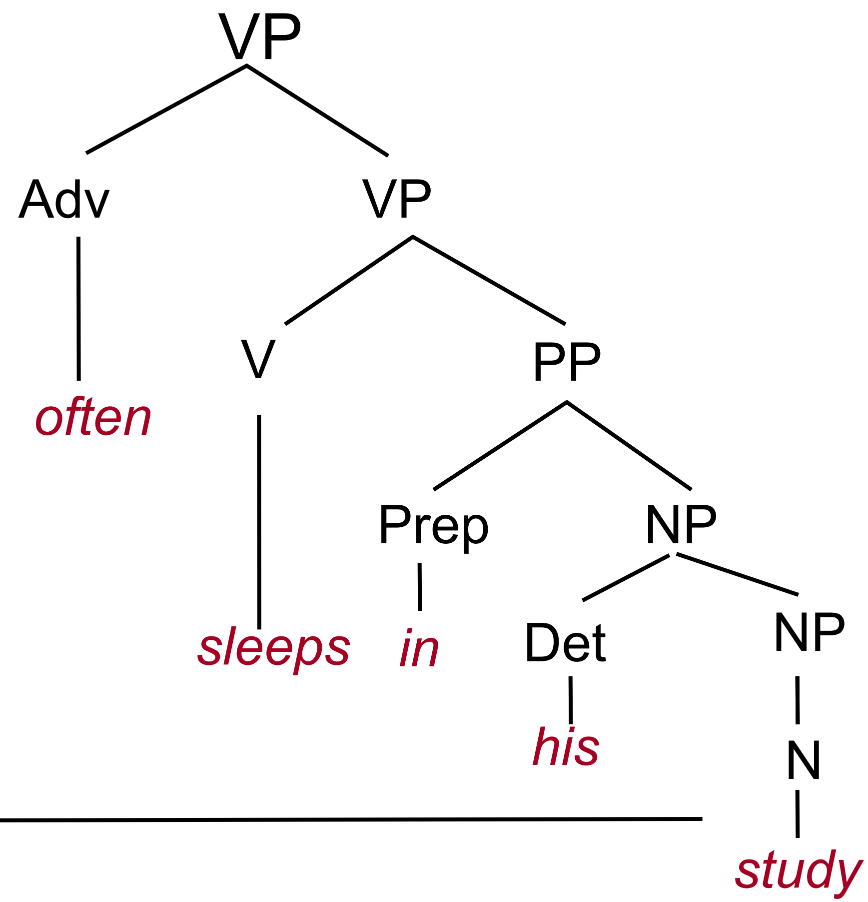
# Phrase structure tree

derivation history / ‘closeness’:

*… often sleeps in his study*




*… often sleeps in his study*




# Phrase structure tree

## Pros

- derivation history / ‘closeness’ of a complementation
- coordination, apposition
- CFG-like
- derivation of a grammar

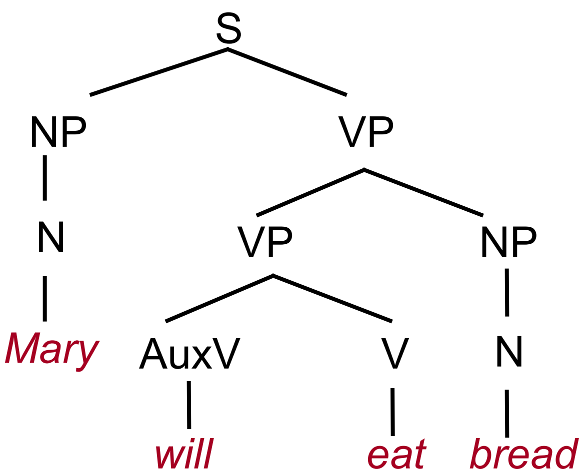
## Contras

- complexity
  (number of non-terminal symbols)
- complement
  (‘two dependencies’)
  *přiběhl bos*
  [(he) arrived barefooted]
- free word order
  discontinuous ‘phrases’
  non-projectivity

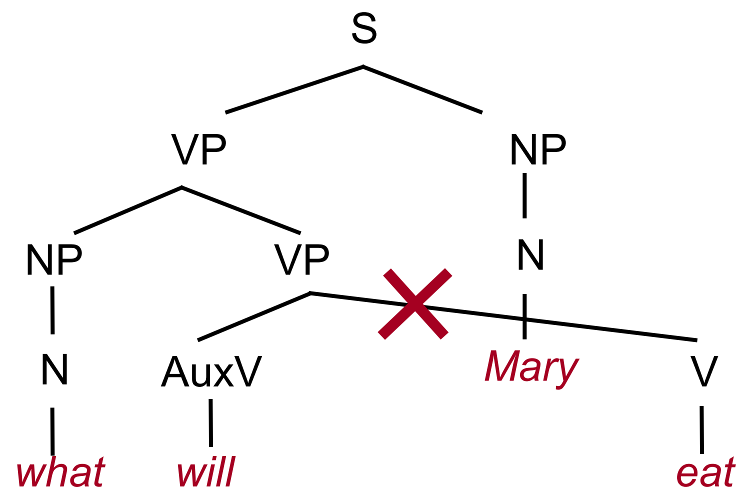
# Phrase structure tree

## discontinuous ‘phrases’: solution for English

*Mary will eat bread.*



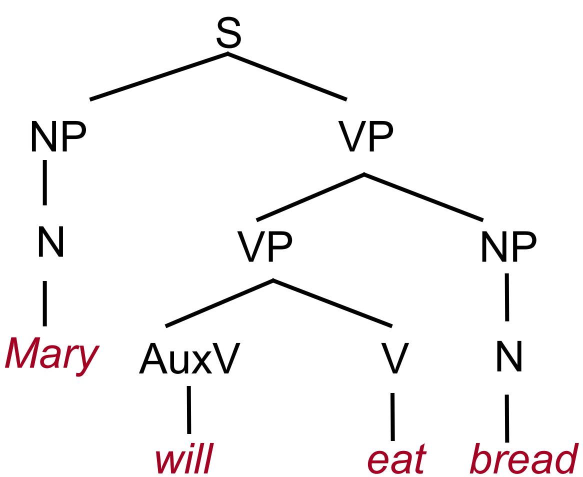

*What will Mary eat?*



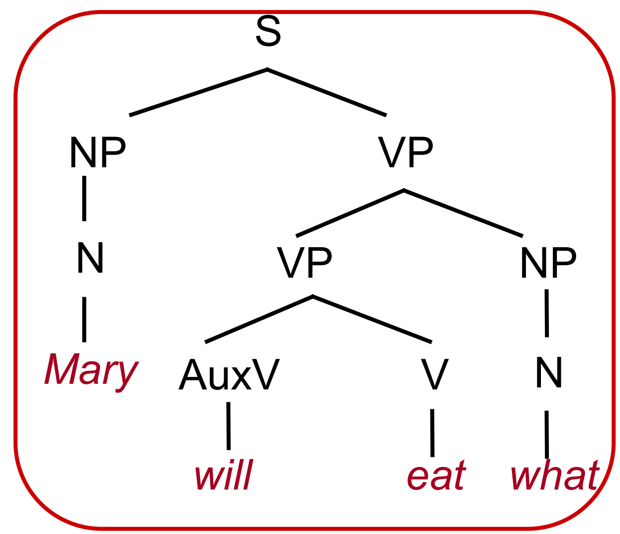
# Phrase structure tree

discontinuous ‘phrases’: solution for English

*Mary will eat bread.*

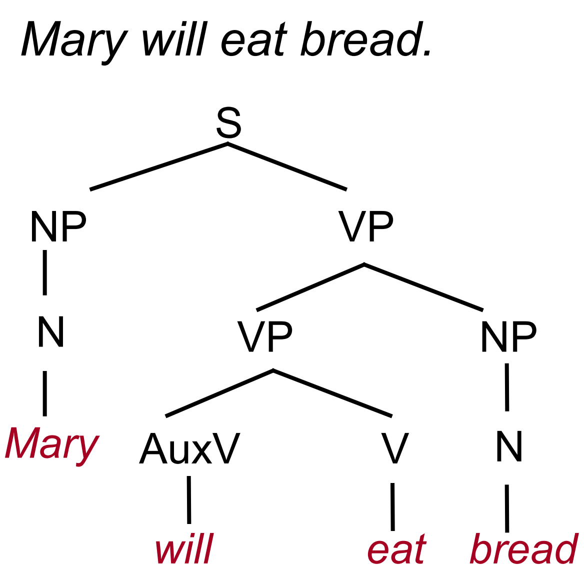


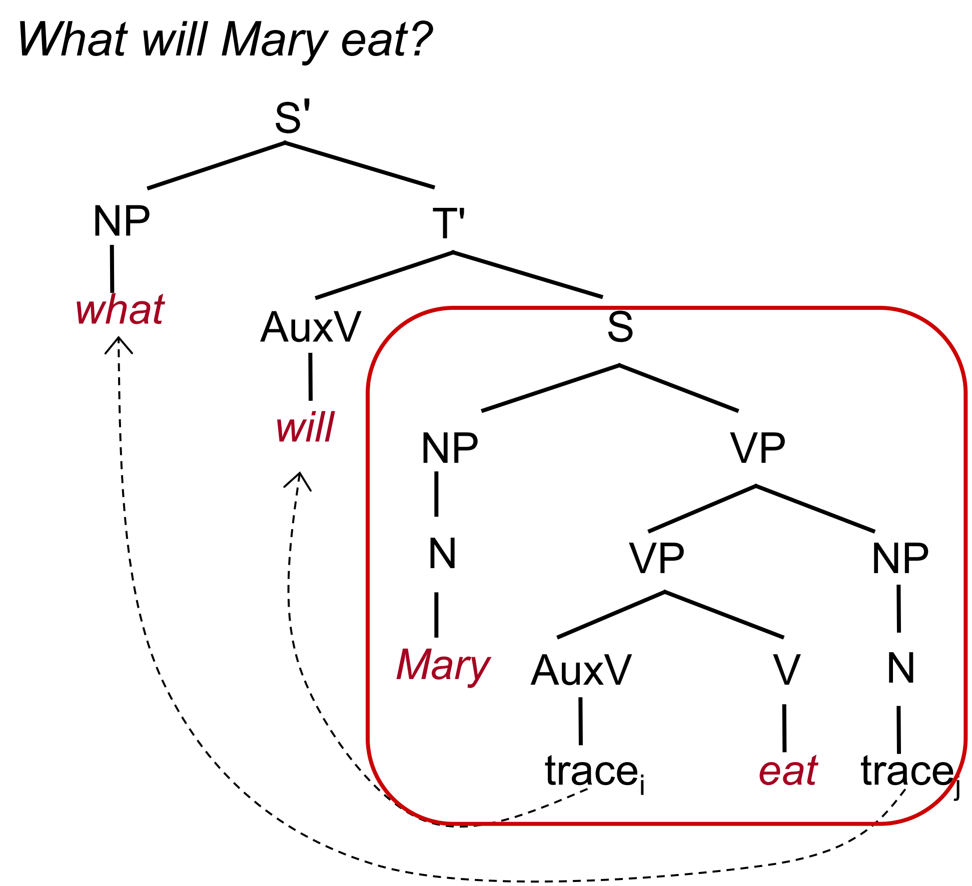

*What will Mary eat?*



# Phrase structure tree

discontinuous ‘phrases’: solution for English

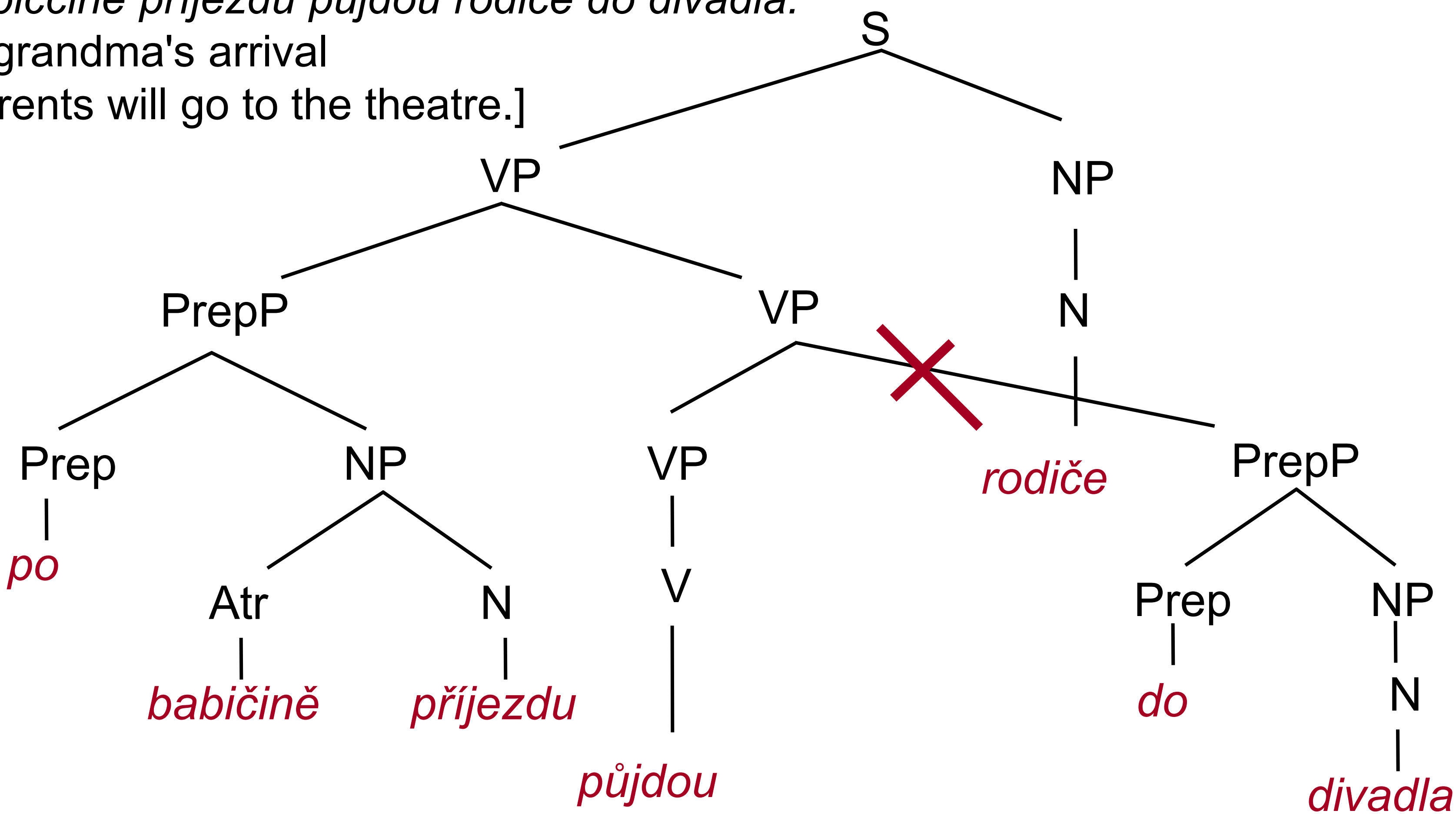


# Phrase structure tree

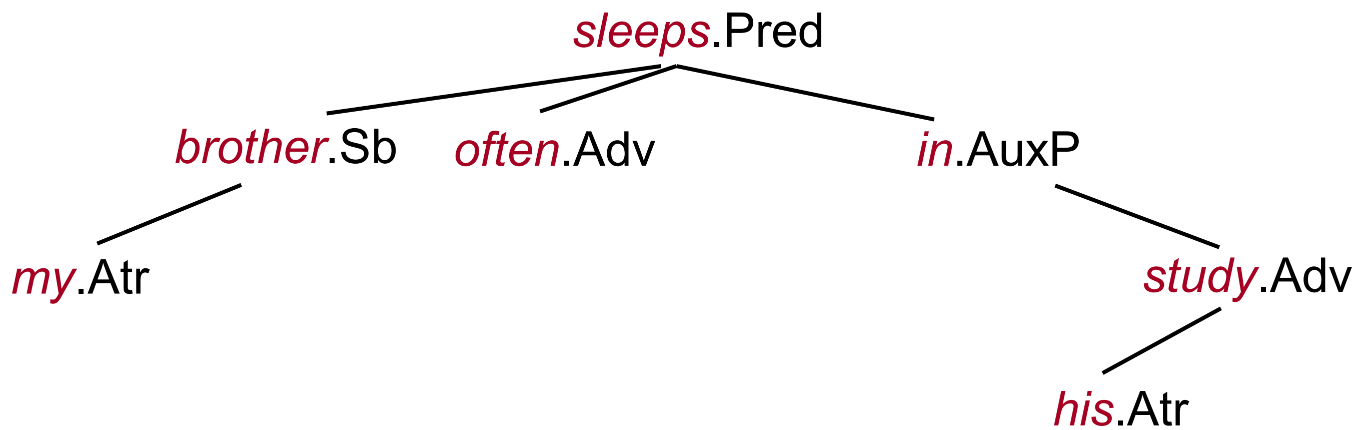
discontinuous ‘phrases’:

*Po babiččině příjezdu půjdou rodiče do divadla.*
[After grandma's arrival
the parents will go to the theatre.]



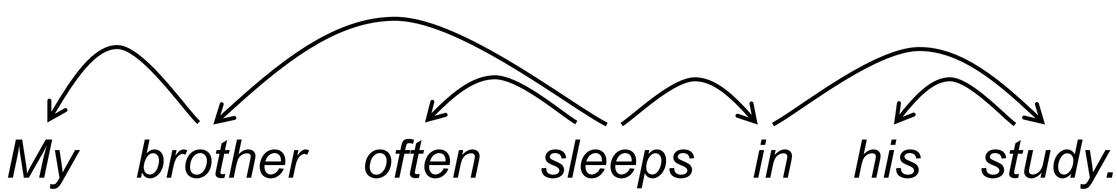
# Dependency tree

*My brother often sleeps in his study.*




Lucien Tesnière (1959) *Éléments de syntaxe structurale.* Editions Klincksieck.

Igor Mel'čuk (1988) *Dependency Syntax: Theory and Practice.* State University of New York Press.



# Dependency tree (definition)

*T* = ⟨ *N, D, Q, WO, L* ⟩

⟨N, D⟩ … ***rooted tree, directed***

Q ... lexical and grammatical categories

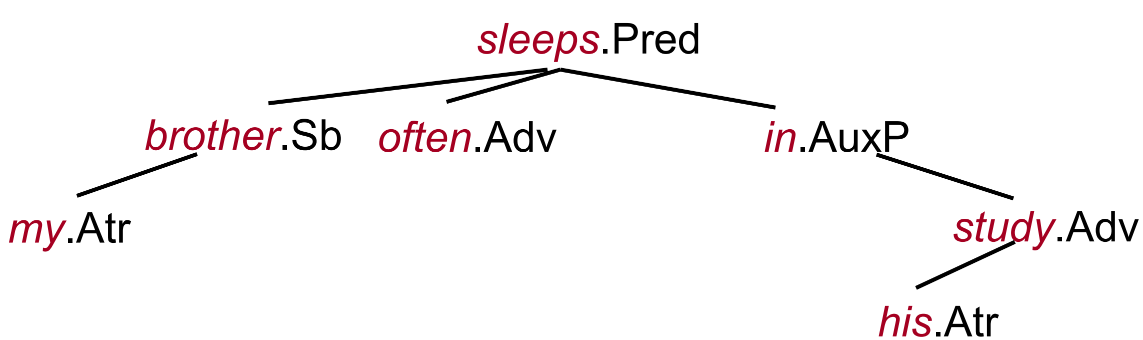
L … labeling function $N \rightarrow Q^{+}$

D … oriented edges ~ relation on lex. and gram. categories

*‘dependency’ relation*

WO ...relation on N ~ (strong total ordering on N) …

*word order*



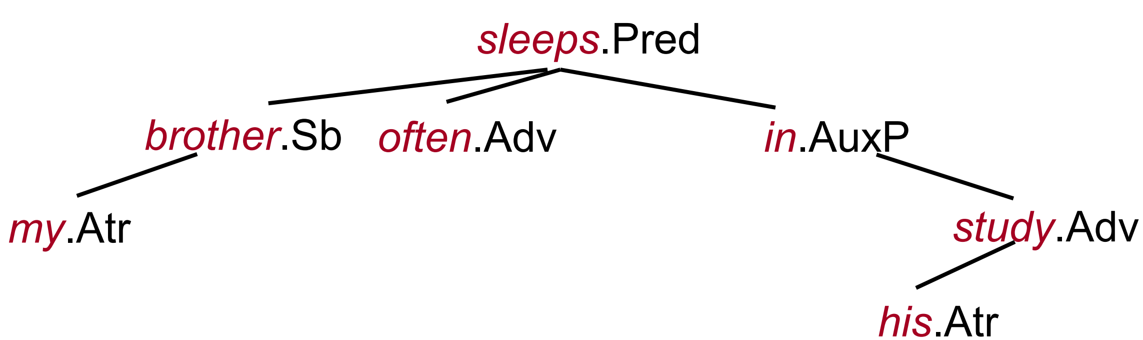
# Dependency tree

## Pros

- economical, clear
  (complex labels, ‘word’~ node)
- free word order
- head of a phrase

## Contras

- no derivation history / 'closeness'
- coordination, apposition
- complement

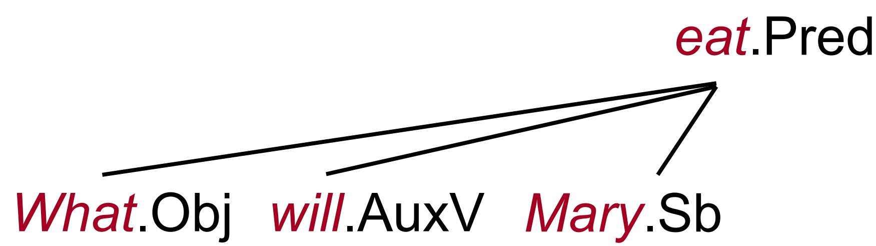


# Dependency tree

discontinuous ‘phrases’: no problem

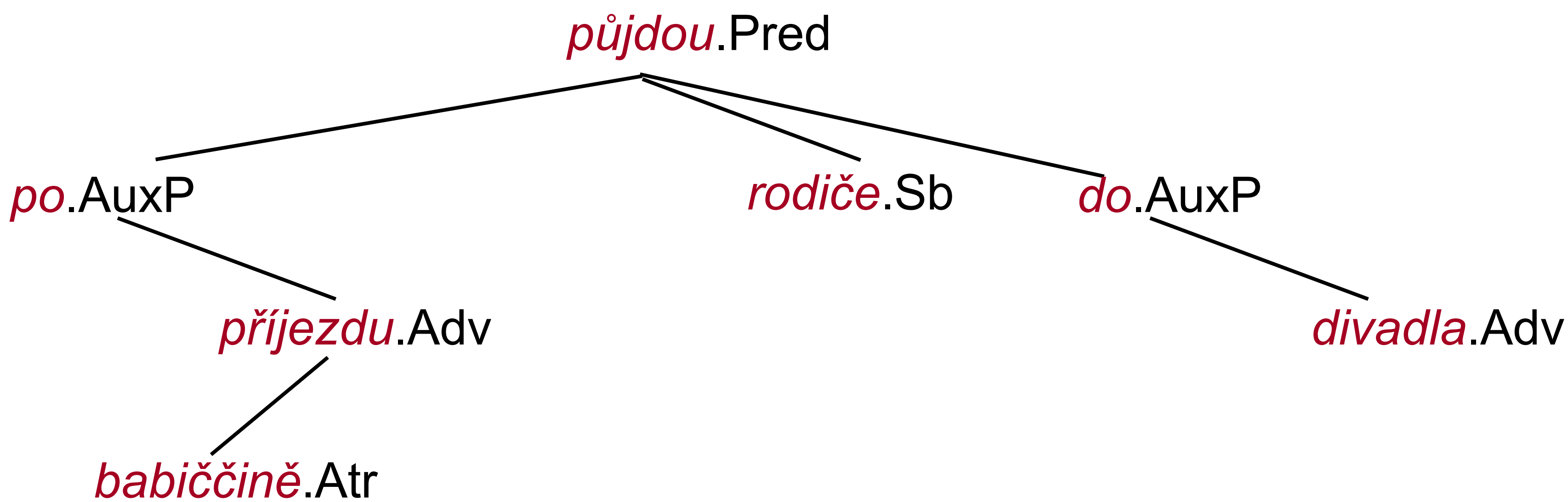
*Mary will eat bread.*


*What will Mary eat?*



# Dependency tree

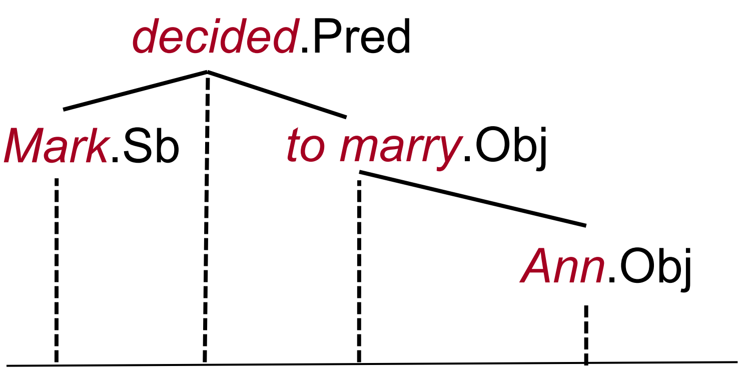
*Po babiččině příjezdu půjdou rodiče do divadla.*
[After grandma's arrival the parents will go to the theatre.]

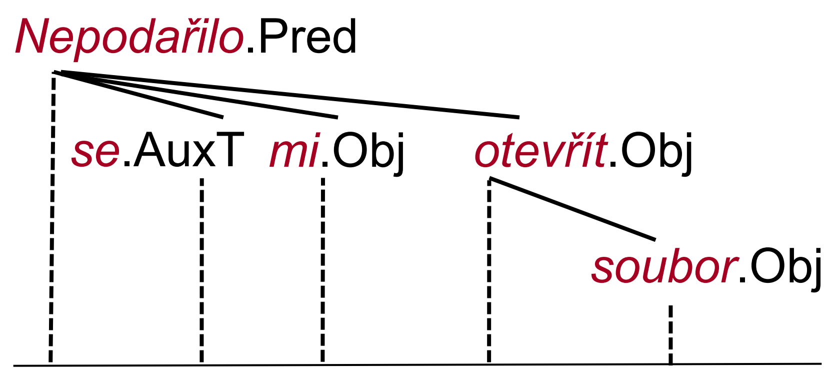


# Projectivity and non-projectivity (definition)

Mark decided to marry Ann.

*Nepodařilo se mi otevřít soubor.*

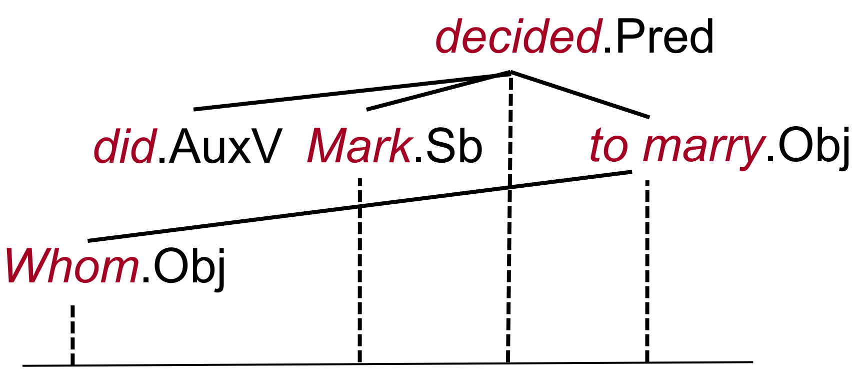


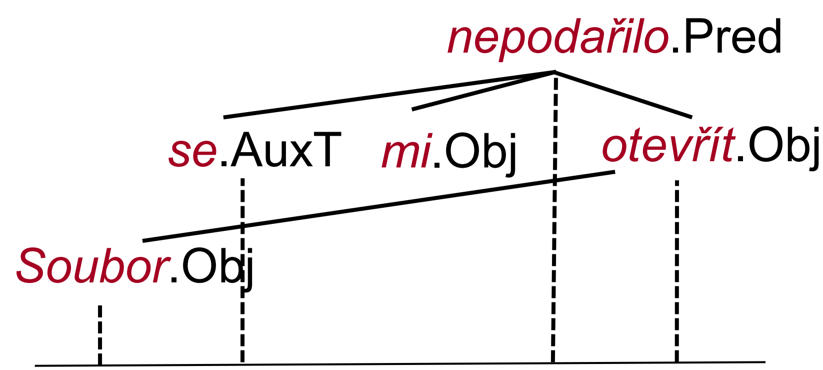



# Projectivity and non-projectivity (definition)

Whom did Mark decided to marry?

*Soubor se mi nepodařilo otevřít.* (Oliva)



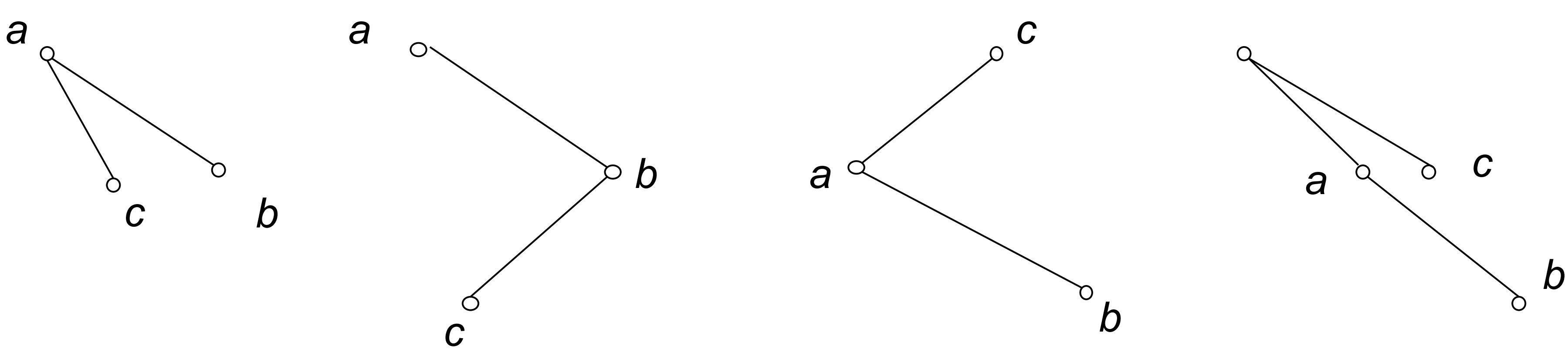
# Projectivity and non-projectivity (definition)

A subtree *S* of a rooted dependency tree *T* is *projective* iff for all nodes *a*, *b* and *c* of the subtree *S* the condition holds:

$$(1)\quad (a \leq_D b) \;\&\; (a <_{WO} c <_{WO} b) \;\Rightarrow\; (a <_D^{*} c)$$

<u>and</u>

$$(2)\quad (a \leq_D b) \;\&\; (b <_{WO} c <_{WO} a) \;\Rightarrow\; (a <_{WO}^{*} c)$$



# Projectivity and free word order

## free word order:

- freedom of word order of dependents within a <u>continuous</u> ‘head domain’ (i.e., substring of head + its dependents)
- <u>relaxation of continuity</u> of a head domain

German:
*Maria hat einen <u>Mann</u> kennengelernt der Schmetterlinge <u>sammelt</u>.*
Mary has a man met the butteries collects
‘Mary has met a man who collects butteries.’

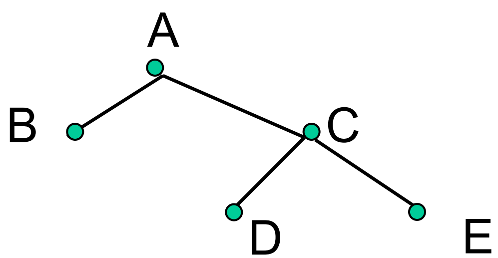
English: long-distance unbounded dependency
*John, Peter thought that Sue said that Mary loves.*

Czech*:*
*Marii se Petr tu knihu rozhodl nekoupit.*
to-Mary PART Peter that book decided not-buy
‘Peter decided not to buy that book to Mary.’

# Projectivity and non-projectivity

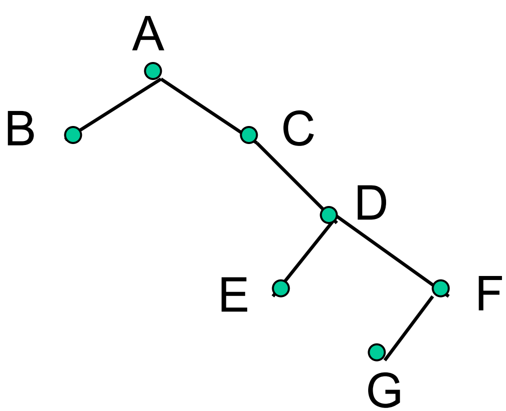
Projective dependency trees can be encoded by *linearization*:

- string of nodes, edges ~ brackets




A ( B C ( D E ) ) without WO ordering

( B ) A ( ( D ) C ( E ) ) with WO




A ( B C ( D ( E F ( G ) ) ) ) without WO

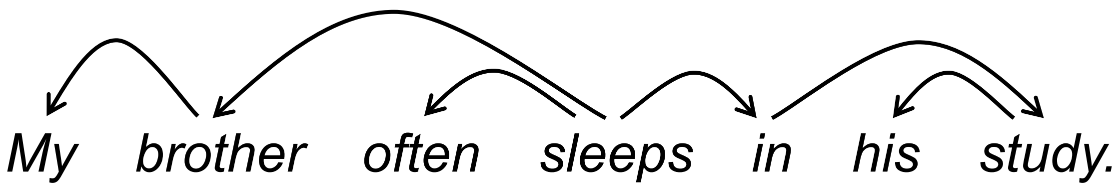
( B ) A ( C ( ( E ) D ( ( G ) F ) ) ) with WO

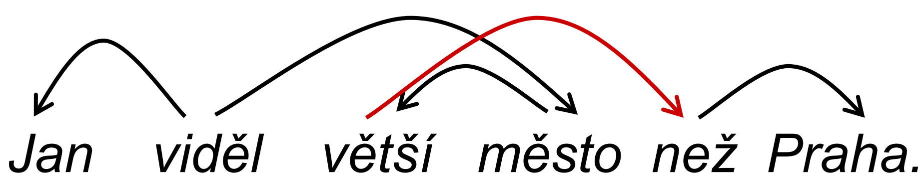
# Planarity

A dependency graph *T* is *planar*, if it does *not* contain nodes *a, b, c, d* such that:

$$linked(a,c) \ \ \& \ \ linked(b,d) \ \ \& \ \ a <_{WO} b <_{WO} c <_{WO} d$$

*linked(i,j)* … 'there is an edge in *T* from *i* to *j*, or vice versa'



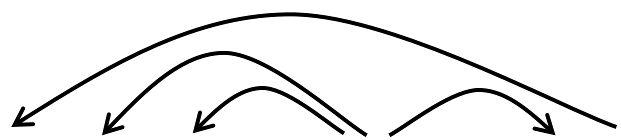

Informally, a dependency graph is planar, if its edges can be drawn above the sentence without crossing.

# Planarity vs. projectivity

projectivity $\Rightarrow$ planarity

projectivity $\not\Leftarrow$ planarity

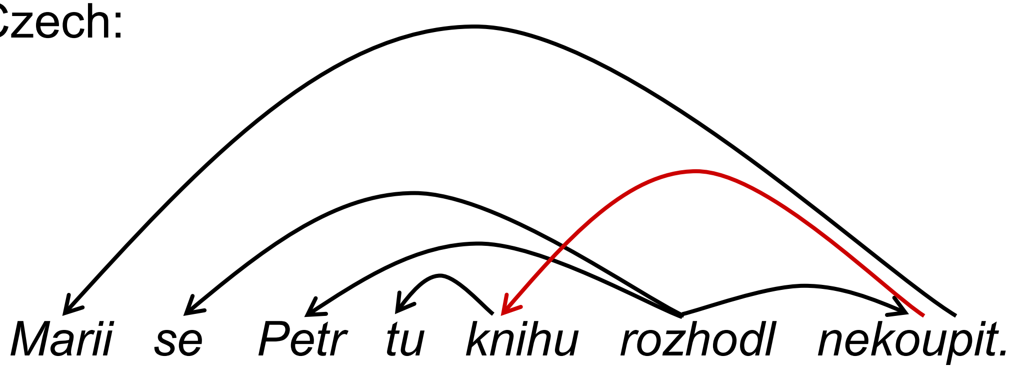
(Kuhlmann, M., Nivre, J., 2006)



*Soubor se mi nepodařilo otevřít .*

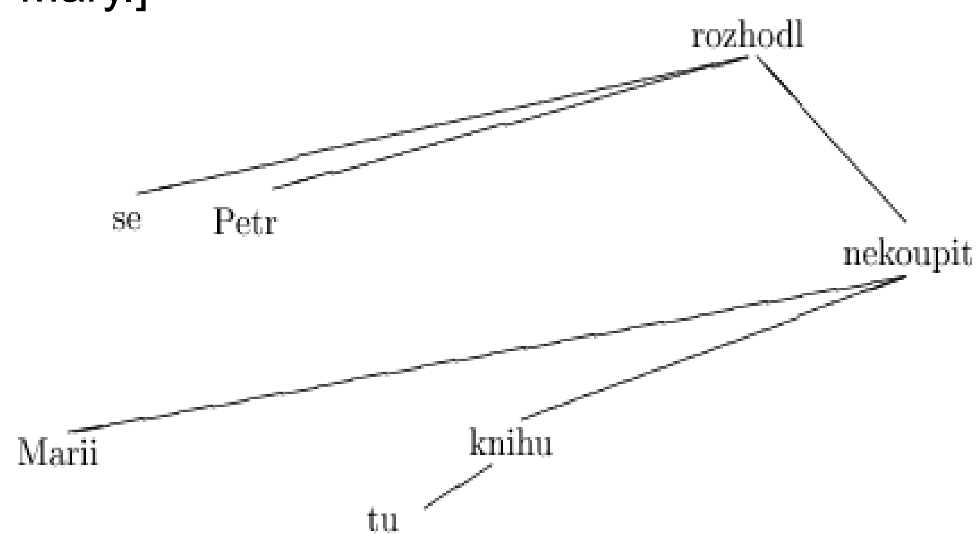
# Projectivity and free word order

Czech:




to-Mary PART Peter that book decided not-buy

[Peter decided not to buy that book to Mary.]

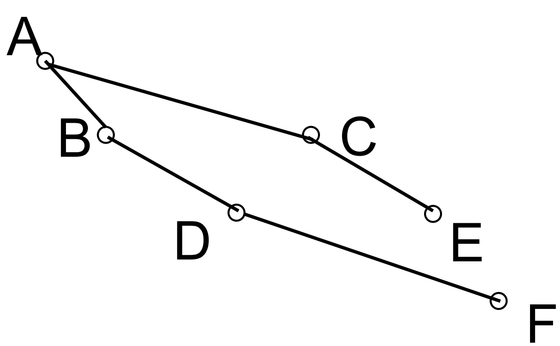


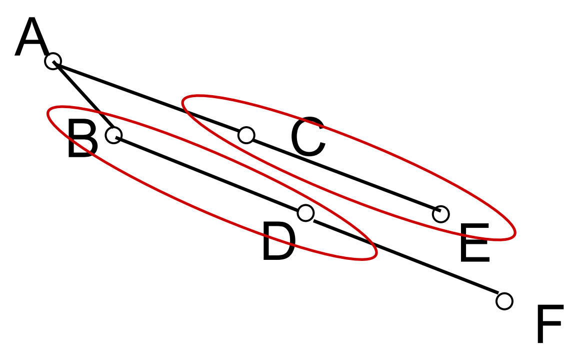
# ‘Well-Nestedness’

Two subtrees $T_1$, $T_2$ *interleave*, if there are nodes $l_1, r_1 \in T_1$ and $l_2, r_2 \in T_2$ such that

$$l_1 <_{WO} l_2 <_{WO} r_1 <_{WO} r_2$$

A dependency graph is *well-nested*, if no two of its disjoint subtrees interleave.’



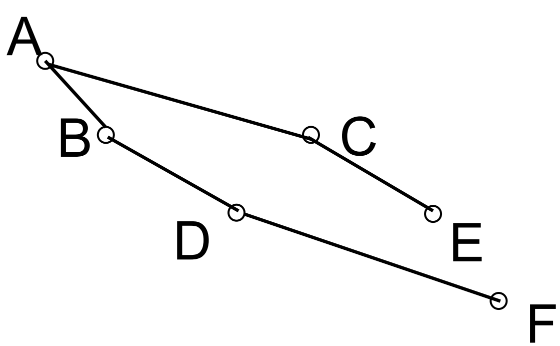



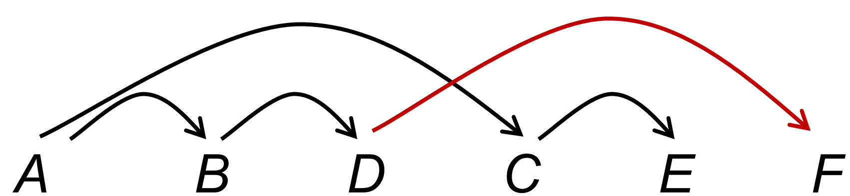
# Planarity vs. projectivity

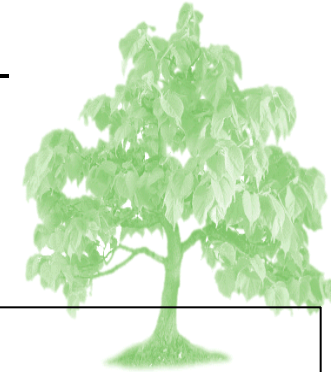
projectivity $\Rightarrow$ planarity $\Rightarrow$ well-nestedness

projectivity $\not\Leftarrow$ planarity $\not\Leftarrow$ well-nestedness

(Kuhlmann, M., Nivre, J., 2006)



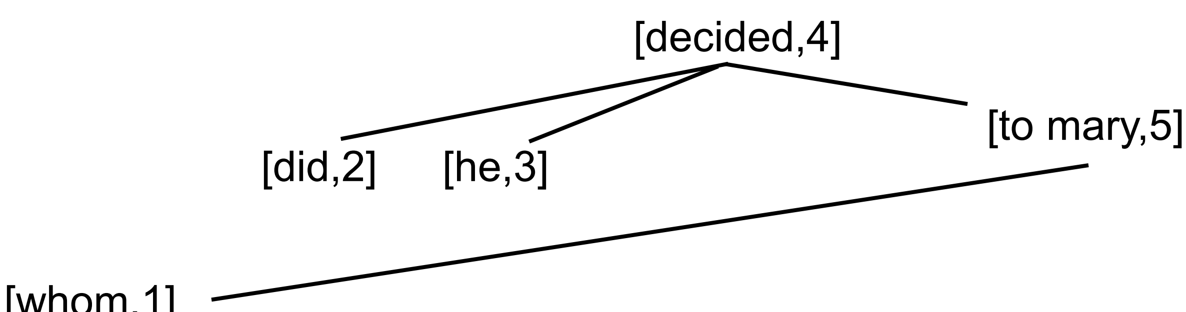


# Gap Degree *dNh(T)*

*Coverage* of a node $u \in T$

$Cov(u,T)$ = { $i$ | $i$ - word order position of $v \in T$ such that, u $\leq_D$ v }

$Cov(u_1,T)=\{1\}$; $Cov(u_2,T)=\{2\}$; $Cov(u_3,T)=\{3\}$; $Cov(u_4,T)=\{1,2,3,4,5\}$; $Cov(u_5,T) = \{1,5\}$



# Gap Degree *dNh(T)*

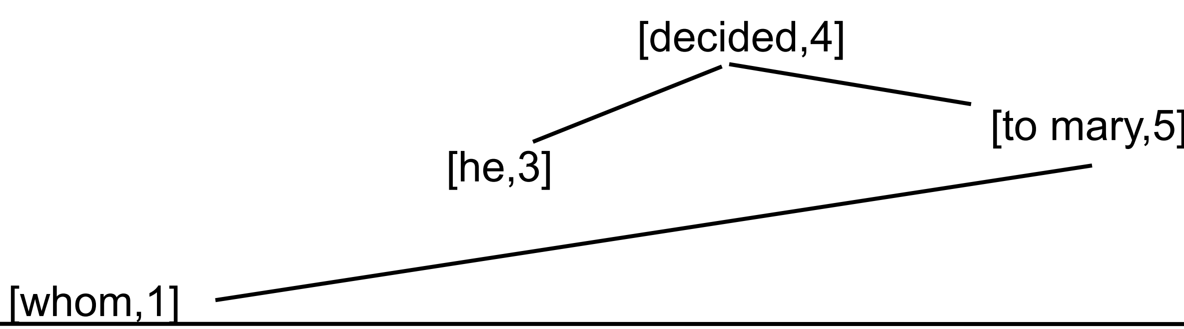
Coverage of a node $u \in T$

$Cov(u,T) = \{\ i \mid i$ - word order position of $v \in T$ such that, $u \leq_D v\ \}$

Gap in Coverage of a node $u \in T$ $\Leftrightarrow_{def}$ $Cov(u,T)$ is not an interval

$dNh(u,T)$ … *number of Gaps* in $Cov(u,T)$

$Cov(u_1,T)=\{1\}$; $Cov(u_2,T)=\{2\}$; $Cov(u_3,T)=\{3\}$; $Cov(u_4,T)=\{1,2,3,4,5\}$; $Cov(u_5,T) = \{1,5\}$



# Gap Degree *dNh(T)*

Coverage of a node $u \in T$

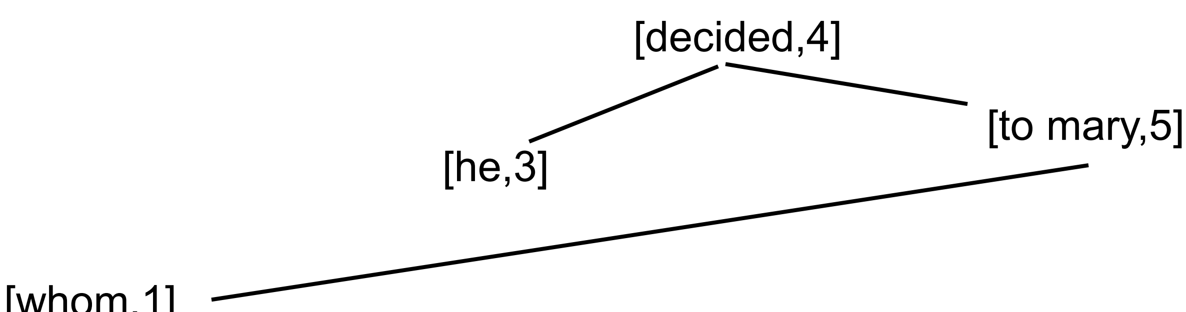
$Cov(u,T)$ = { $i$ | $i$ - word order position of $v \in T$ such that, $u \leq_D v$ }

Gap in Coverage of a node $u \in T$ $\Leftrightarrow_{def}$ $Cov(u,T)$ is not an interval

$dNh(u,T)$ … *number of Gaps* in $Cov(u,T)$

Tree Gap Degree $dNh(T) = \max \{dNh(u,T) | u \in T\}$

$Cov(u_1,T)=\{1\}$; $Cov(u_2,T)=\{2\}$; $Cov(u_3,T)=\{3\}$; $Cov(u_4,T)=\{1,2,3,4,5\}$; $Cov(u_5,T) = \{1,5\}$



# Gap Degree *dNh(T)*

*Coverage* of a node $u \in T$

$Cov(u,T)$ = { $i$ | $i$ - word order position of $v \in T$ such that, $u \leq_D v$ }

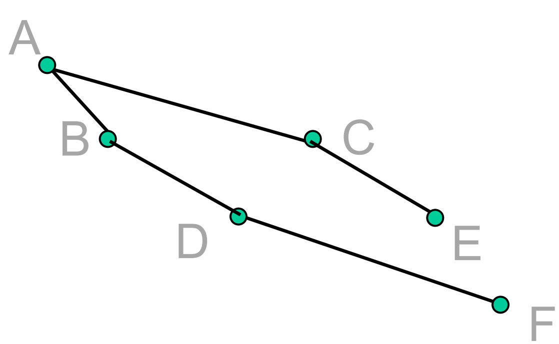
*Gap in Coverage* of a node $u \in T$ $\Leftrightarrow_{def}$ $Cov(u,T)$ is not an interval

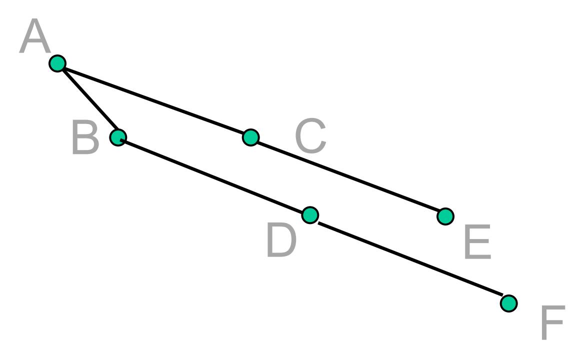
$gd(u,T)$ … *number of Gaps* in $Cov(u,T)$

*Tree Gap Degree* $gd(T) = \max\{gd(u,T) \mid u \in T\}$

# Edge Degree

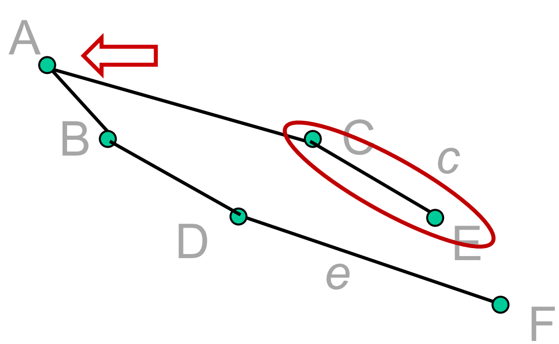
Let $T = (N, E)$ dependency tree, $e = [i, j]$ an edge in $E$,
$T_e$ the subgraph of $T$ induced by the nodes contained in the span of $e$.
***Degree of an edge*** $e \in E$, ***ed(e)***, is the number of connected components $c$ in $T_e$ such that the root of $c$ is not dominated by the head of $e$.
***Edge degree of T, ed(T)*** … max $\{ed(e) \mid e \in T\}$

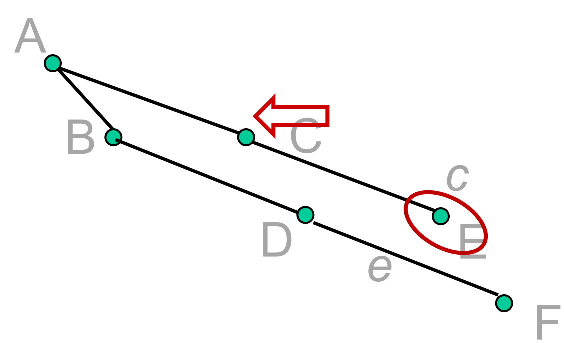


# Edge Degree

Let $T = (N, E)$ dependency tree, $e = [i, j]$ an edge in $E$,
$T_e$ the subgraph of $T$ induced by the nodes contained in the span of $e$.
***Degree of an edge*** $e \in E$, ***ed(e)***, is the number of connected components $c$ in $T_e$ such that the root of $c$ is not dominated by the head of $e$.
***Edge degree of T, ed(T)*** … max $\{ed(e) \mid e \in T\}$



# Edge Degree

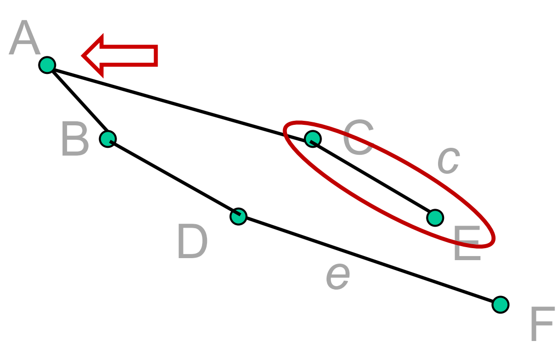
Let $T = (N, E)$ dependency tree, $e = [i, j]$ an edge in $E$,
$T_e$ the subgraph of $T$ induced by the nodes contained in the span of $e$.
***Degree of an edge*** $e \in E$, ***ed(e)***, is the number of connected components $c$ in $T_e$ such that the root of $c$ is not dominated by the head of $e$.
***Edge degree of T, ed(T)*** … max $\{ed(e) \mid e \in T\}$

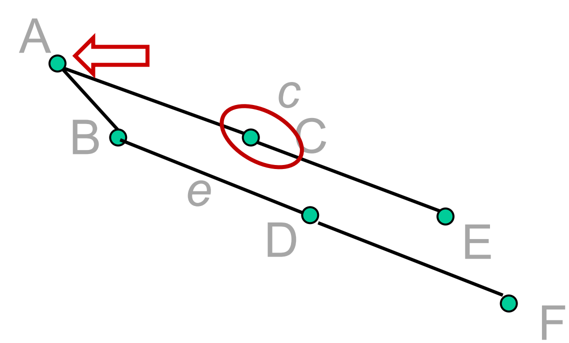
# Edge Degree

Let $T = (N, E)$ dependency tree, $e = [i, j]$ an edge in $E$, $T_e$ the subgraph of $T$ induced by the nodes contained in the span of $e$.

***Degree of an edge*** $e \in E$, ***ed(e)***, is the number of connected components $c$ in $T_e$ such that the root of $c$ is not dominated by the head of $e$.

***Edge degree of T, ed(T)*** … max $\{ed(e) \mid e \in T\}$



# Planarity vs. projectivity

projectivity $\Rightarrow$ planarity $\Rightarrow$ well-nestedness

projectivity $\not\Leftarrow$ planarity $\not\Leftarrow$ well-nestedness

$gd(T) = 0 \Leftrightarrow ed(T) = 0 \Leftrightarrow$ projectivity

well-nestedness … independent from gap/edge degree

$\forall\ d > 0$ well-nested and non-well-nested trees exist such that $gd(T) = d$ and $ed(T) = d$

(Kuhlmann, M., Nivre, J., 2006)

| property | DDT | | PDT | |
|---|---|---|---|---|
| *all structures* | $n = 4393$ | | $n = 73088$ | |
| gap degree 0 | 3732 | 84.95% | 56168 | 76.85% |
| gap degree 1 | 654 | 14.89% | 16608 | 22.72% |
| gap degree 2 | 7 | 0.16% | 307 | 0.42% |
| gap degree 3 | – | – | 4 | 0.01% |
| gap degree 4 | – | – | 1 | < 0.01% |
| edge degree 0 | 3732 | 84.95% | 56168 | 76.85% |
| edge degree 1 | 584 | 13.29% | 16585 | 22.69% |
| edge degree 2 | 58 | 1.32% | 259 | 0.35% |
| edge degree 3 | 17 | 0.39% | 63 | 0.09% |
| edge degree 4 | 2 | 0.05% | 10 | 0.01% |
| edge degree 5 | – | – | 2 | < 0.01% |
| edge degree 6 | – | – | 1 | < 0.01% |
| projective | 3732 | 84.95% | 56168 | 76.85% |
| planar | 3796 | 86.41% | 60048 | 82.16% |
| well-nested | 4388 | 99.89% | 73010 | 99.89% |
| *non-projective structures only* | $n = 661$ | | $n = 16920$ | |
| planar | 64 | 9.68% | 3880 | 22.93% |
| well-nested | 656 | 99.24% | 16842 | 99.54% |

Kuhlmann, M., Nivre, J. (2006)

# Corpora with dependency trees

- PropBank (1995)
  http://propbank.github.io/
- family of Prague dependency treebanks: Czech, Arabic, English, …
  http://ufal.mff.cuni.cz/pdt.html
- HamleDT project (from 2012) http://ufal.mff.cuni.cz/hamledt
- Universal Dependencies (from 2013) http://universaldependencies.org/

- Danish Dep. Treebank
  http://mbkromann.github.io/copenhagen-dependency-treebank/
- Finnish: Turku Dependency Treebank
  http://bionlp.utu.fi/fintreebank.html
- Negra corpus
  http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html
- TIGERCorpus
  http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html/
- SynTagRus Dependency Treebank for Russian

# References


- Partee, B. H.; ter Meulen, A.; Wall, R. E. (1990) *Mathematical Methods in Linguistics*. Kluwer Academic Publishers
- Kuhlmann, M., Nivre, J. (2006) Mildly Non-Projective Dependency Structures. In COLING/ACL Main Conference Poster Sessions, 507–514.
- Havelka, J. (2007) Mathematical Properties of Dependency Trees and their Application to Natural Language Syntax. PhD Thesis, MFF UK
- Holan, T., Kuboň, V., Oliva, K., Plátek, M. (2000) On Complexity of Word Order. *Les grammaires de dépendance – Traitement automatique des langues,* vol. 41, no. 1, 273-300
- Petkevič, V. (1995) A New Formal Specification of Underlying Structure. *Theoretical Linguistics*, vol. 21, No.1

# Projectivity and non-projectivity (definition)

NENÍ ekvivalentní podmínka, viz Jura PhD, str. 30 !!!

A subtree *S* of a rooted dependency tree *T* is ***projective*** iff for all nodes *a*, *b* and *c* of the subtree *S* the condition holds:

(1) $(a \leq_D b)\ \&\ (a <_{WO} b)\ \&\ (b \leq_D^{*} c) \Rightarrow (a <_{WO} c)$

<u>and</u>

(2) $(a \leq_D b)\ \&\ (b <_{WO} a)\ \&\ (b \leq_D^{*} c) \Rightarrow (c <_{WO} a)$