Prague Dependency Treebank: Introduction

Markéta Lopatková, Jiří Mírovský

Institute of Formal and Applied Linguistics, MFF UK

lopatkova@ufal.mff.cuni.cz

NPFL075 Prague Dependency Treebank

Lectures:

Markéta Lopatková ~~Thu, S1, 15:40-17:10 (cz)~~
Fri, S7, 10:40-12:10 (eng)

Practical sessions:

Jiří Mírovský Fri, SU2, 9:00-10:30

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

Requirements:

- Homework (45%)
- Activity (15%)
- Final test (40%)

Assessment:

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

Prague Dependency Treebank

Collection of:

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

Prague Dependency Treebank

Collection of:

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

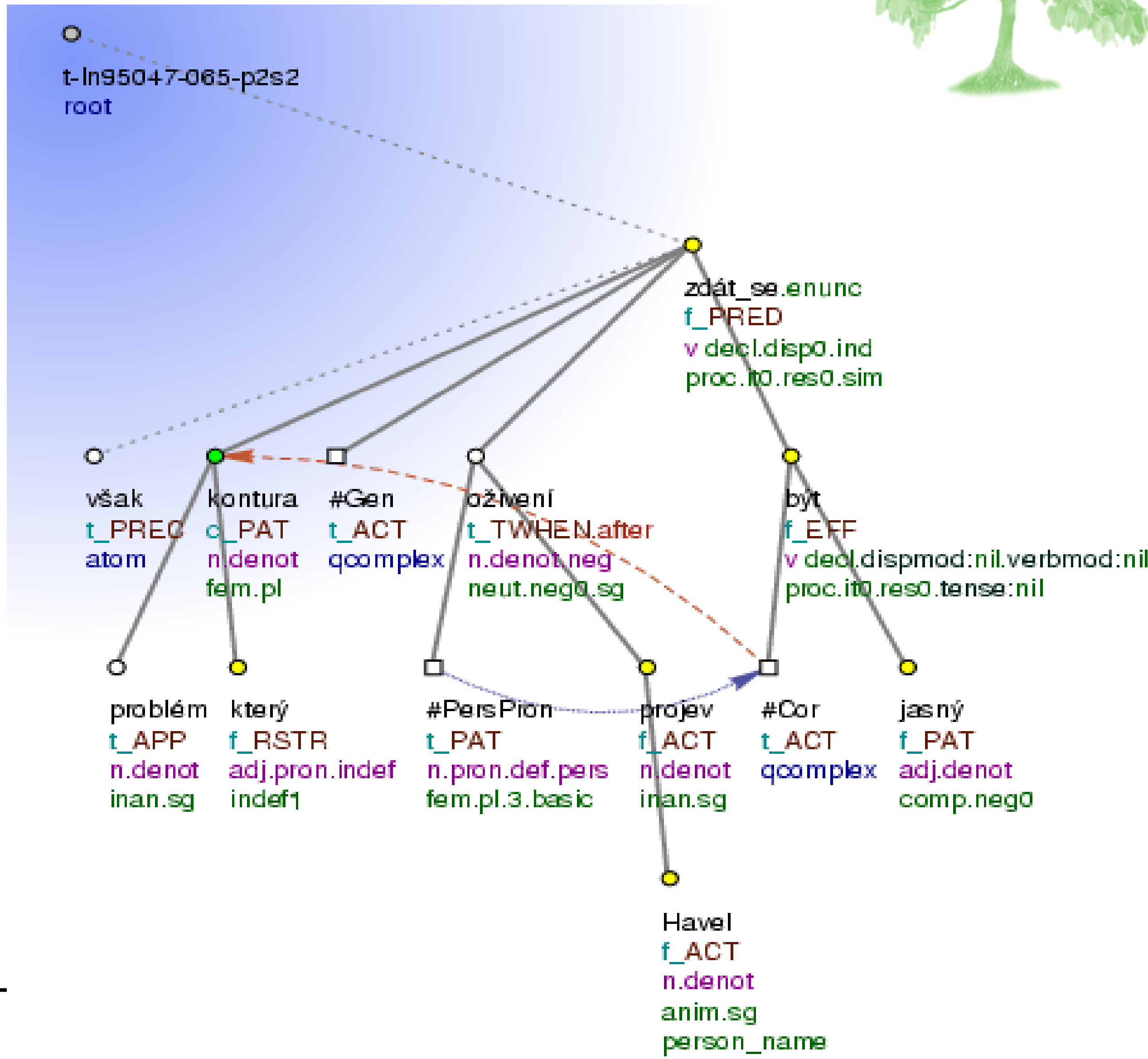
Another point of view:

- annotation scheme
- framework for annotation of different languages
- underlying linguistic theory (Functional Generative Description)

Outline of the lecture

- trees (graph theory and data format)
- phrase structure trees and dependency trees
- dependency and non-dependency relations
- non-projectivity

How to capture sentence structure?



Graph theory: tree

tree (graph theory):

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

⇒ (undirected) graph

any two nodes are connected by exactly one simple path

Graph theory: tree

tree (graph theory):

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

⇒ (undirected) graph

any two nodes are connected by exactly one simple path

rooted tree

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

Graph theory: tree

tree (graph theory):

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

⇒ (undirected) graph

any two nodes are connected by exactly one simple path

rooted tree

- rooted ⇒ 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 (or away from) a particular node ~ the ***root***

Data structure: tree

tree as a data structure:

- finite directed graph $\langle N, E \rangle$, N~nodes, E~edges $[n_1, n_2]$
  - no cycles
  - connected
  - with root
- each non-root node has exactly one parent, and the root has no parent

  (each node has zero or more children nodes)

Data structure: tree

tree as a data structure:

- finite directed graph ⟨N, E⟩, N~nodes, E~edges [n_1, n_2]
  - no cycles
  - connected
  - with root
- each non-root node has exactly one parent, and the root has no parent
  (each node has zero or more children nodes)

+

- (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)

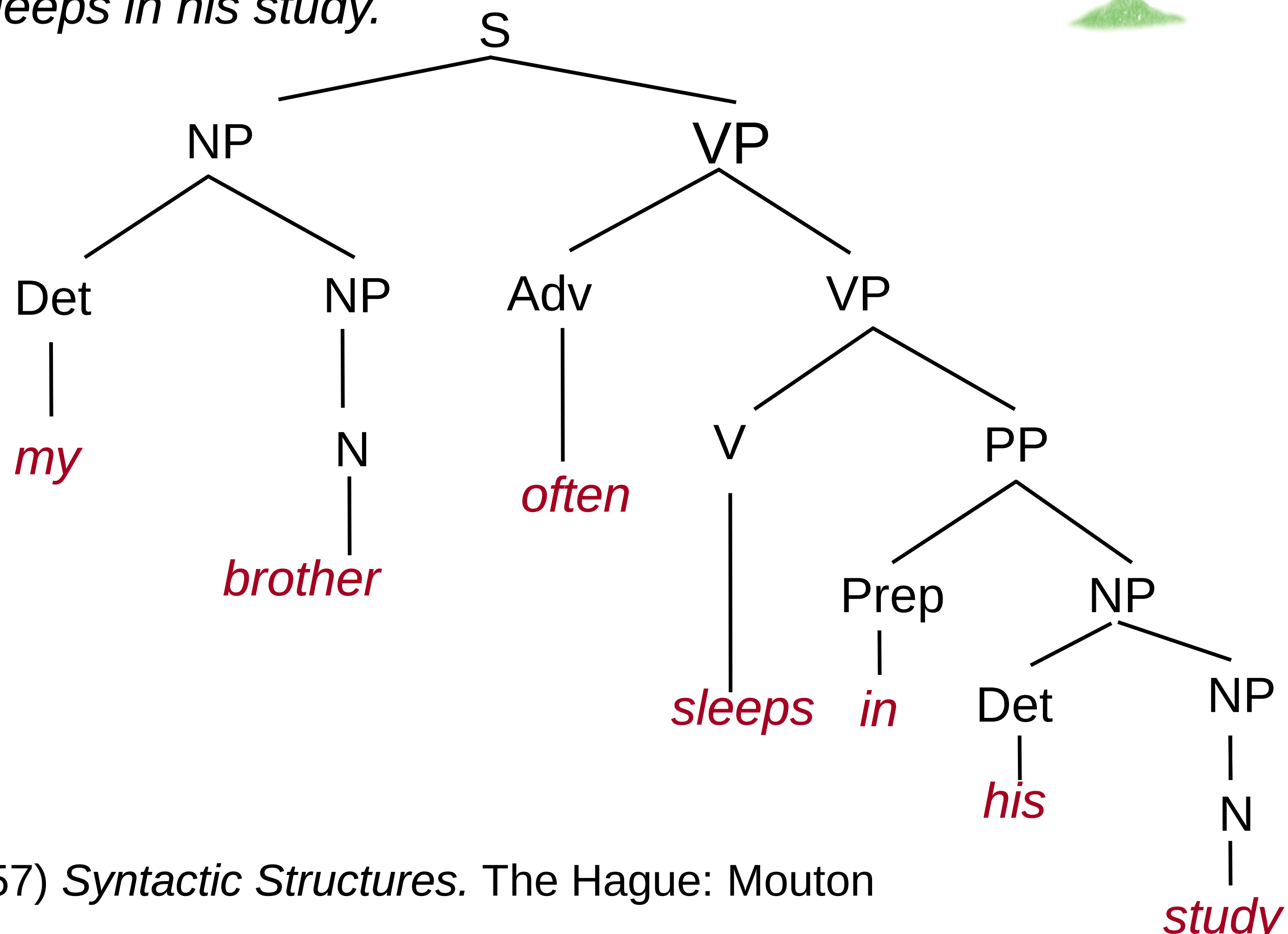
Tree-based structures in CL

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)

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

$\langle N, D \rangle$ … ***tree*** (as a data structure)

Q ... lexical and grammatical categories

L … labeling function $N \rightarrow Q$

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

dominance relation

+

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

relation of *precedence*

Phrase structure tree (definition)

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

$\langle N, D \rangle$ … ***tree*** (as a data structure)

Q ... lexical and grammatical categories

L … labeling function $N \rightarrow Q$

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

dominance relation

+

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

relation of *precedence*

+

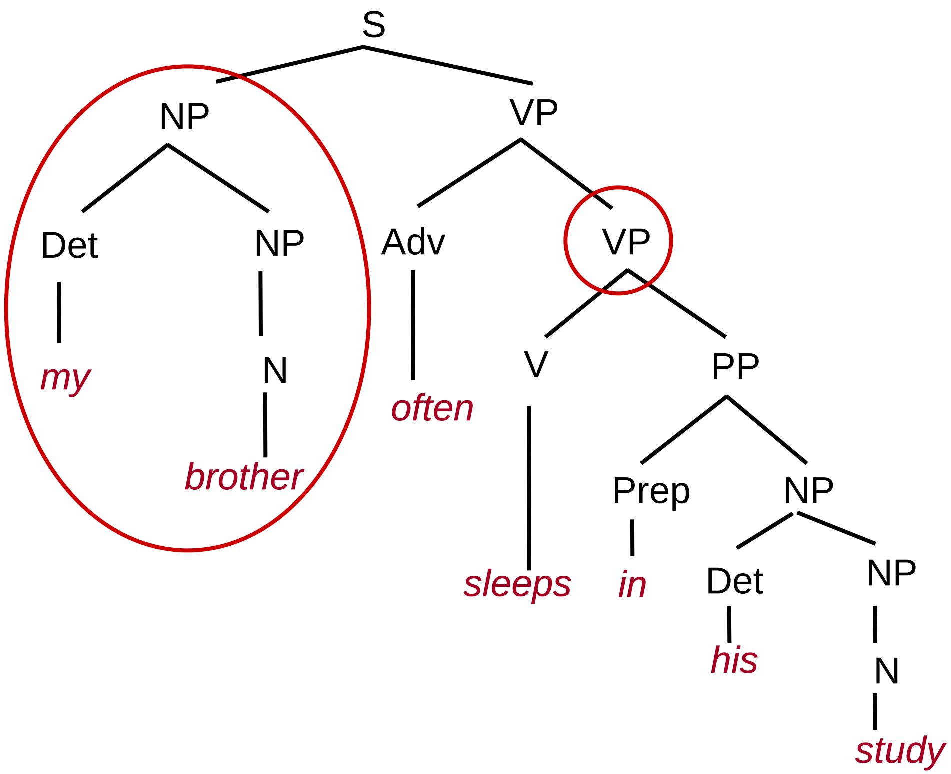
Relating dominance and precedence relations:

- *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)

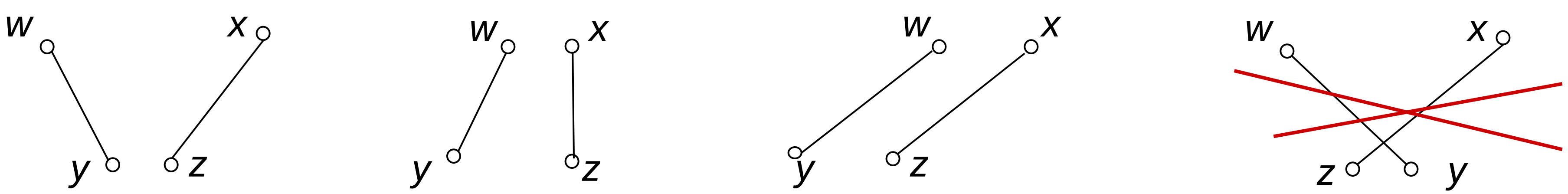
- *exclusivity* condition for D and P relations

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

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

⇨

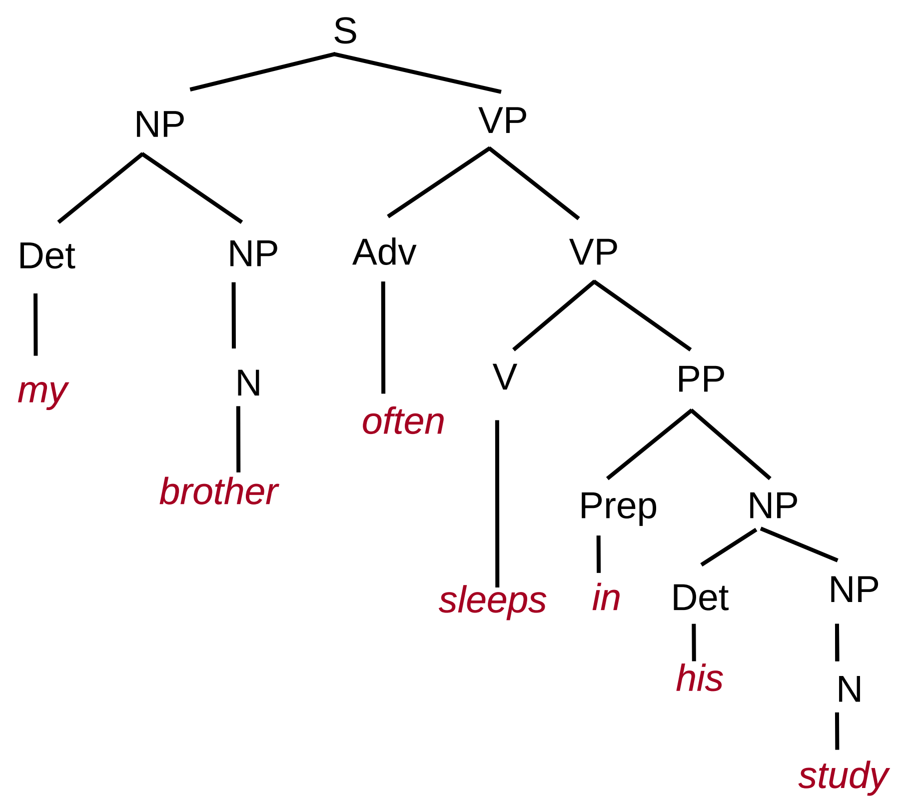
T = $\langle$ N,D,Q, P,L $\rangle$ phrase structure tree

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

Phrase structure tree

Pros

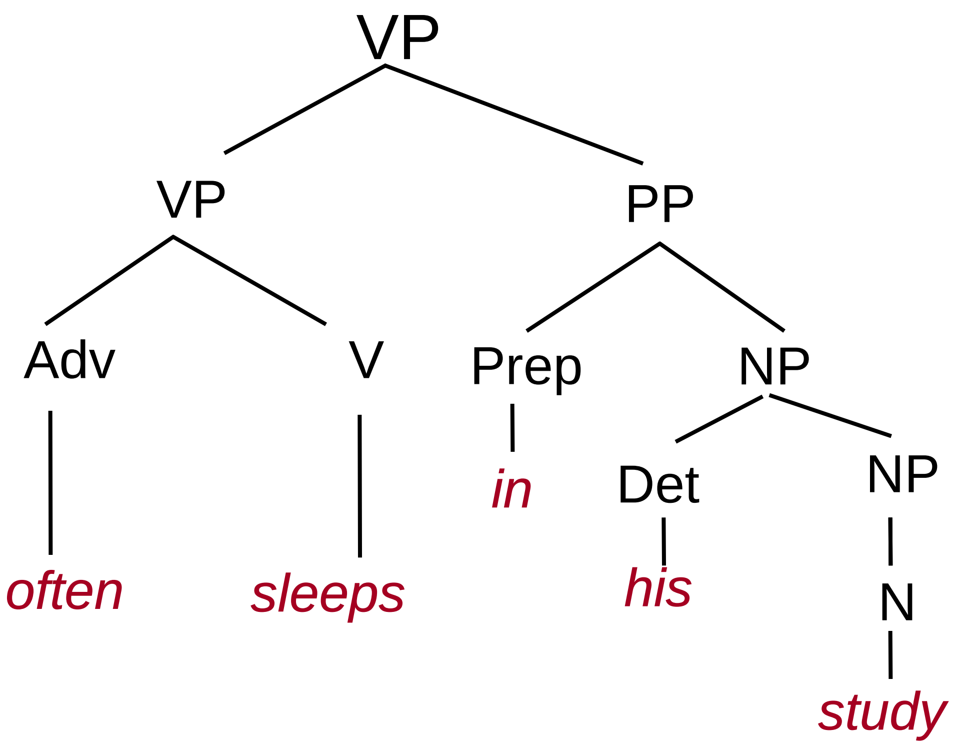
- derivation history / ‘closeness’ of a complementation
- coordination, apposition
- CFG-like
- derivation of a grammar



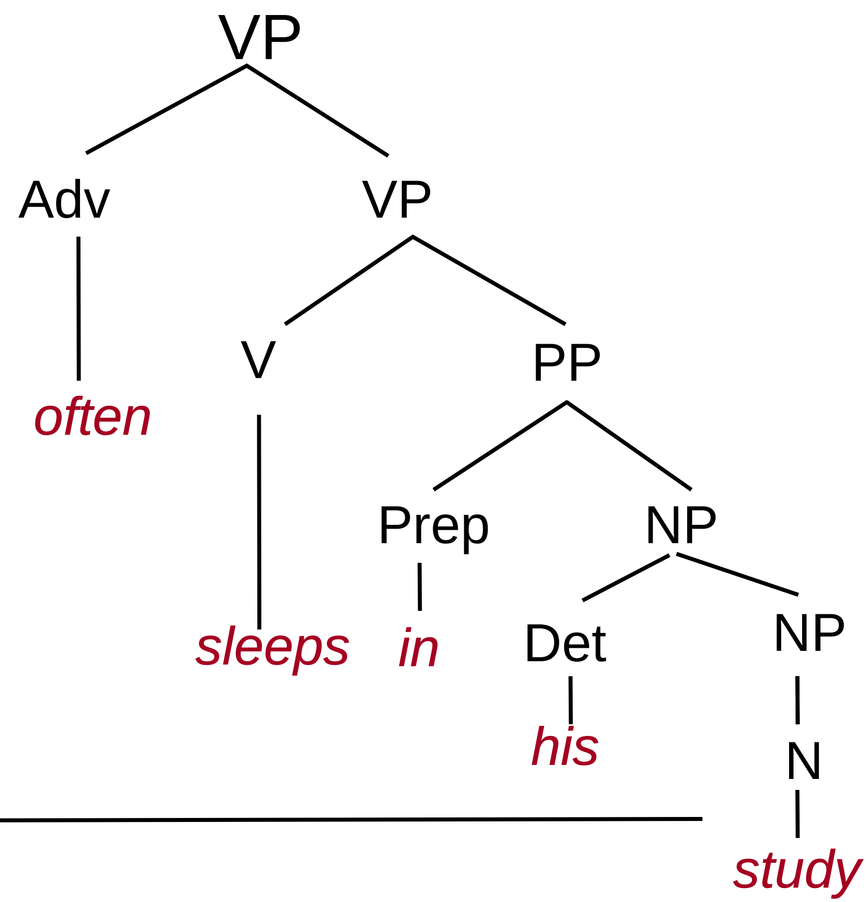
Phrase structure tree

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

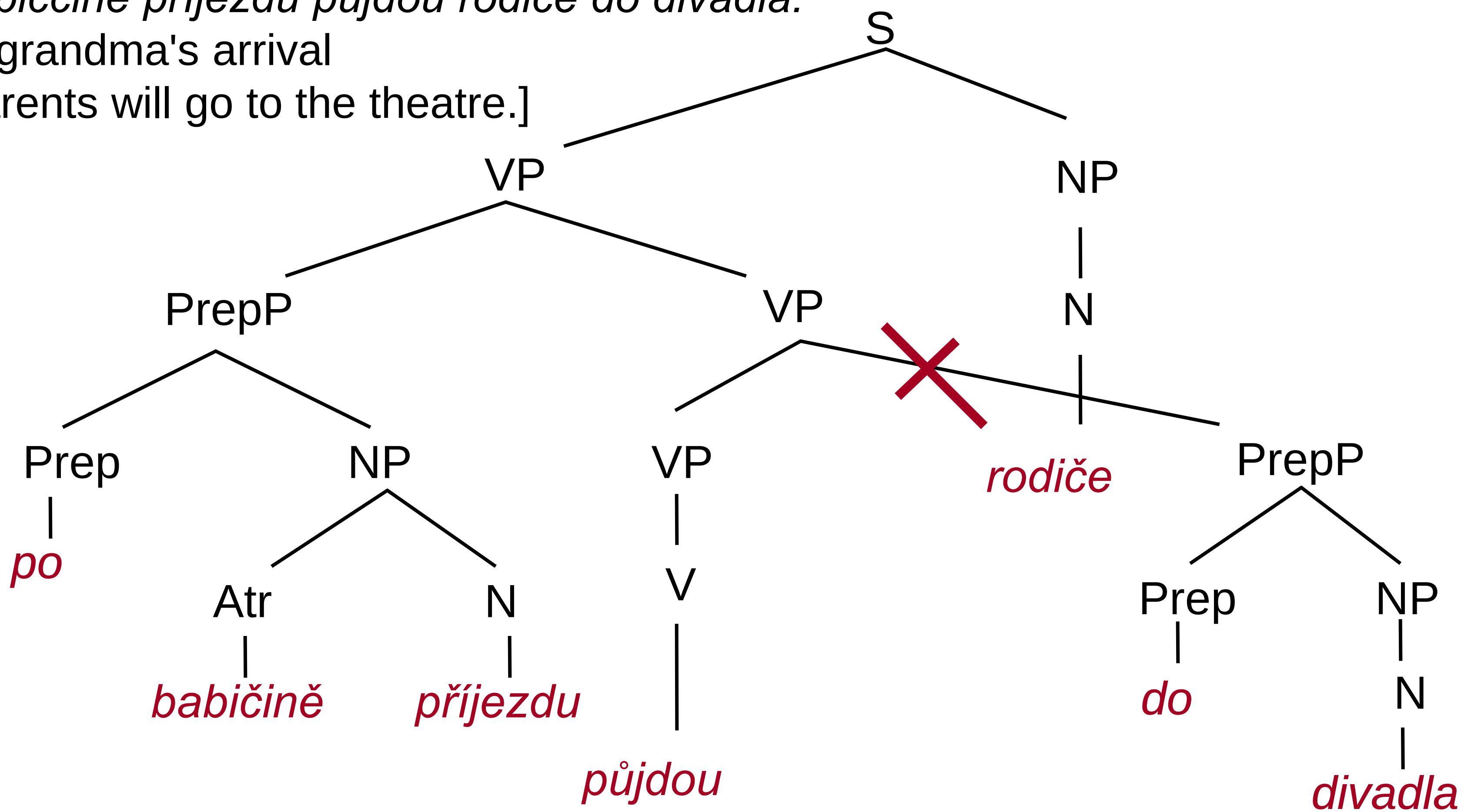
Contras

- 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’:

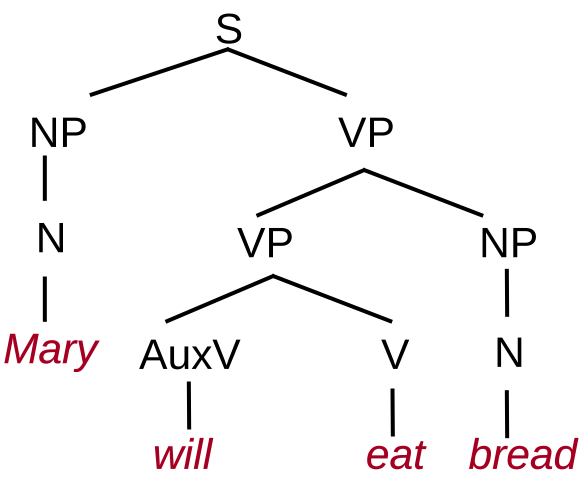
Po babiččině příjezdu půjdou rodiče do divadla.
[After grandma's arrival
the parents will go to the theatre.]



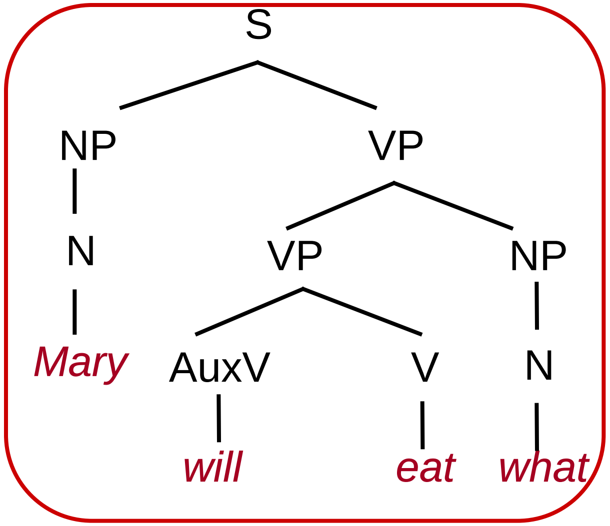
Phrase structure tree

discontinuous 'phrases': solution for English

Mary will eat bread.



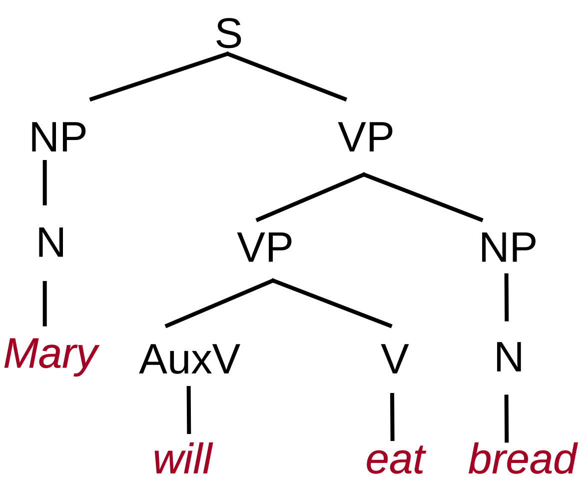
What will Mary eat?



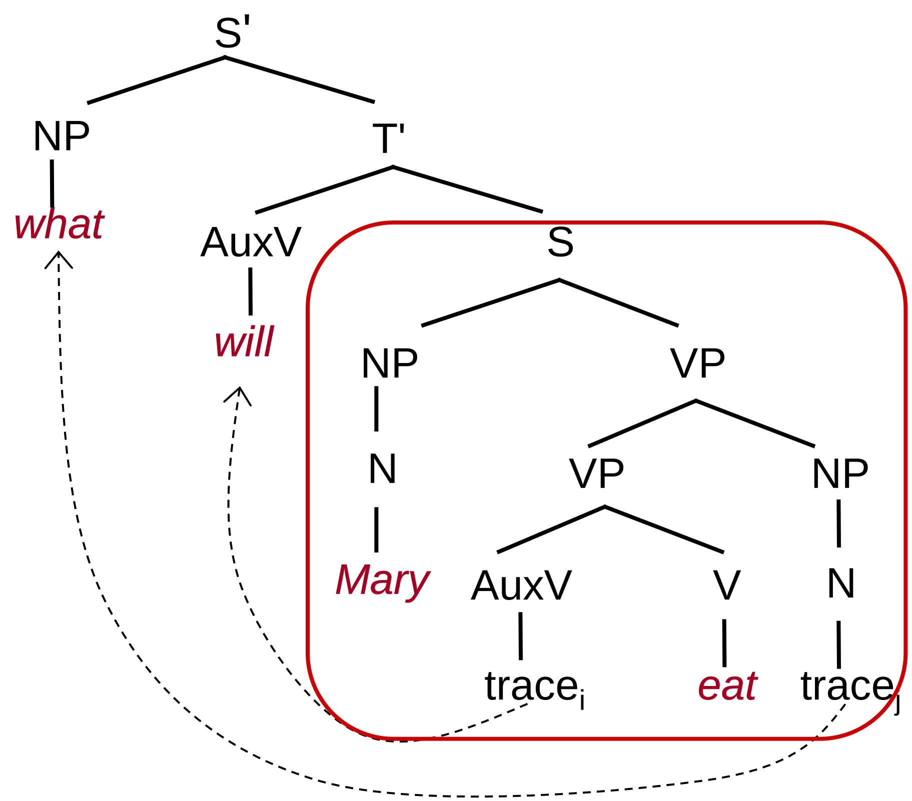
Phrase structure tree

discontinuous ‘phrases’: solution for English

Mary will eat bread.



What will Mary eat?



Corpora with phrase structure trees

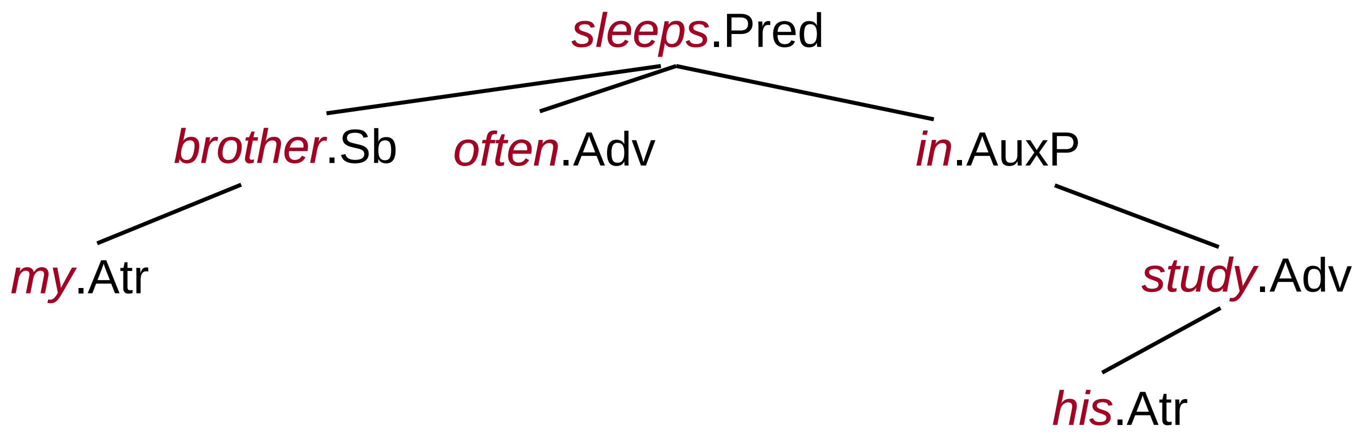
- Penn Treebank (1995)
  Mitchel Marcus (1993) Computational Linguistics, vol. 19
  http://www.cis.upenn.edu/~treebank/

  Penn Arabic Treebank, Penn Chinese Treebank
- International English Treebank (ICE)
  http://ice-corpora.net/ice/index.htm
- Paris 7
  http://www.llf.cnrs.fr/Gens/Abeille/French-Treebank-fr.php
- Szeged Treebank 2.0
  http://www.inf.u-szeged.hu/projectdirs/hlt/en/Szeged%20Treebank%202.0_en.html
- many others

Dependency tree

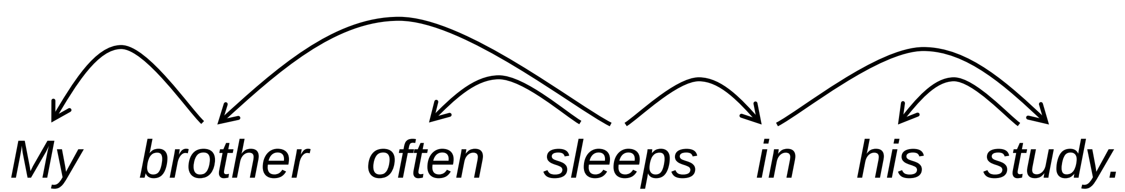
Dependency tree

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⟩ … ***tree*** (as a data structure)

Q ... lexical and grammatical categories

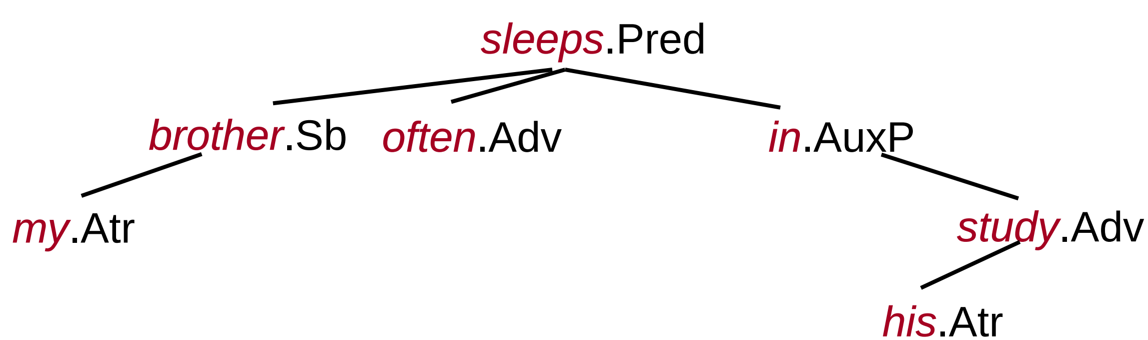
L … labeling function N → Q

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

‘dependency’ relation

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

word order



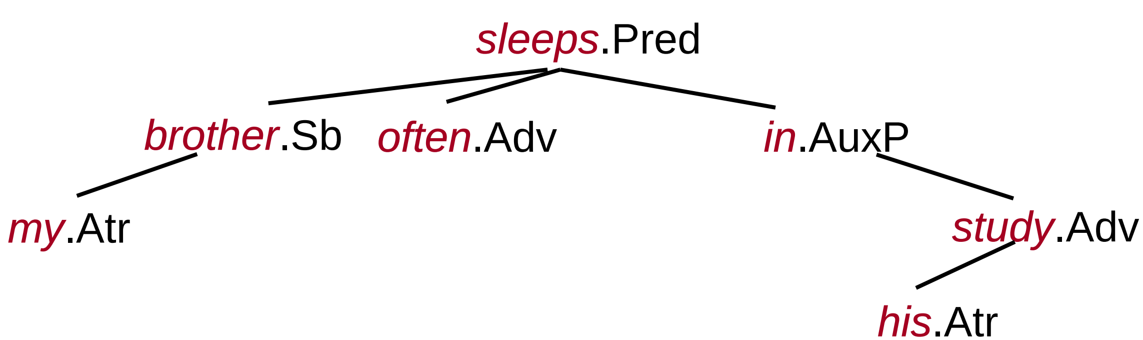
Dependency tree

Pros

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

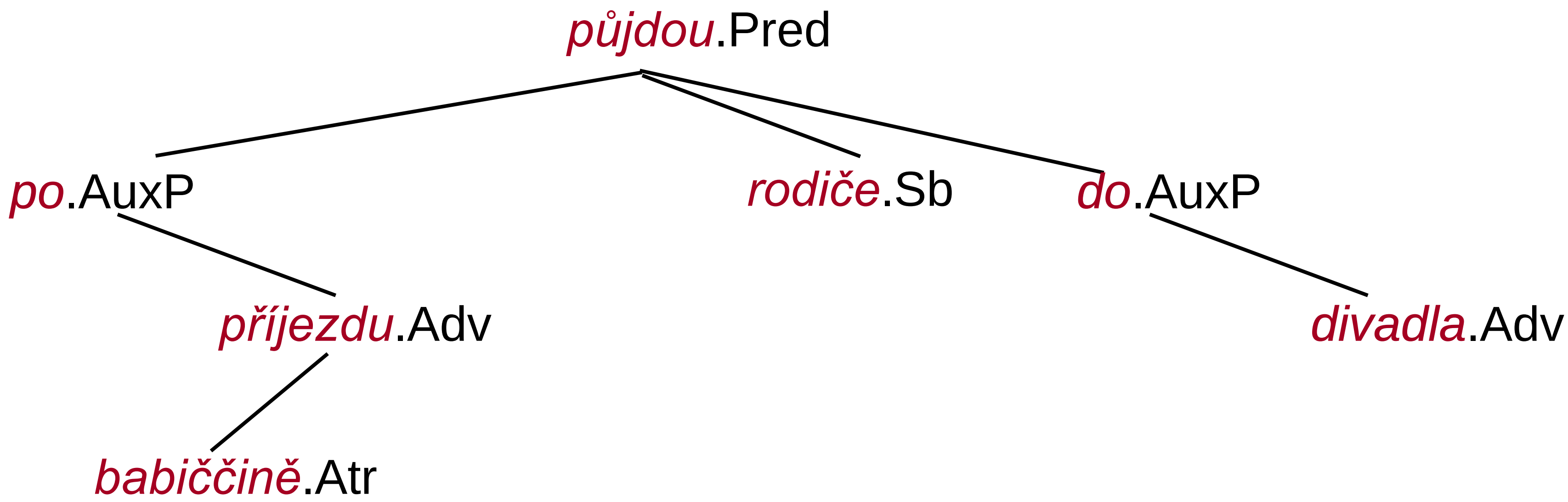
Contras

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



Dependency tree

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



Corpora with dependency trees

- PropBank (1995)
- family of Prague dependency treebanks: Czech, Arabic, English
  http://ufal.mff.cuni.cz/pdt.html
- Danish Dep. Treebank
  http://code.google.com/p/copenhagen-dependency-treebank/wiki/CDT
- 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/projekte/TIGER/
- SynTagRus Dependency Treebank for Russian

Dependency and non-dependency relations

Dependency and non-dependency relations

edges ~ *dependency relations* (prototypically)

- dependency relation: binary relation
- governing/modified unit (head) – dependent/modifying unit (modifier)
- criterion: possible reduction

  … dependent member of the pair may be deleted

  while the distributional properties are preserved (→ correctness is preserved)

Dependency and non-dependency relations

edges ~ ***dependency relations*** (prototypically)

- dependency relation: binary relation
- governing/modified unit (head) – dependent/modifying unit (modifier)
- criterion: possible reduction

  … dependent member of the pair may be deleted

  while the distributional properties are preserved (→ correctness is preserved)

  - endocentric constructions … OK

    malý stůl [small table], *přišel včas* [(he) came in time], *velmi brzo* [very soon]

  - exocentric constructions … ***principle of analogy*** on word classes

    Prší. [(It) rains.] … ∃ subjectless verbs
    ⇒ *Král zemřel.* [The king died.] … a verb rather than a noun is the head

    The girl painted a bag. → *The girl painted. ...* ∃ objectless verbs
    ⇒ *The girl carried a bag* … an object is considered as depending on a verb

Dependency and non-dependency relations

BUT also other relations:

coordination … "multiplication" of a single syntactic position

- different referents
- coordination of sentence members / sentences

  My sister Mary and John came late.

  Mary came in time but John was late.

  Nemohu odejít, neboť ještě nepřestalo pršet.

  [I can't leave since it hasn't stopped raining yet.]
- coordination may be embedded

  krásné a romantické hrady a zámky [nice and romantic towers and castles]

Dependency and non-dependency relations

BUT also other relations:

coordination … "multiplication" of a single syntactic position

- different referents
- coordination of sentence members / sentences

  My sister Mary and John came late.
  Mary came in time but John was late.
  Nemohu odejít, neboť ještě nepřestalo pršet.
  [I can't leave since it hasn't stopped raining yet.]
- coordination may be embedded

  krásné a romantické hrady a zámky [nice and romantic towers and castles]

apposition … "multiplication" of a single syntactic position

- identical referent

  Charles IV, Holy Roman Emperor
  The Hobbit, or There and Back Again

Dependency and non-dependency relations

BUT also other relations:

coordination … "multiplication" of a single syntactic position

- different referents
- coordination of sentence members / sentences

  My sister Mary and John came late.
  Mary came in time but John was late.
  Nemohu odejít, neboť ještě nepřestalo pršet.
  [I can't leave since it hasn't stopped raining yet.]
- coordination may be embedded

  krásné a romantické hrady a zámky [nice and romantic towers and castles]

apposition … "multiplication" of a single syntactic position

- identical referent

  Charles IV, Holy Roman Emperor
  The Hobbit, or There and Back Again

⇒ necessary to enrich the data structure

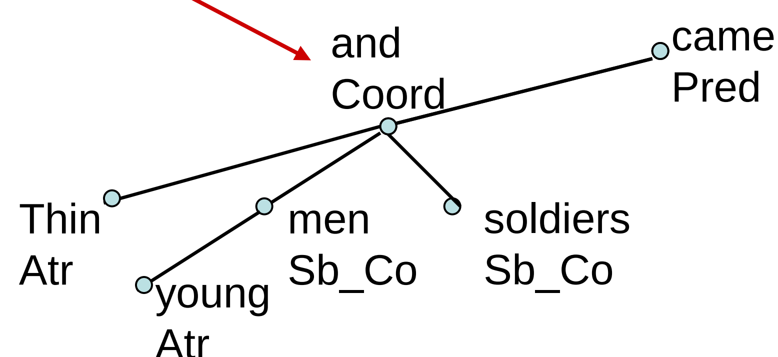
Coordination/apposition in dependency trees

PDT 2.0:

'connecting' constructions ~ coordination, apposition (, OPER)

specific types of nodes and edges:

- *connecting node* (= node for coordinating / appositing conjunction)



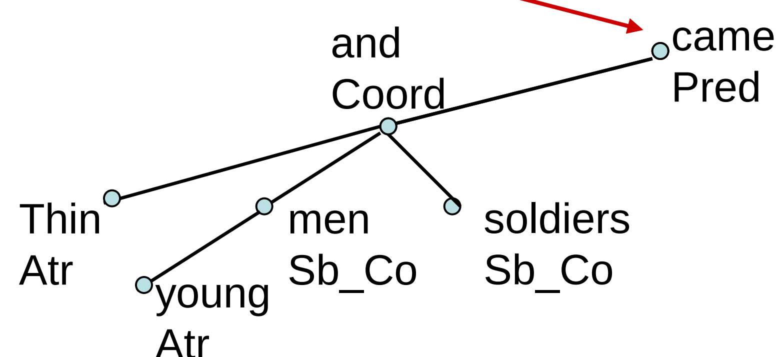
Coordination/apposition in dependency trees

PDT 2.0:

'connecting' constructions ~ coordination, apposition (, OPER)

specific types of nodes and edges:

- *connecting node* (= node for coordinating / appositing conjunction)
- *effective parent* (= node for governing node, i.e. node modified by the whole construction, 'linguistic parent')



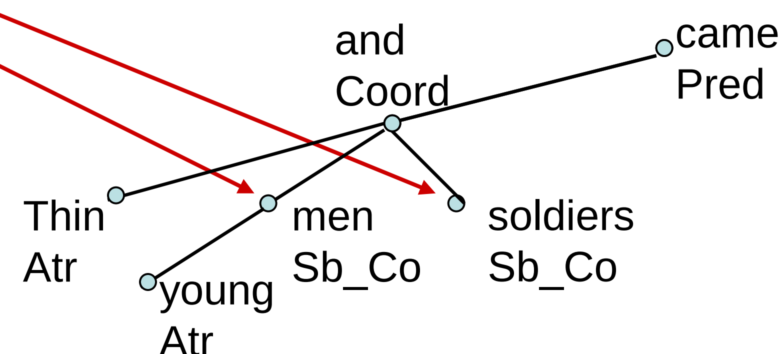
Coordination/apposition in dependency trees

PDT 2.0:

'connecting' constructions ~ coordination, apposition (, OPER)

specific types of nodes and edges:

- *connecting node* (= node for coordinating / appositing conjunction)
- *effective parent* (= node for governing node, i.e. node modified by the whole construction, 'linguistic parent')
- *members of a connecting construction* (= nodes that are coordinated / are in apposition)
  - `is_member`



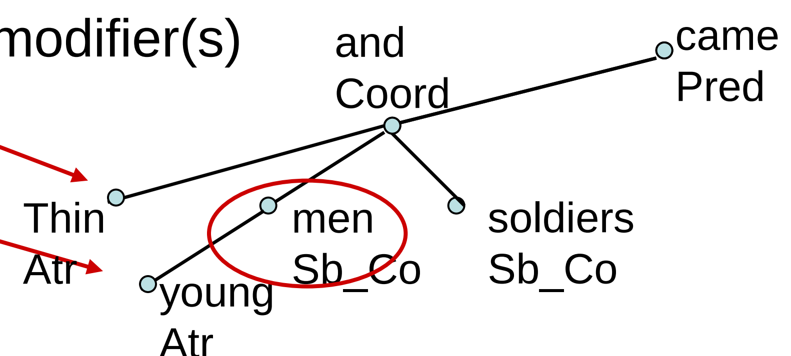
Coordination/apposition in dependency trees

PDT 2.0:

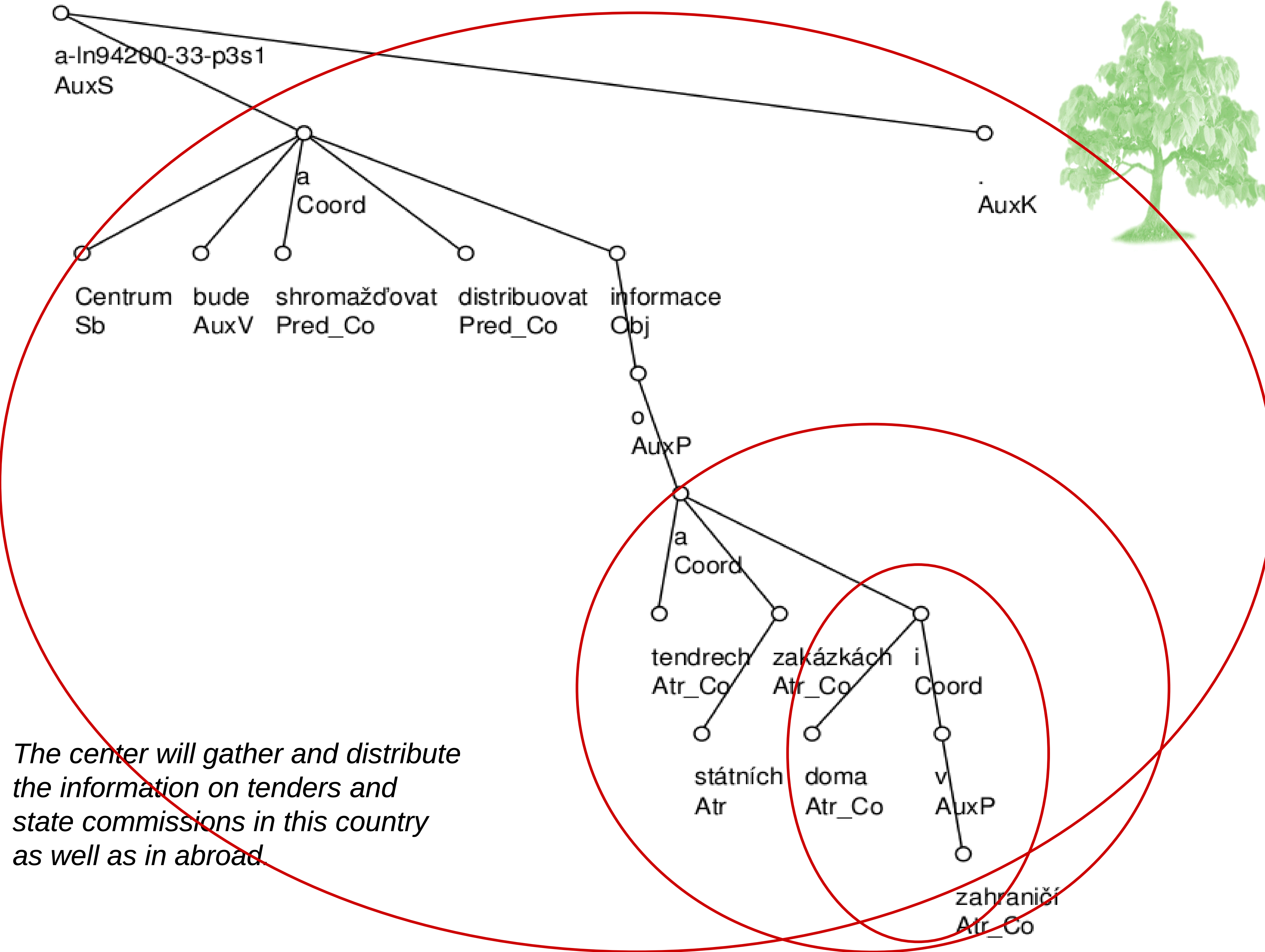
'connecting' constructions ~ coordination, apposition (, OPER)

specific types of nodes and edges:

- *connecting node* (= node for coordinating / appositing conjunction)
- *effective parent* (= node for governing node, i.e. node modified by the whole construction, 'linguistic parent')
- *members of a connecting construction* (= nodes that are coordinated / are in apposition)
  - `is_member`
- *effective child(ren)* … modification(s) of the individual member of the connecting construction **+** common/shared modifier(s)



- *'pass-through' nodes*



The center will gather and distribute the information on tenders and state commissions in this country as well as in abroad.

Coordination/apposition in dependency trees

PDT 2.0:

- embedded connecting constructions ⇒ recursivity

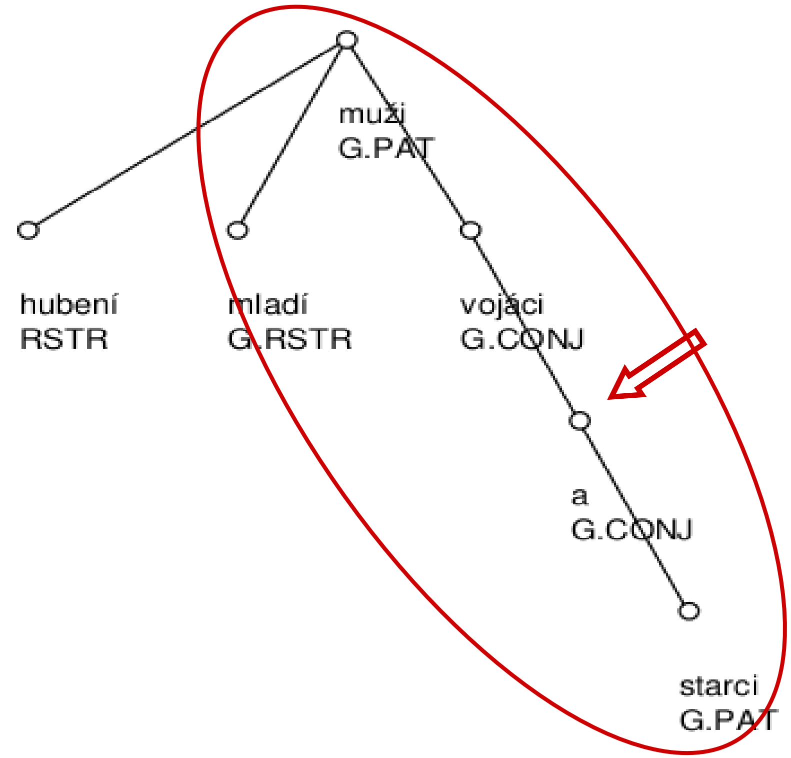
- *TrEd* (Tree Editor, Pajas):

functions `GetEChildren, GetEParents`

Coordination/apposition in dependency trees

Mel'čuk (1988):

‘grouping’ (G) … shared modification vs. modification of a single member



Hubení ((mladí muži) , vojáci a starci)
[Thin young men, soldiers and old-men]

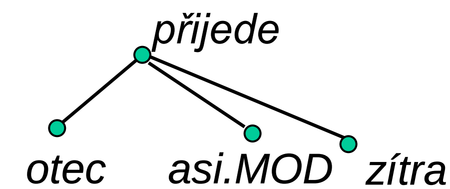
Dependency and non-dependency relations

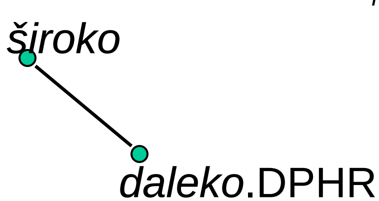
other non-dependency relations in PDT:

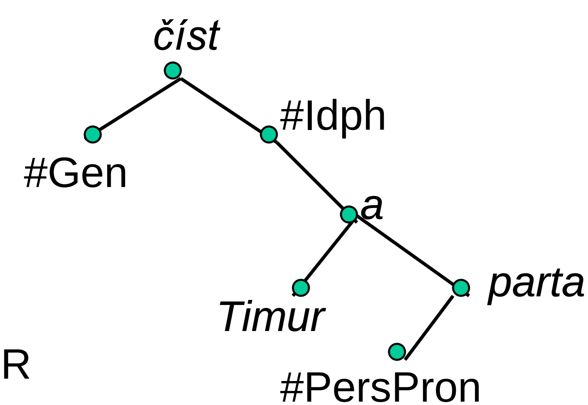
- technical root – effective root of a sentence
- syntactically unclear expressions

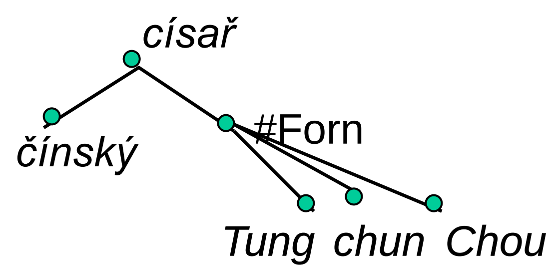
  rhematizers; sentence, linking and modal adverbial expressions, conjunction modifiers
- list structures

  names, foreign expressions
- phrasemes









Projectivity and non-projectivity

Projectivity and non-projectivity (definition)

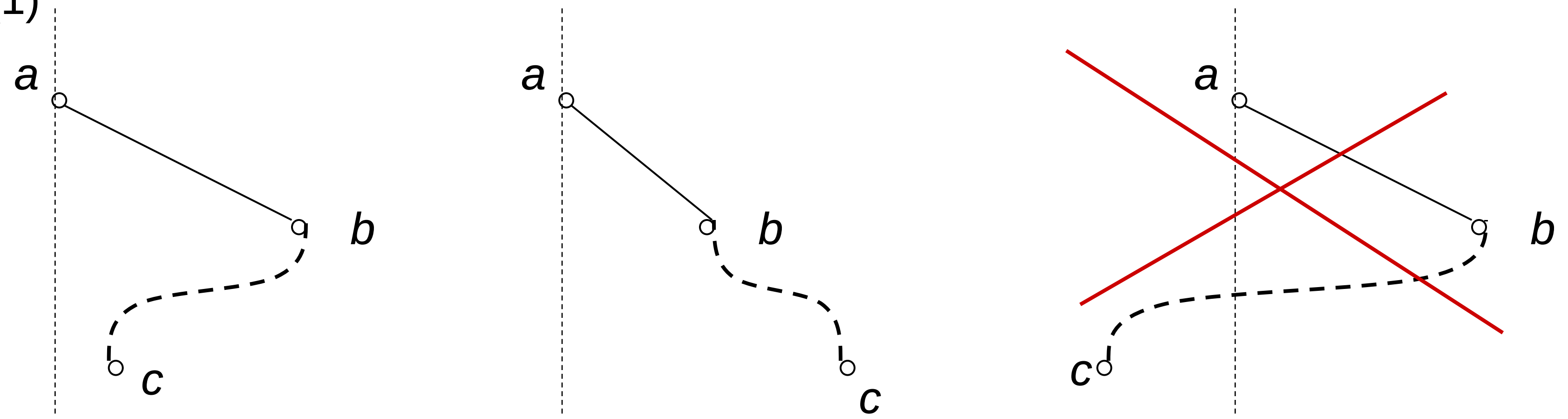
A subtree *S* of a rooted dependency tree *T* is ***projective*** iff for all nodes *a*, *b* and *c* of the subtree *S* the condition holds:

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

<u>and</u>

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

(1)



Projectivity and free word order

free word order:

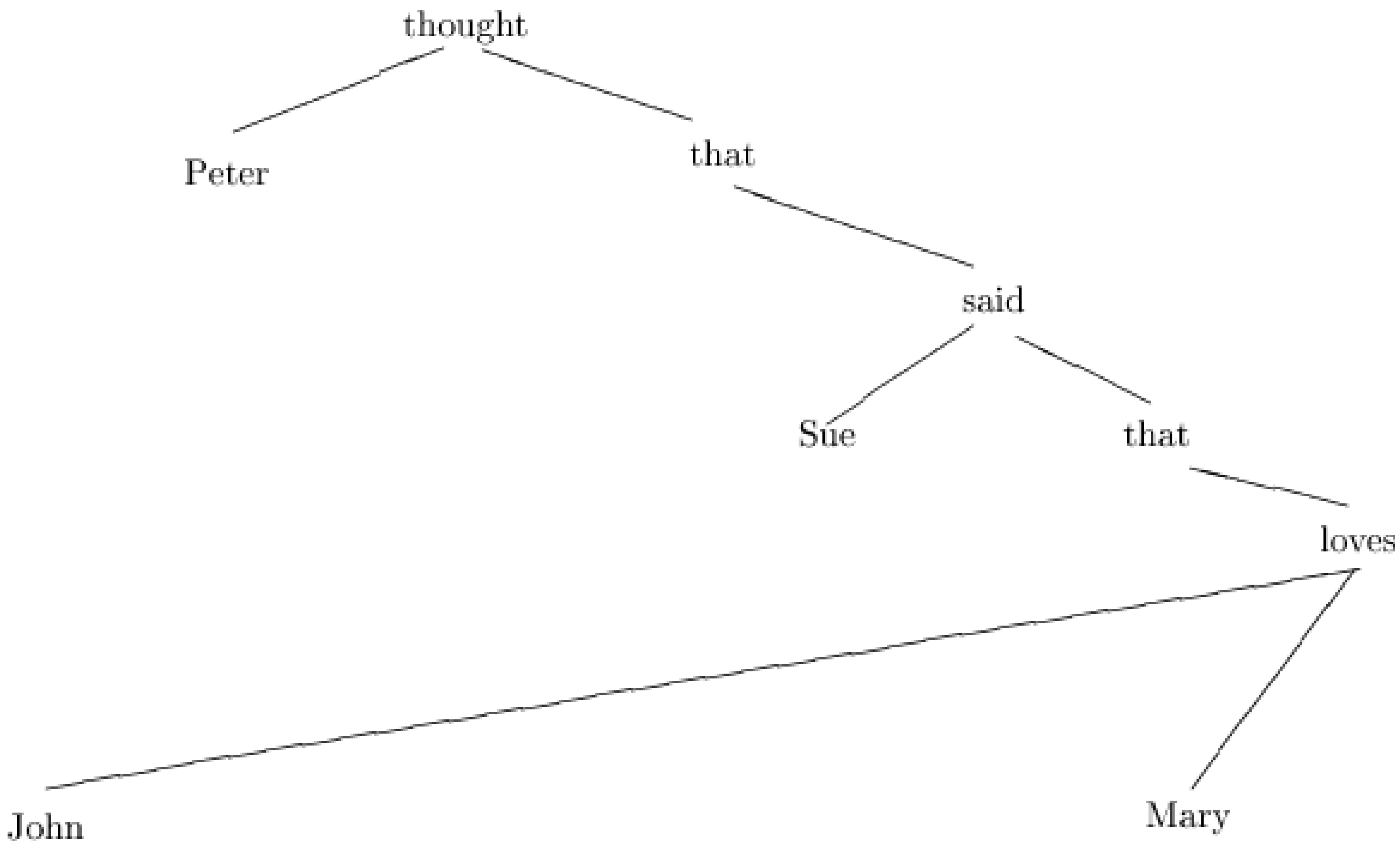
- freedom of word order of dependents within a continuous ‘head domain’ (i.e., substring of head + its dependents)
- relaxation of continuity of a head domain

German:
Maria hat einen Mann kennengelernt der Schmetterlinge sammelt.
Mary has a man met who butteries collects
Mary has met a man who collects butteries

Projectivity and free word order

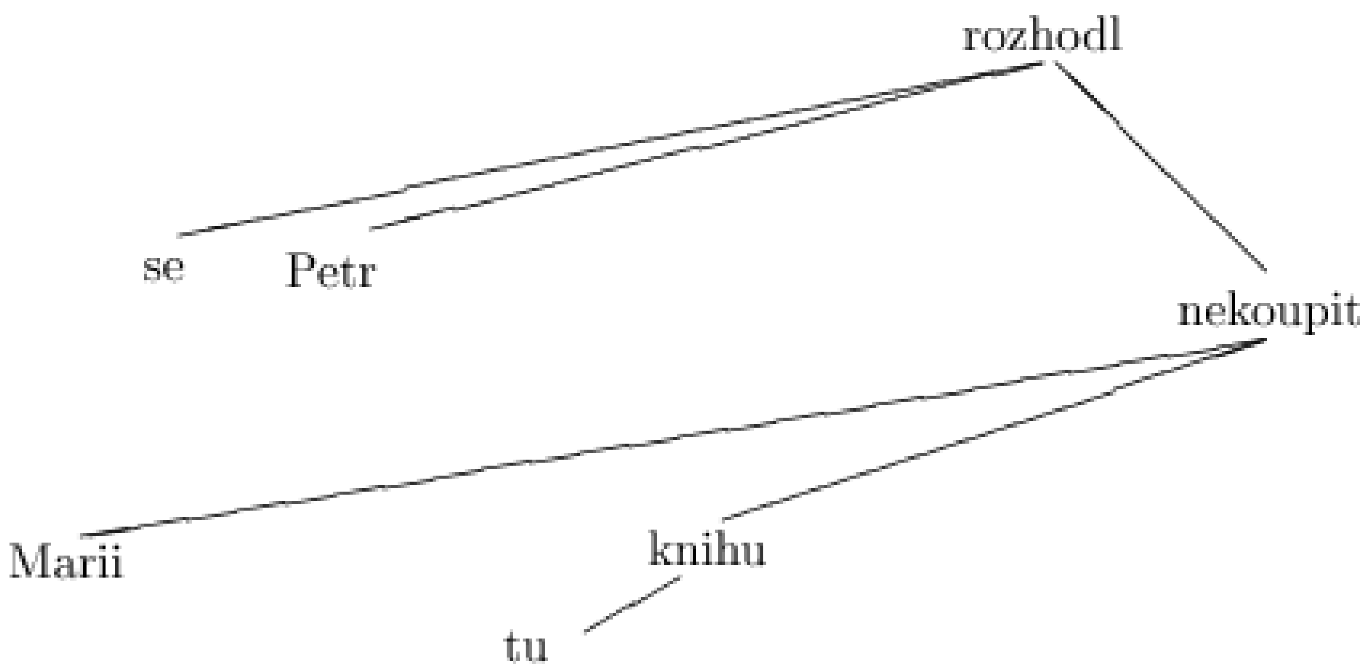
English: long-distance unbounded dependency

John, Peter thought that Sue said that Mary loves.



Projectivity and free word order

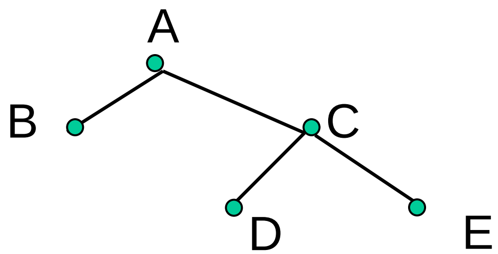
Czech:
Marii se Petr tu knihu rozhodl nekoupit.
to-Mary PART Peter that book decided not-buy
[Peter decided not to buy that book to Mary.]



Projectivity and non-projectivity

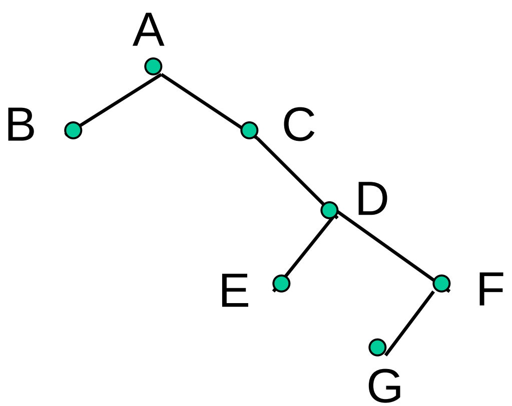
Projective dependency trees can be encoded by *linearization*:

- string of nodes, edges ~ brackets



A (B C (D)) without WO ordering

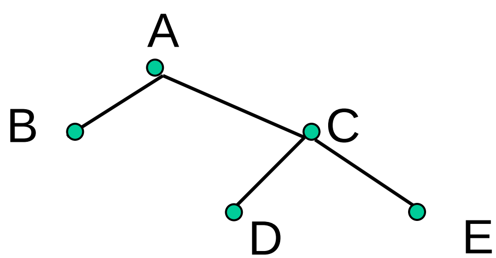
(B) A ((D) C (E)) with WO



Projectivity and non-projectivity

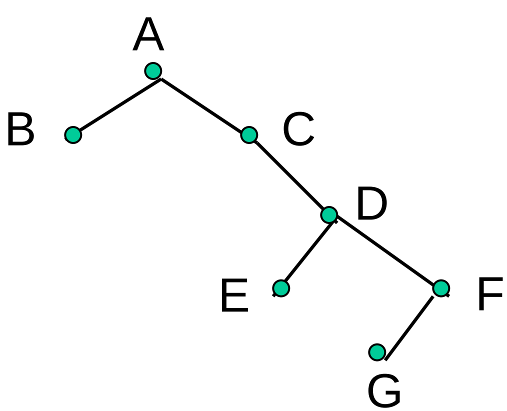
Projective dependency trees can be encoded by *linearization*:

- string of nodes, edges ~ brackets



A (B C (D))   without WO ordering

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



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

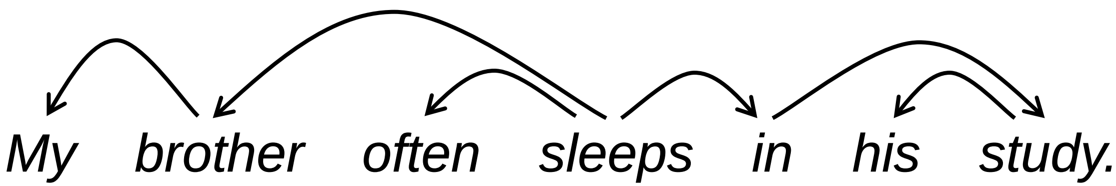
(B) A (C ((E) D ((G) F)))   with WO

Planarity

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

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

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

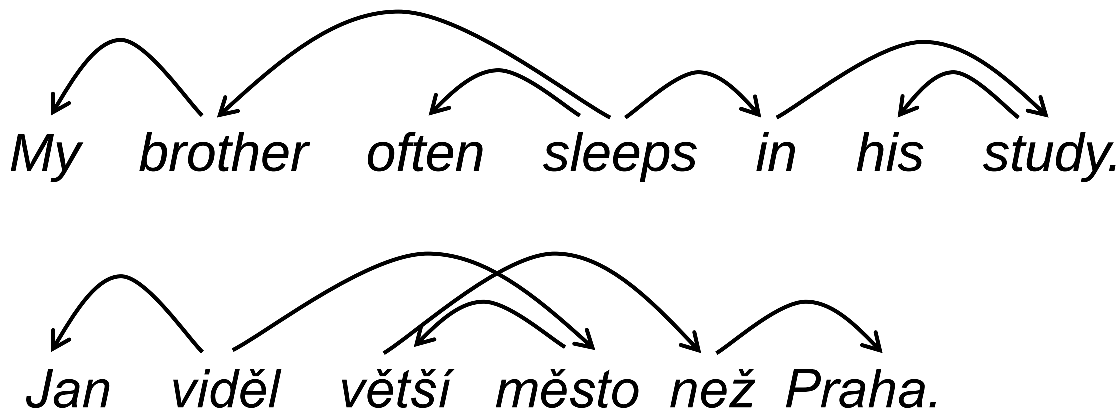


Planarity

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'

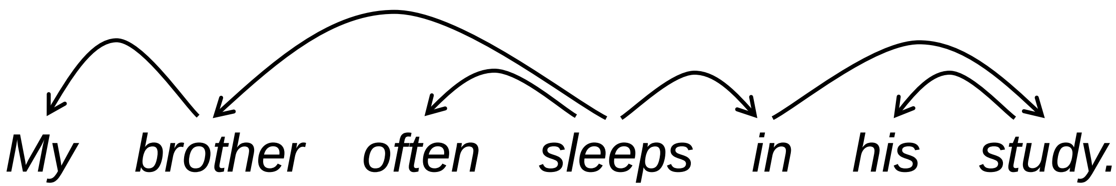


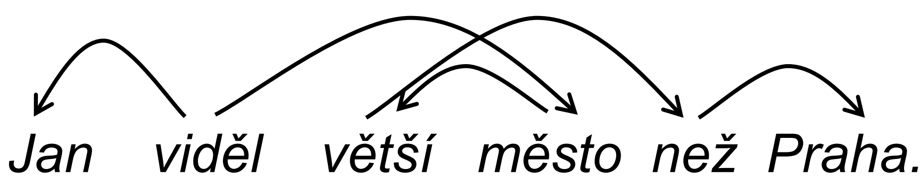
Planarity

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'





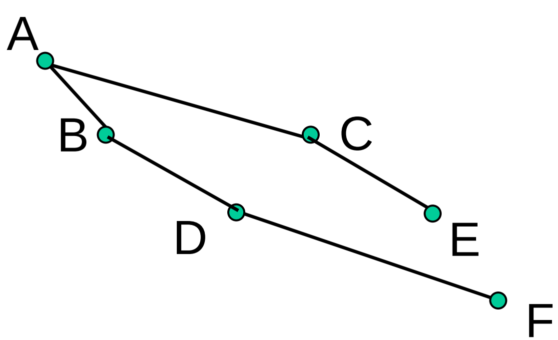
Informally, a dependency graph is planar, if its edges can be drawn above the sentence without crossing.

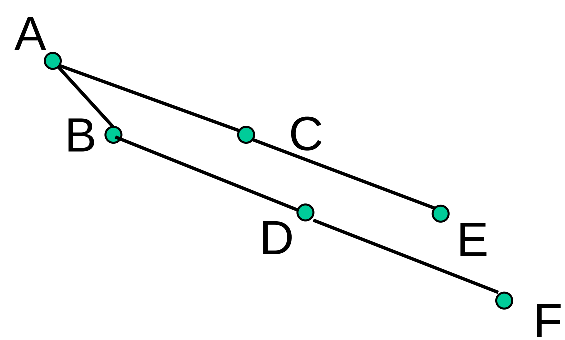
‘Well-Nestedness’

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.’



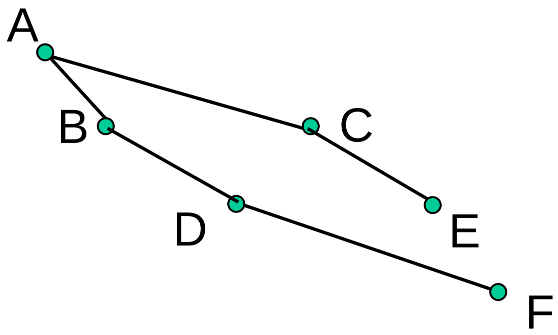


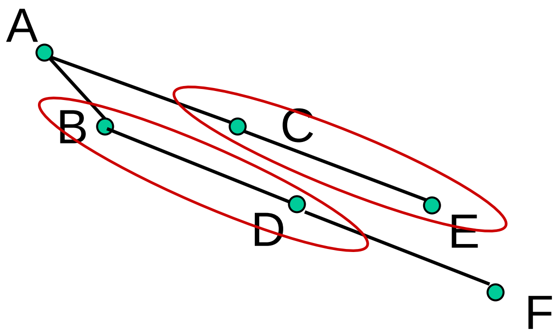
‘Well-Nestedness’

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.’



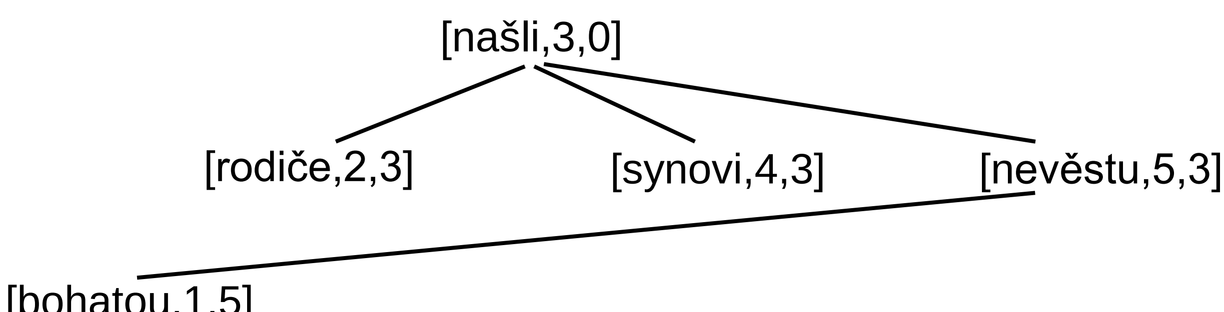


Gap Degree *dNh(T)*

Coverage of a node $u \in T$

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

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



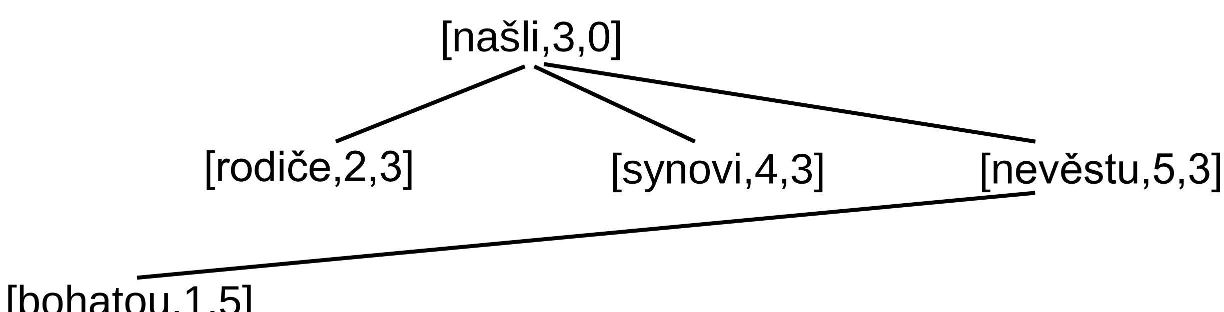
Gap Degree *dNh(T)*

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

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

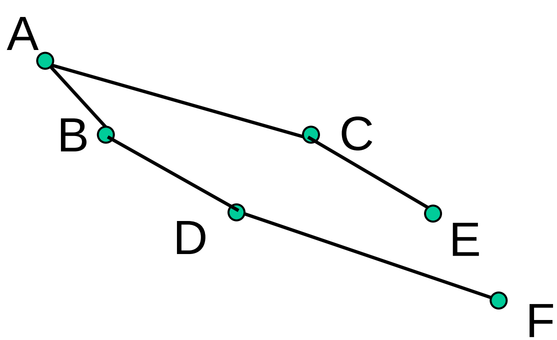


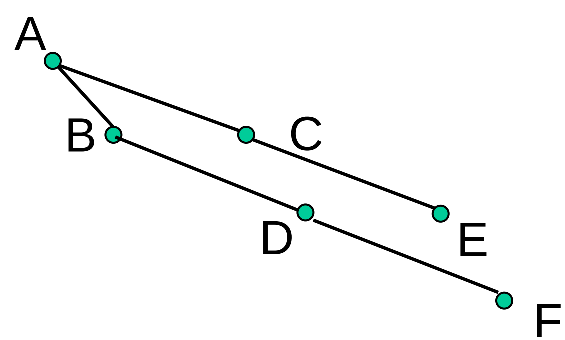
Gap Degree *dNh(T)*

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





Gap Degree *dNh(T)*

Coverage of a node $u \in T$

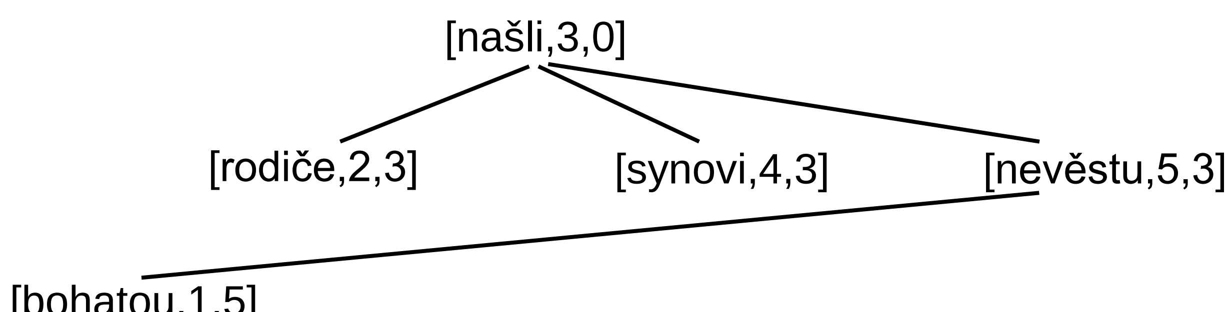
$Cov(u,T) = \{\, i \mid i$ - word order position of $v \in T$ such that, $u \leq_D v \,\}$

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 $Cov(u,T)$

Tree Gegree Degree $dNh(T) = \max \{dNh(u,T) \mid u \in T\}$

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



Gap Degree *dNh(T)*

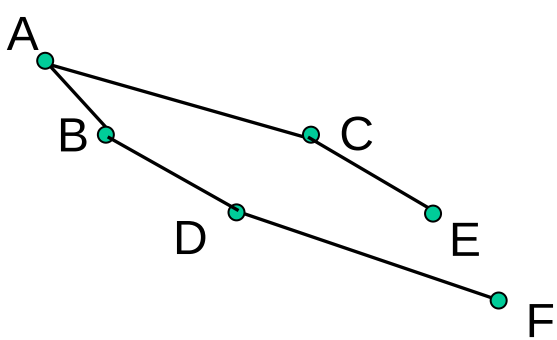
Coverage of a node $u \in T$

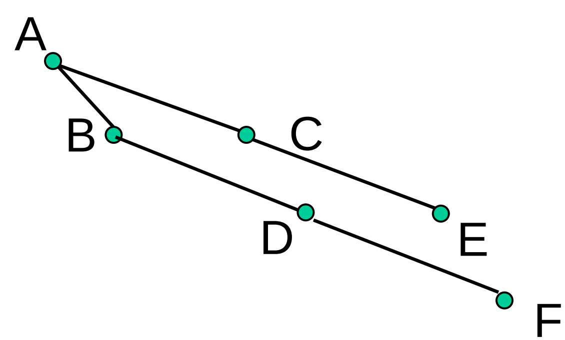
$Cov(u,T)$ = { i | i - word order position of $v \in T$ such that, u $\leq_D$ v }

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

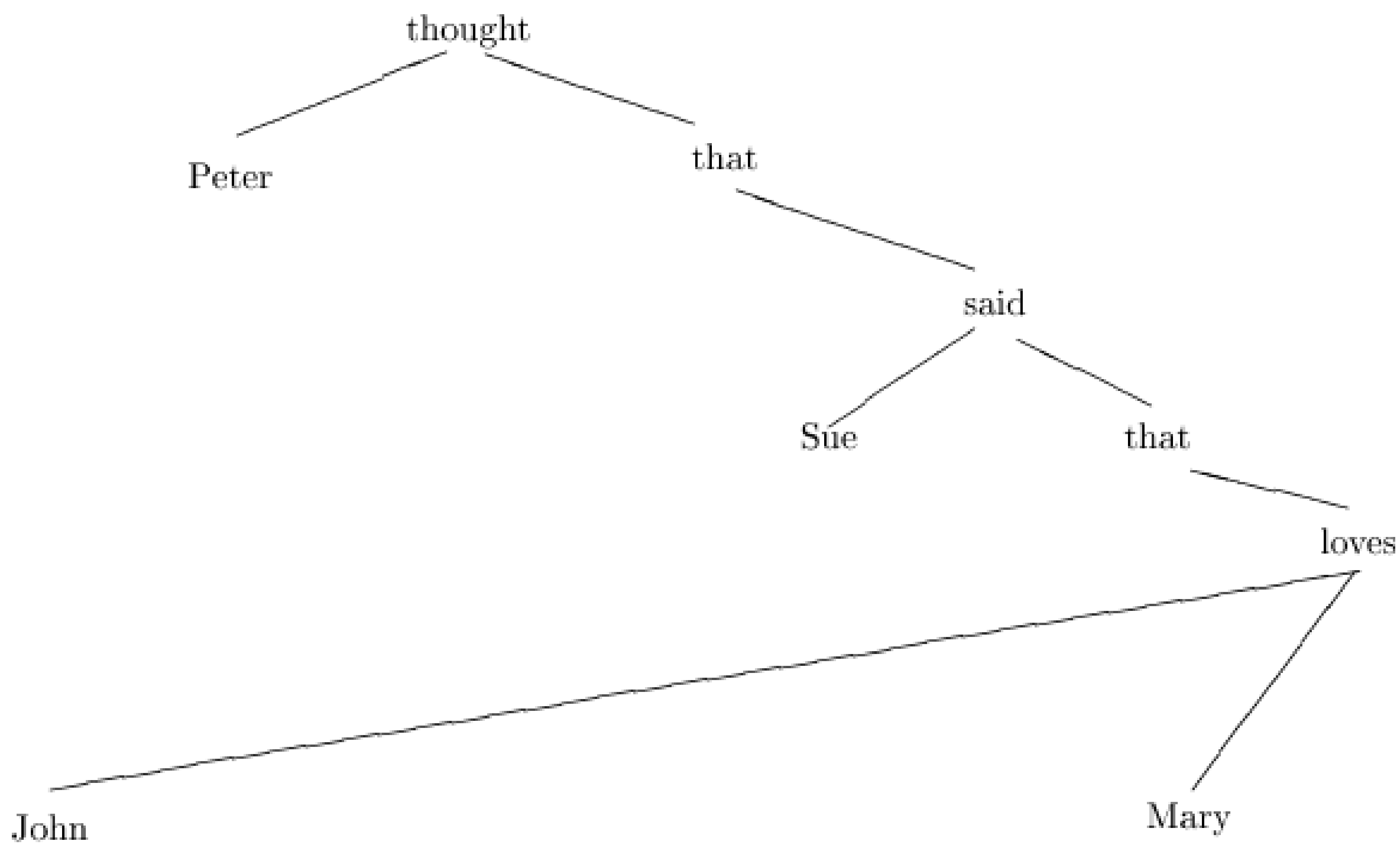




Projectivity and free word order

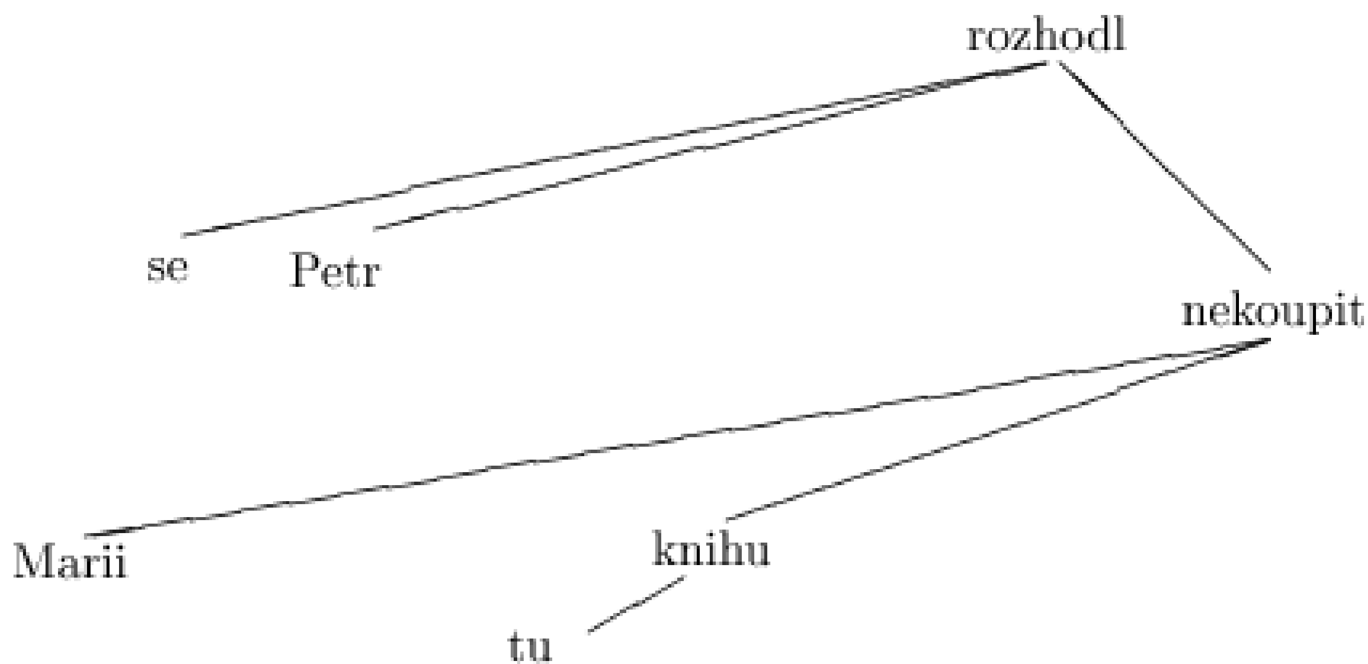
English: long-distance unbounded dependency

John, Peter thought that Sue said that Mary loves.



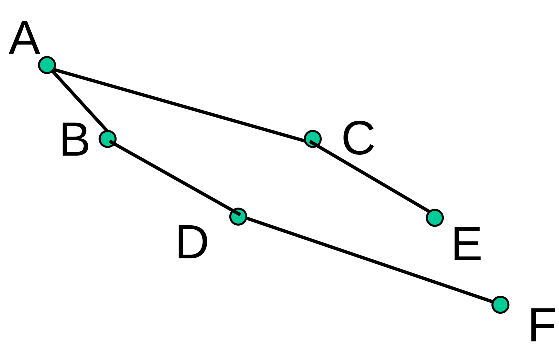
Projectivity and free word order

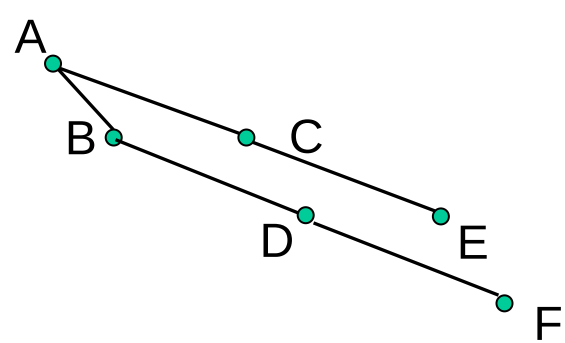
Czech:
Marii se Petr tu knihu rozhodl nekoupit.
to-Mary PART Peter that book decided not-buy
[Peter decided not to buy that book to Mary.]



Edge Degree

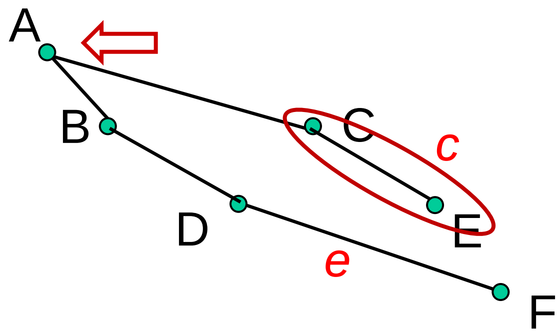
Let $T = (N, E)$ dependency tree, $e = [i, j]$ an edge in E,
T_e the subgraph of T induced by the nodes contained in the span of e.
Degree of an edge $e \in E$, *ed(e)*, is the number of connected components c in T_e such that the root of c is not dominated by the head of e.
Edge degree of T, ed(T) …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

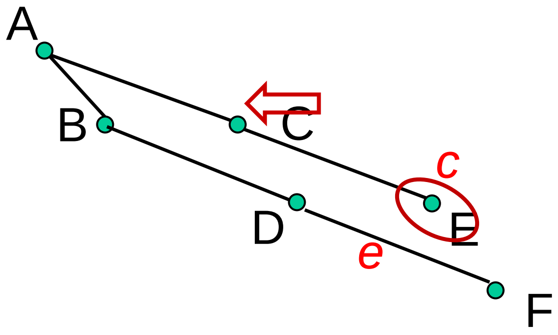




Edge Degree

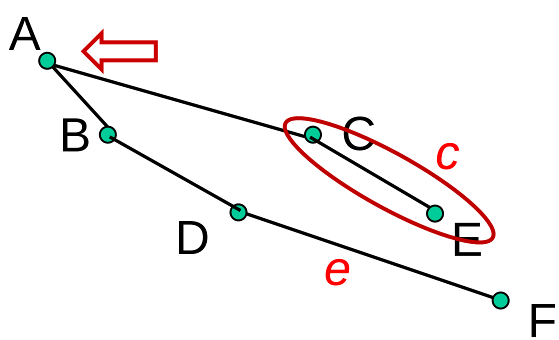
Let $T = (N, E)$ dependency tree, $e = [i, j]$ an edge in E,
T_e the subgraph of T induced by the nodes contained in the span of e.
Degree of an edge $e \in E$, *ed(e)*, is the number of connected components c in T_e such that the root of c is not dominated by the head of e.
Edge degree of T, ed(T) …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

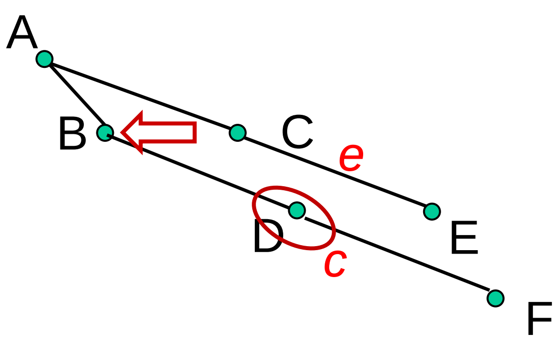




Edge Degree

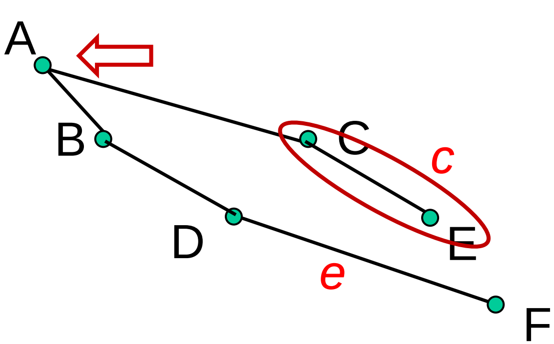
Let $T = (N, E)$ dependency tree, $e = [i, j]$ an edge in E,
T_e the subgraph of T induced by the nodes contained in the span of e.
Degree of an edge $e \in E$, *ed(e)*, is the number of connected components c in T_e such that the root of c is not dominated by the head of e.
Edge degree of T, ed(T) …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

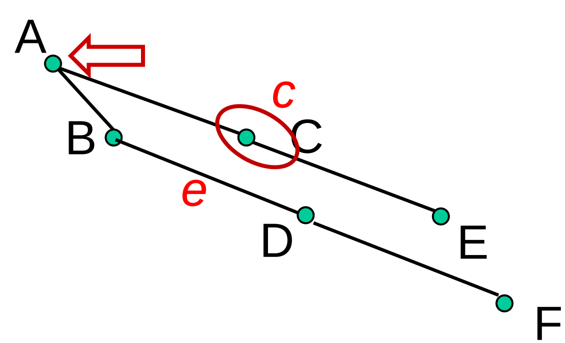




Edge Degree

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 $\{ed(e) \mid e \in T\}$





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%

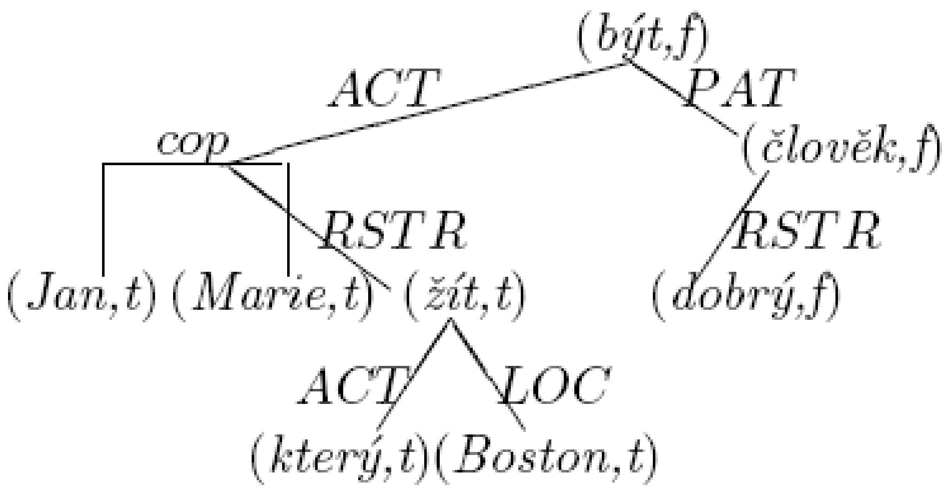
Kuhlmann, M., Nivre, J. (2006)

References

- Hajičová, E., Havelka, J., Sgall, P., Veselá, K., Zeman, D. (2004) Issues of Projectivity in the Prague Dependency Treebank. *PBML,* vol. 81
- 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
- Kuhlmann, M., Nivre, J. (2006) Mildly Non-Projective Dependency Structures. In COLING/ACL Main Conference Poster Sessions, 507–514.
- Mel'čuk, I. (1988) *Dependency Syntax: Theory and Practice.* State University of New York Press, Albany
- Partee, B. H.; ter Meulen, A.; Wall, R. E. (1990) *Mathematical Methods in Linguistics*. Kluwer Academic Publishers
- Petkevič, V. (1995) A New Formal Specification of Underlying Structure. *Theoretical Linguistics*, vol. 21, No.1
- Sgall, P., Hajičová, E., Panevová, J. (1986) *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reidel Publishing Company, Dordrecht/Academia, Prague
- Štěpánek, J. (2006) *Závislostní zachycení větné struktury v anotovaném syntaktickém korpusu*. PhD Thesis, MFF UK

Coordination/apposition in dependency trees

Petkevič (1995) … formal representation of FGD



$$\langle[(Jan,t);\ (Marie,t)]_{cop}\ {}_{RSTR}\langle\langle(který,t)\rangle_{ACT}\ (žít,t)\ {}_{LOC}\langle(Boston,t)\rangle\rangle\rangle_{ACT}\ (být,f)\ {}_{PAT}\langle\langle(dobrý,f)\rangle_{RSTR}\ (člověk,f)\rangle$$