The Prague Dependency Treebank and Valency Annotation (part 4)

Jan Hajič
Institute of Formal and Applied Linguistics
School of Computer Science
Faculty of Mathematics and Physics
Charles University, Prague
Czech Republic

Prague Dependency Treebank
Deep syntax & valency (part 4)

- Valency in the PDT
  - Valency lexicon for PDT
  - General valency lexicon
- Valency in deep vs. surface syntax
  - Links between the layers w.r.t. valency
- Valency and word sense
  - Sense-disambiguated occurrences:
    - Links from data to the lexicon
- Valency in translation, text generation

Definition of Valency

- Ability (“desire”) of words (verbs, nouns, adjectives) to combine themselves with other units of meaning
- Properties of valency:
  - Specific for every word meaning (in general)
    - leave: *sb left sth for sb* vs. *sb left from somewhere*
    - same as in PropBank *leave.02* vs. *leave.01*
  - Typically strongly correlates with surface form
    - morphological case (~ ending), preposition+case, ...
  - ~~Semantic constraints~~ are very dangerous

Structure of Valency

- word (lemma)
  - word sense group 1
    - valency frame:
      - $slot_1$ $slot_2$ $slot_3$
    - surface expression
  - word sense group 2
    - ...

vyměnit (*to replace*)

vyměnit$_1$

ACT PAT EFF

Nom. Acc. za+Acc.

vyměnit$_2$

...

The Valency Lexicon PDT-VALLEX

- Valency frames
  - each verb, some nouns, adjectives
- Basic set prepared in advance, annotators add entries on-the-go, checking and approval process follows (consistency)
- VALLEX
  - more detailed and complex annotation of valency
  - Žabokrtský, Lopatková (2005), VALLEX 1.0
  - All about valency: http://ufal.ms.mff.cuni.cz/~semecky/vallex/

PDT-VALLEX Entry

- dosáhnout: *“to reach”, “to get [sb to do sth]”*
- browser/user-formatted example:

*** dosáhnout**

ACT(.1) PAT(.2,.4) v-w714f1 **Used:** 272x

dosáhnout určité úrovně

mzda d. v tomto oboru 80 tisíc

d. pokročilého věku

ACT(.1) PAT(.2,*aby*[.v]) ?ORIG(*na-1*[.6],*od-1*[.2]) v-w714f2 **Used:** 7x

dosáhl na něm slibu

dosáhli na sobě slibu

ACT(.1) DPHR(*svůj-1*.2) v-w714f3 **Used:** 2x

dosáhl svého

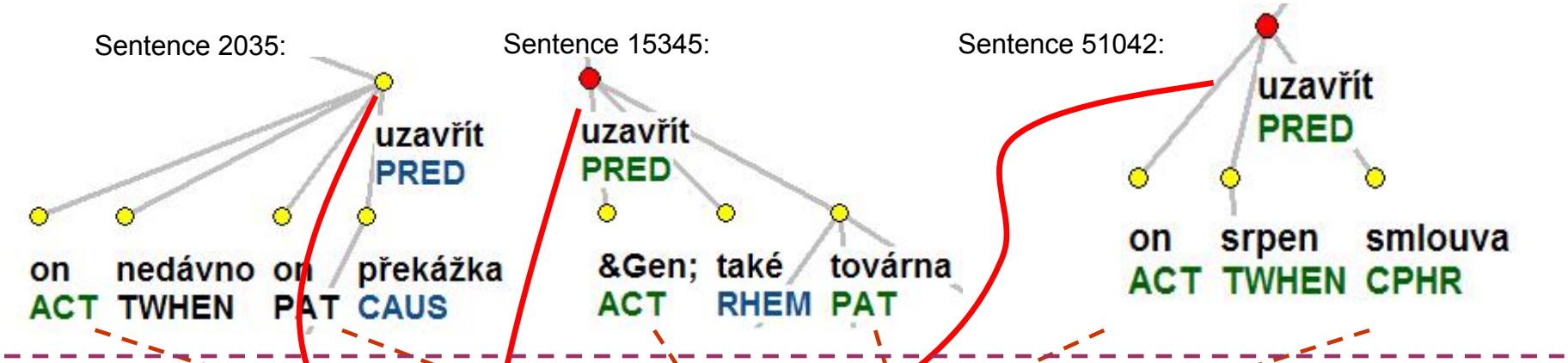
ACT(.1) DIR3(*) v-w714f4 **Used:** 2x

dosáhl na strop

rukou.MEANS

Corpus <-> Valency Lexicon

- Corpus:



- Lexicon:

ENTRY: uzavřít

vf_1: ACT(.1) CPHR({smlouva}.4)

ex: u. dohodu *(close a contract)*

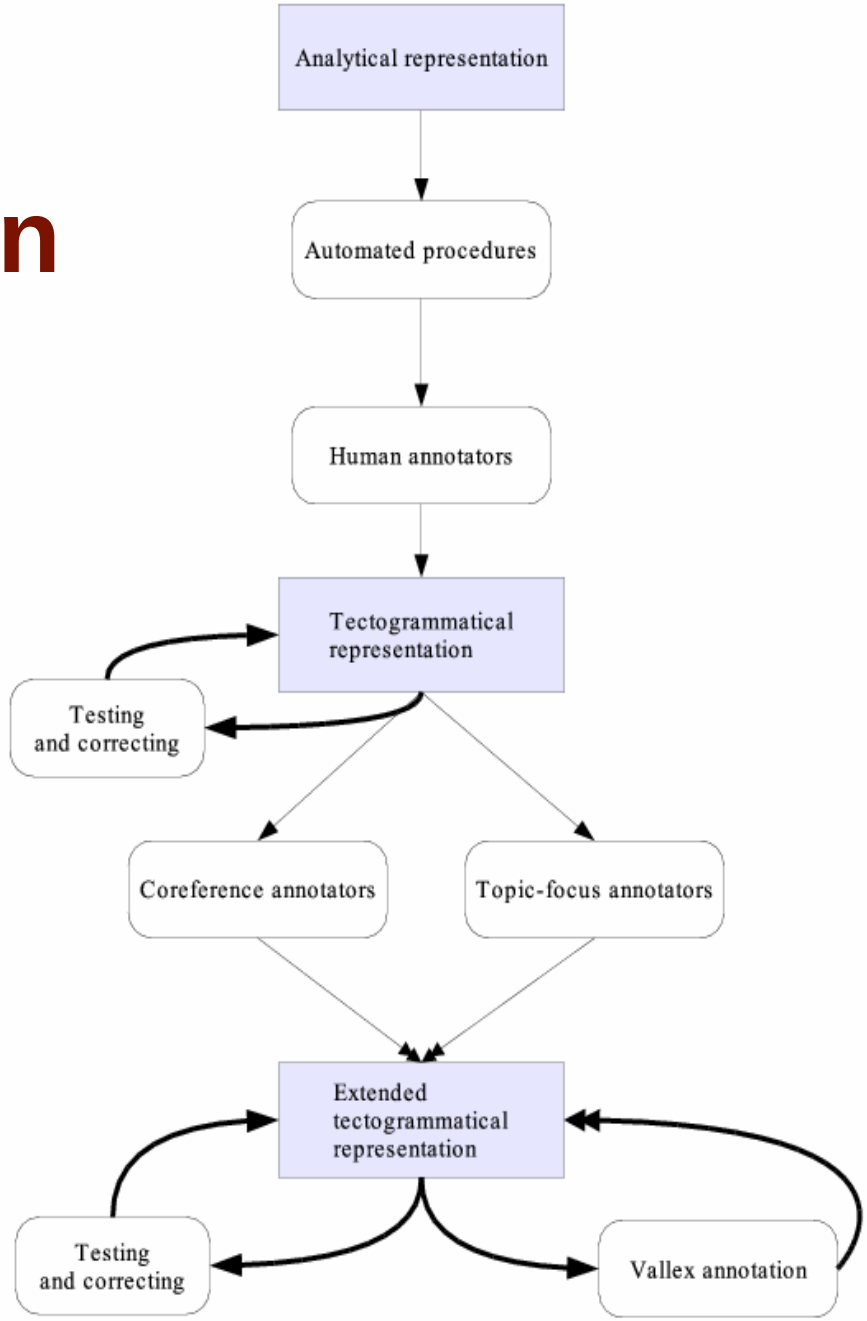
vf_2: ACT(.1) PAT(.4)

ex.: u. pokoj *(close a room, house)*

The Annotation Process

- 4 sublayers
  - work on structure first, rest in parallel
- Structure
  - automatic preprocessing - programmed conversion from analytical layer annotation
- Grammatemes
  - mostly automatically (based on lower layers' annotation), manual checking, corrections
- Cross-sublayer/cross-layer checking
  - partly automatic, then manual

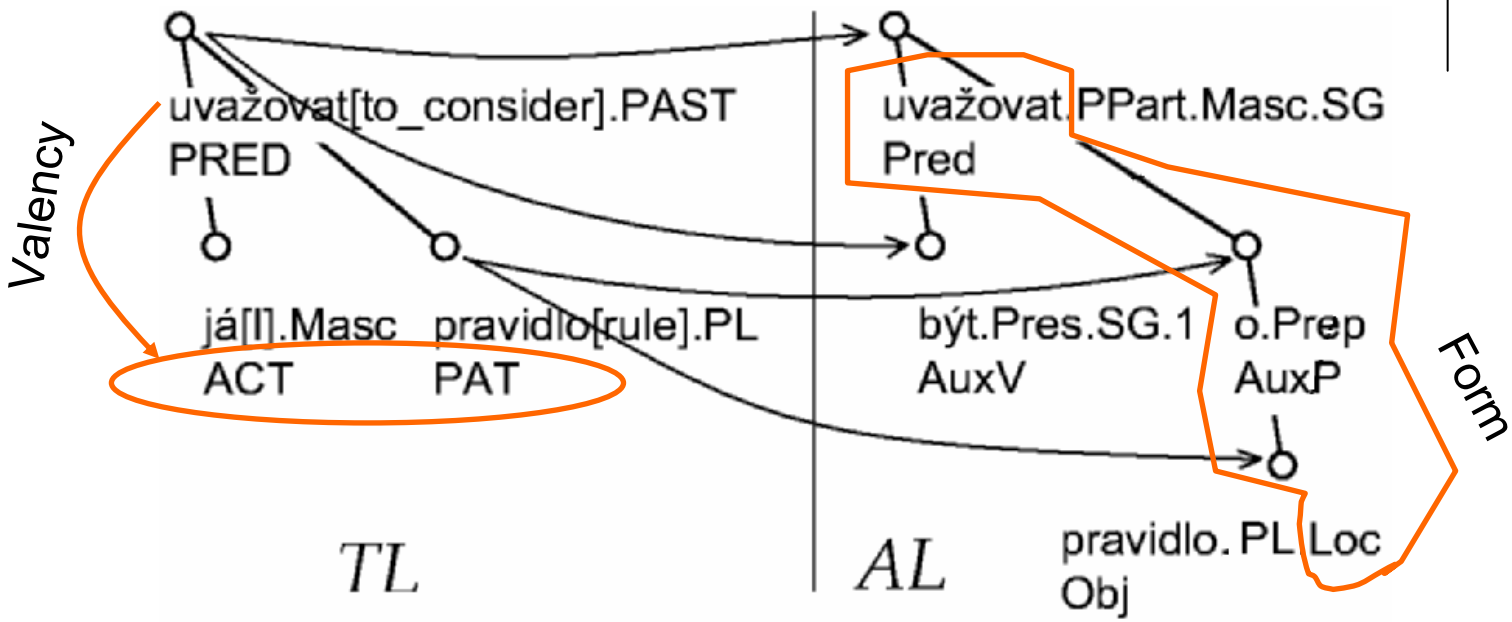
The Annotation Process Scheme



Valency & Tectogrammatical Annotation

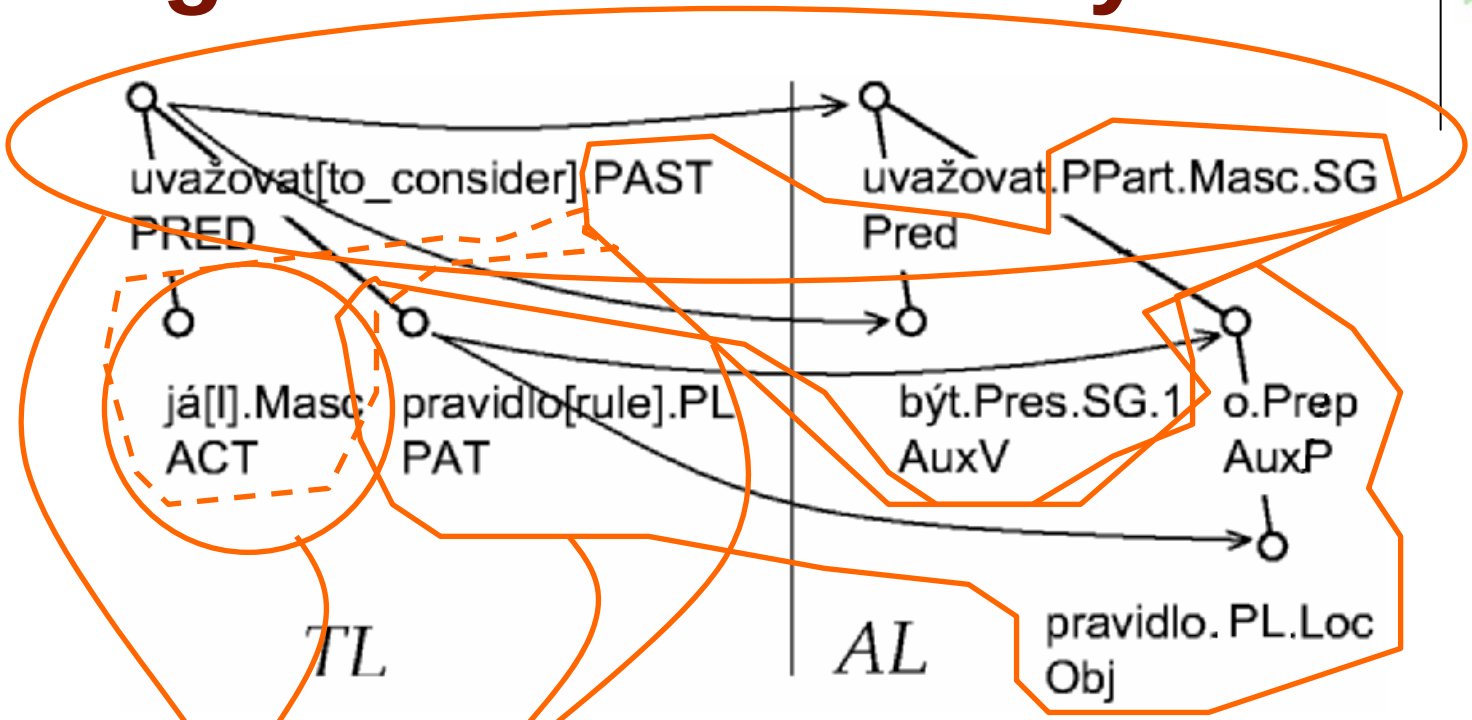
- Valency and...
  - (surface) form
- Annotation tools
  - TrEd
    - structural annotation
    - valency lexicon integration
- Search
  - TrEd, Netgraph

Valency & Form



lemma (AL): uvažovat
ACT: surface ellipsis, node disappears
PAT: preposition ‘o’ and a locative case

Tectogrammatical / Analytical



uvažovat – uvažovat
PAST / já.Masc – PPart.Masc.SG(Pred) / být.Pres.SG.1(AuxV)
pravidlo.PL.PAT – o.Prep(AuxP) / pravidlo.PL.Loc(Obj)
já - 0

?

CONTEXT NEEDED

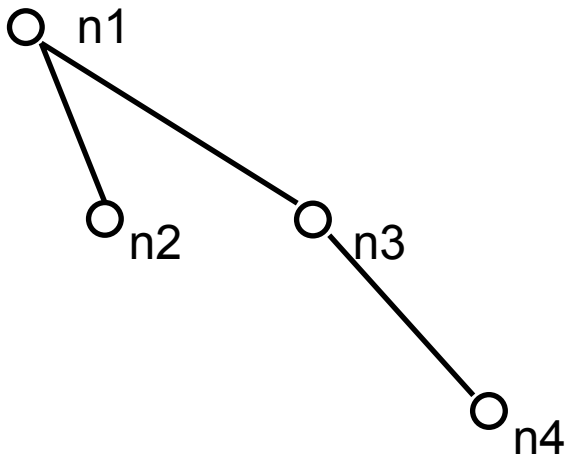
from another sentence: pravidlo.PL.PAT – pravidlo.PL.Acc(Obj)

Valency & Form

- Valency frame:
  - (per each sense of word)
  - (obligatory) modifiers ↔ functors
  - functor → form
- Simplest case:
  - surface form of a functor: particular case
  - Ex.: ACT in nominative (<u>he</u> says)
  - Ex.: PAT in accusative (she sees <u>him</u>)
- ... but it is not always so simple (as we have already seen)!

Valency & Form: Constraints

- Tree structure:



- (Sets of) Constraints:
  - n1: lemma=uvažovat mode=active
  - n2: case=Nom afun=Sb
  - n3: lemma=o afun=AuxP
  - n4: case=Loc afun=Obj

(General) Valency Lexicon Entries

Entry	Sense #	Frame #	Valency Optinality		Form alternatives
1	1	1	ACT PAT		 $\{c_i\}$  $\{c_i\}$
	2	2	ACT PAT LOC		 $\{c_i\}$  $\{c_i\}$
		3	ACT PAT DIR3		 $\{c_i\}$
	3	4	ACT PAT		 $\{c_i\}$  $\{c_i\}$
2	1	1	ACT		 $\{c_i\}$
	2	2	ACT INTT		 $\{c_i\}$  $\{c_i\}$
3	1	1	ACT PAT		 $\{c_i\}$  $\{c_i\}$
	2	2	ACT PAT		 $\{c_i\}$  $\{c_i\}$

Valency Lexicon Simplification

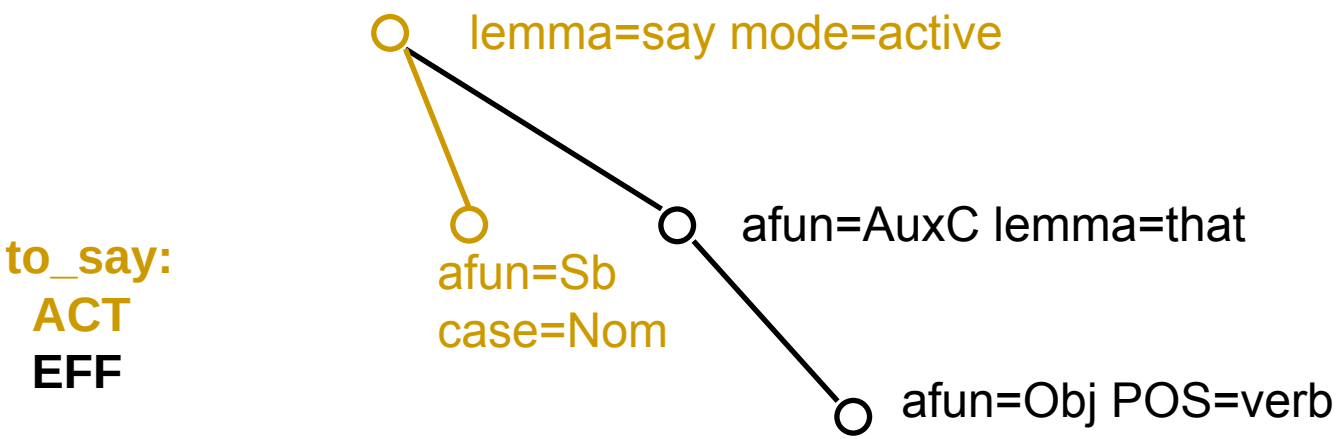
- Independent form for each slot of a particular valency frame
  - ACT, PAT, ...: own constraint, not a global one
- $\text{Functor}_{\text{oblig./opt.}} \quad \leftrightarrow \quad \text{constraints}_{\text{Functor}}$
- Ex.:
  - lemma1 ACT(Nom.) PAT(o+6) (to consider a rule)
  - lemma2 ACT(Nom.) PAT(4) (create a rule)
- Standard “transformations” of frame form
  - passivization, reflexivization, ...

Example: Valency & Form

- Simple 1:1:
  - ex.: create: ACT(Nom) PAT(Acc)
  - verb in infinitive: INTT(Inf)
  - subordinate clause: PAT(verb)
  - class of words with generic verbs: CPHR({class})
  - no constraint: (often) LOC, TWHEN
    - general constraint for a given functor applies
  - ...more!

Example: Valency & Form

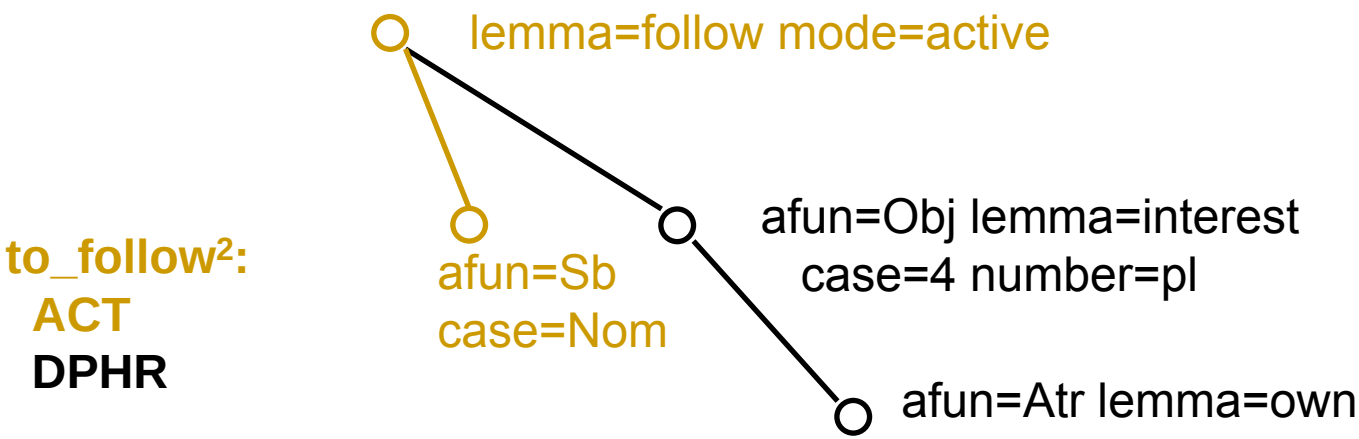
- 1:2
  - relative clause



- linear representation: EFF(that[.v])

Example: Valency & Form

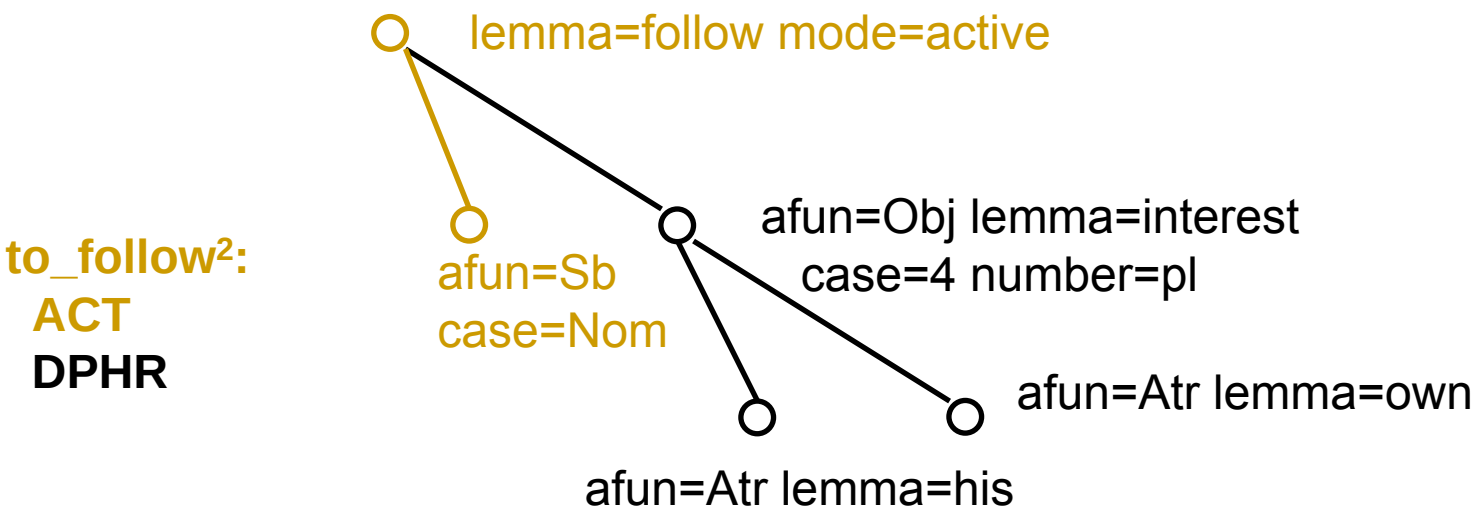
- 1:2
  - idomatic phrase



- linear representation: DPHR(interest.P4[own.#])

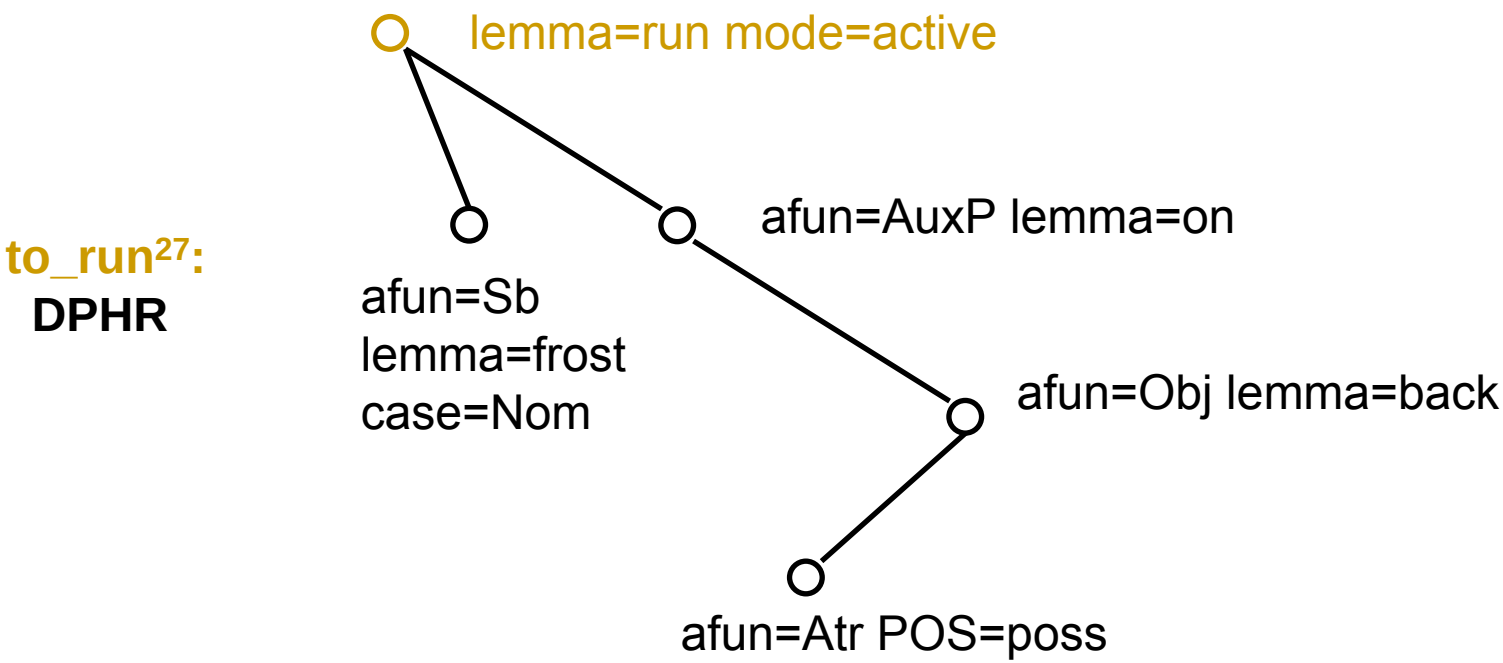
Example: Valency & Form

- 1:3
  - idomatic phrase



Example: Valency & Form

- 1:4
  - idomatic phrase



Valency and Translation

- leave:
  - leave-1
    - to leave [from] somewhere
  - leave-2
    - to leave sth for sb
- Translating (from English into Czech):
  - which equivalent to chose?
    - nechat vs. odjet/opustit
  - which prepositions, cases, ... to use?
    - accusative vs. “z” (*“from”*) with genitive vs. ...?

Valency and Translation

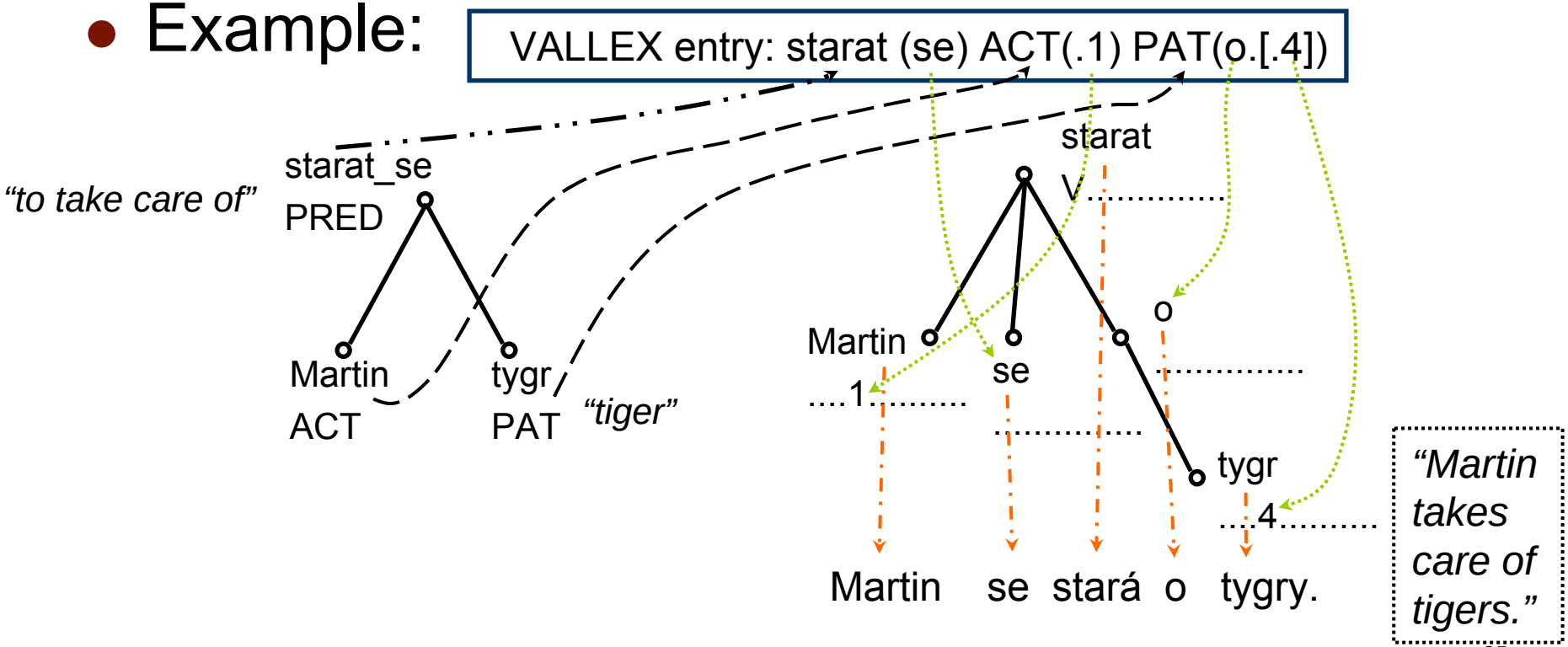
- leave-1 ⟷ nechat-3
  - ACT() PAT() LOC() ACT(.1) PAT(.4) LOC()
- leave-2 odjet-1
  - ACT() DIR1(from.) ACT(.1) DIR1(z.[.2])

Valency and Text Generation

- Tectogrammatical Representation
  - has *all* the information to (re)generate the surface form of the sentence:
    - in a “generalized” form
    - non-redundant (almost... but for generation, it is o.k.)
  - ...except the links to a-layer, however
    - links used only for *training* [statistical models for] parsing/generation modules
    - not present when e.g. doing text planning, translation, ...
  - valency dictionary: form of “learned” knowledge

Valency and Text Generation

- Using valency for...
  - ...getting the correct (lemma, tag) of verb arguments
- Example:



Tectogrammatical Annotation Tools

- Manual annotation
  - 4 groups of annotators ~ 4 sublayers
  - Special graphical tool (TrEd)
    - Customizable graphical tree editor
- Preprocessing
  - Data from analytical layer, preprocessed
  - Online dependency function preassignment

The [Manual] Annotation Tool

- Perl/PerlTk based, platform-independent
  - Linux, Windows 95/98/2000, Solaris, ...
- Perl as the “macro” language
  - “unlimited” online processing capability
- Flexibility for interactive checking
  - split screen, graphical “diff” function
- Customization, printing, “plugins”, ...
- !! See also J. Stepanek’s lecture / tools

The “TrEd” Tree Editor

- Graphical tool

  TrEd

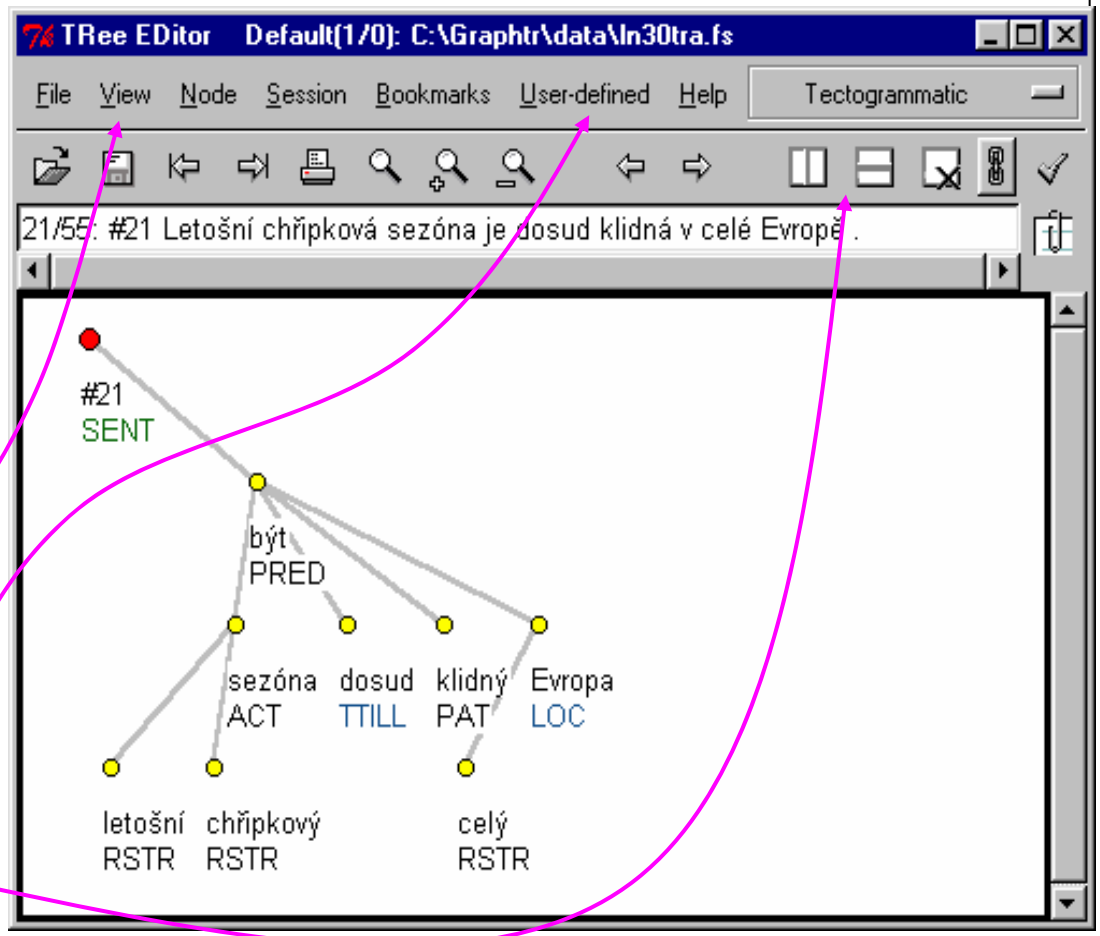
- Main screen:

Original sentence:
[This year’s flu season is still quiet in Europe.]

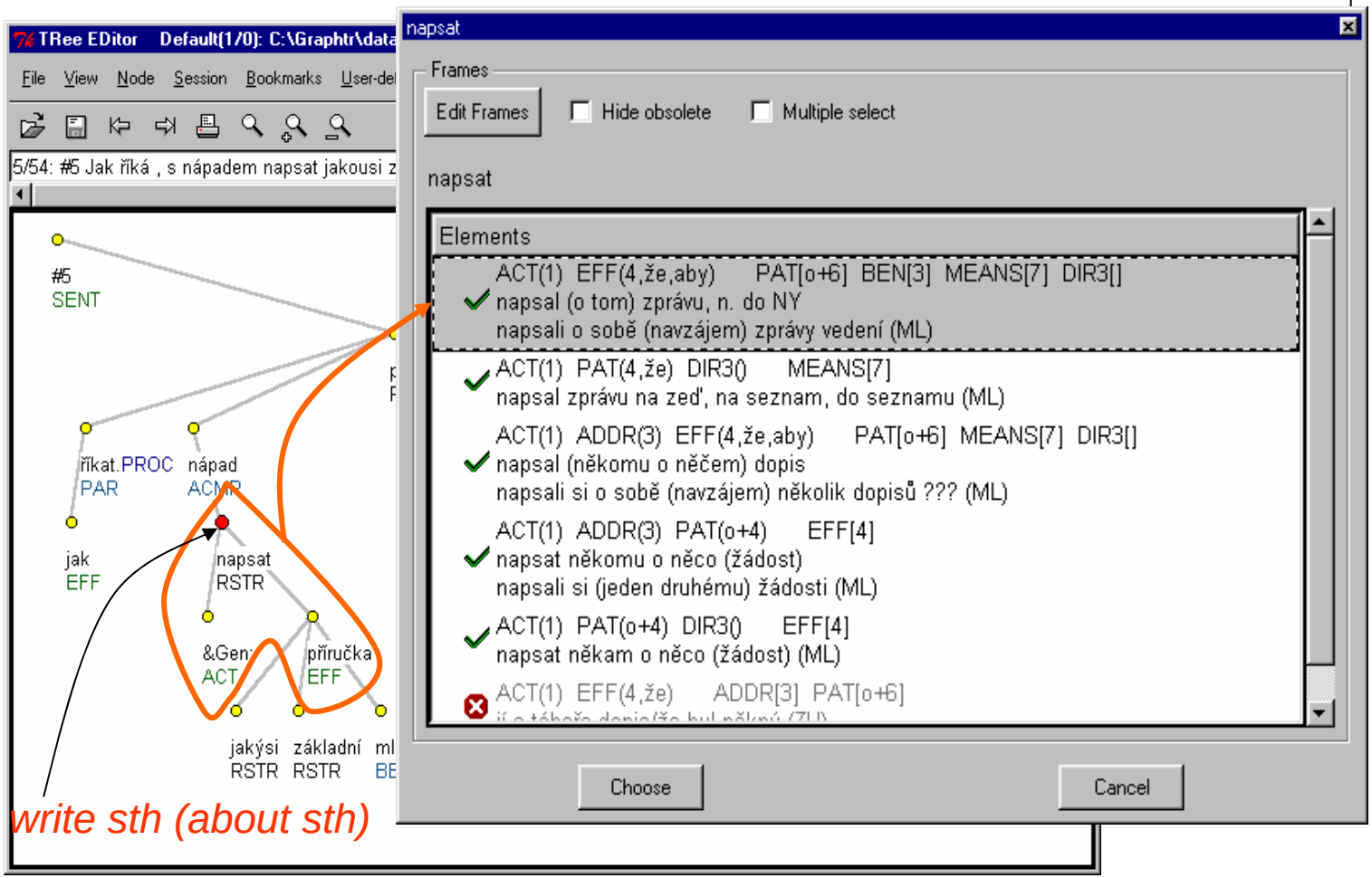
Editing window customization

Run a macro

Multiwindow editing/compare

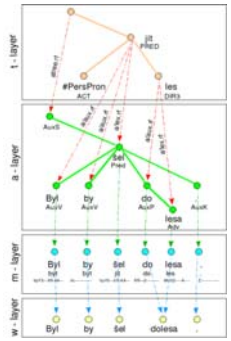


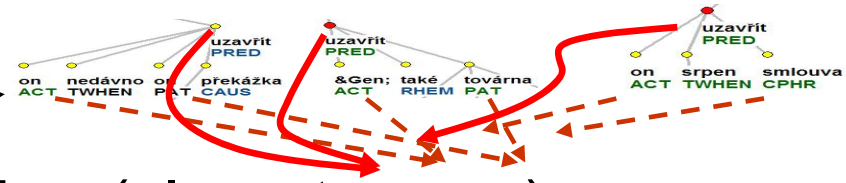
Valency Lexicon in TrEd



to write sth (about sth)

Annotating the Links





- Stand-off annotation principles
  - Links to another layer
  - Links to lexicon
- Minimal work on link annotation (close to zero)
- Macro commands in TrEd
  - transparently keeps track of merged nodes, splits, etc., and adapts links correspondingly.
- Result:
  - almost no extra work
  - final check after annotators do the last pass

The “Old” PDT 1.0

- Morphology (1.8MW) & Surface syntax (1.5MW)
- SGML format (csts.dtd) + compact “FS”
- Mixed (single-file) annotation
  - 7 attributes + dependency
- TrEd (graphical viewer/editor), NetGraph (search capability)
  - simple visualization

What's New in PDT 2.0

- Tectogrammatical layer (0.8MW)
  - 39 node attributes + dependency
  - valency dictionary (PDT-VALLEX)
- XML stand-off annotation ("PML", 4 layers)
- New data division (train/dtest/etest)
  - added morphological annotation to all data
  - corrections of PDT 1.0 files (morphology, syntax)
- Improved tools:
  - TrEd, btred/ntred (batch tree corpus processing)
    - new features, better visualization

Tectogrammatical attributes I

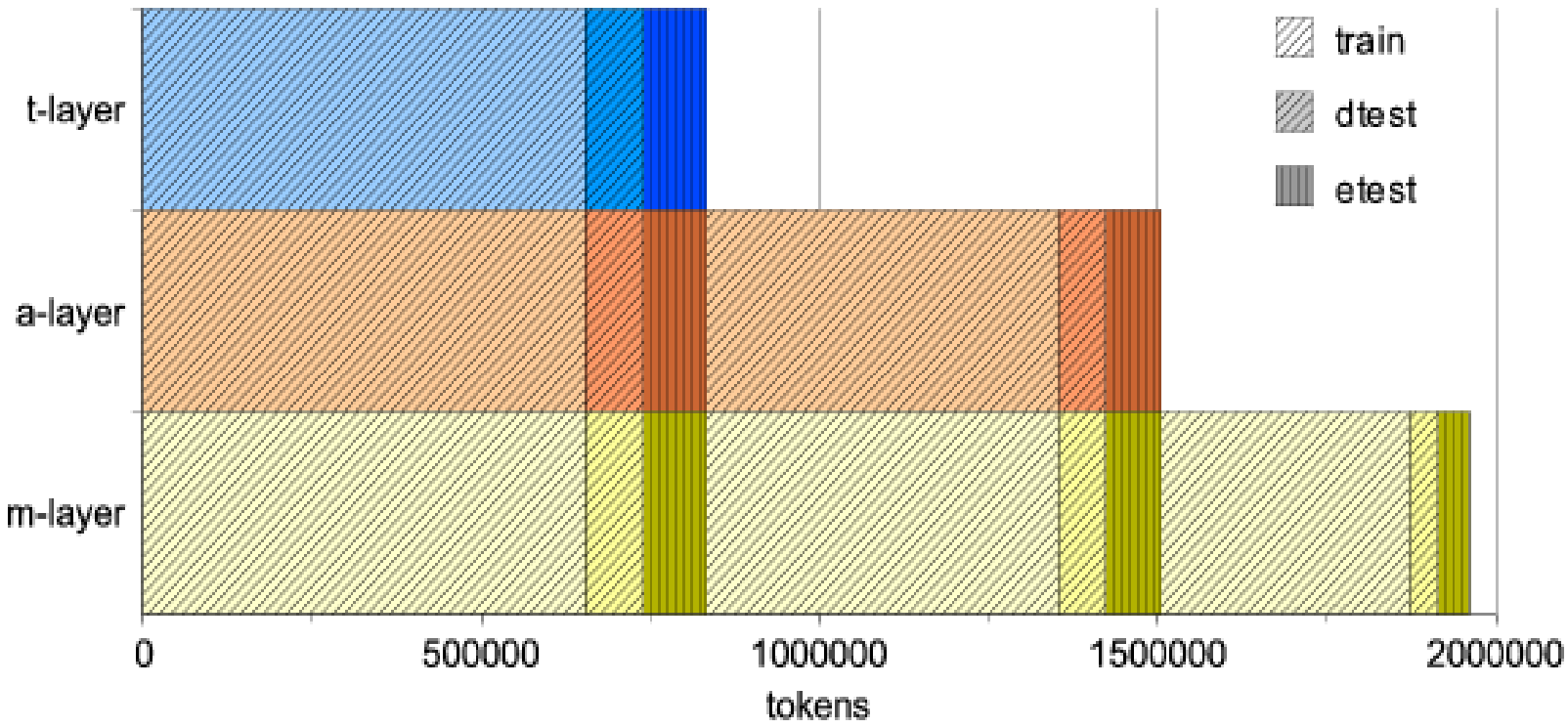
- node typing
  - complex, coap, qcomplex, root, atom, ...
- functor, subfunctor
  - TWHEN: TWHEN.basic, TWHEN.before
- is_member, is_generated, is_parenthesis, is_dsp_root, is_state, quot_type, ...
- grammatemes (16):
  - aspect, degcmp, deontmod, sempos, tense, indeftype, politeness, person, ...

Tectogrammatical attributes II

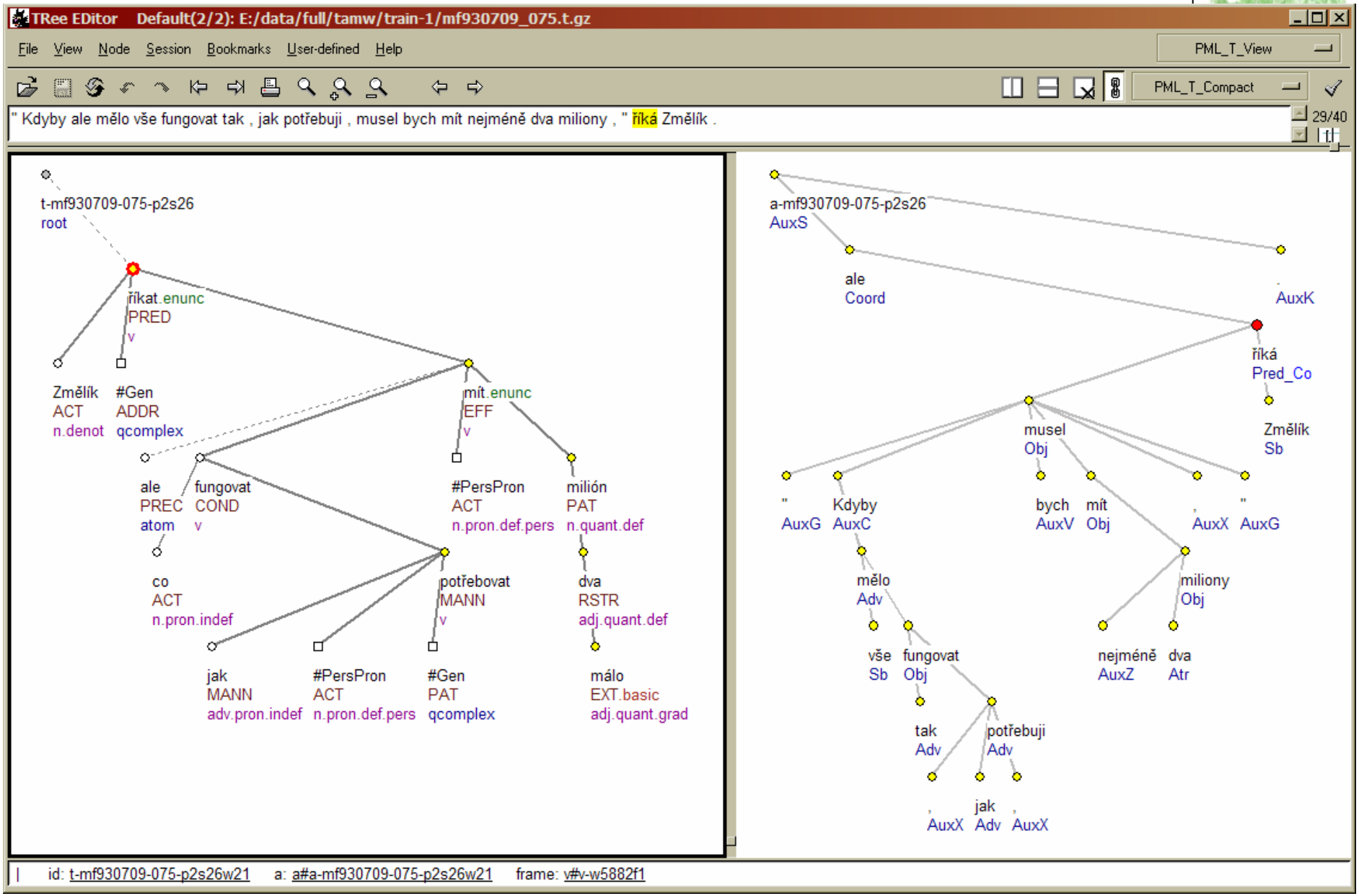
- topic/focus:
  - tfa, deepord
- valency: t_lemma, val_frame.rf
- bookkeeping: id
- coref_gram.rf, coref_text.rf, compl.rf
  - reference to TR node, type of coreference
- sentmod
- Linking to analytical layer
  - a.lex.rf (“main” anal. node), a.aux.rf (others)

PDT 2.0: The Data

- Data sizes



TrEd


TRee EDitor Default(2/2): E:/data/full/tamw/train-1/mf930709_075.t.gz
File View Node Session Bookmarks User-defined Help
PML_T_View
PML_T_Compact
" Kdyby ale mělo vše fungovat tak , jak potřebuji , musel bych mít nejméně dva miliony , " říká Změlík .
29/40
t-mf930709-075-p2s26
root
říkat.enunc
PRED
v
Změlík
ACT
n.denot
#Gen
ADDR
qcomplex
mít.enunc
EFF
v
ale
PREC
atom
fungovat
COND
v
#PersPron
ACT
n.pron.def.pers
milión
PAT
n.quant.def
co
ACT
n.pron.indef
potřebovat
MANN
v
dva
RSTR
adj.quant.def
jak
MANN
adv.pron.indef
#PersPron
ACT
n.pron.def.pers
#Gen
PAT
qcomplex
málo
EXT.basic
adj.quant.grad
a-mf930709-075-p2s26
AuxS
ale
Coord
.
AuxK
říká
Pred_Co
musel
Obj
Změlík
Sb
"
AuxG
Kdyby
AuxC
bych
AuxV
mít
Obj
,
AuxX
"
AuxG
mělo
Adv
miliony
Obj
vše
Sb
fungovat
Obj
nejméně
AuxZ
dva
Atr
tak
Adv
potřebuji
Adv
,
AuxX
jak
Adv
,
AuxX
id: t-mf930709-075-p2s26w21
a: a#a-mf930709-075-p2s26w21
frame: v#v-w5882f1

Using the Results (t-layer)

- Preliminary!
  - PDT 2.0 published July 2006
  - 50k sentences for training (t-layer)
- Functor assignment
  - > 80% accuracy on manually annotated structure
- Tectogrammatical parser
  - Part of the “toolchain” (run_all, see p. 5, p. 7, J. Štěpánek)
- Coreference
  - preliminary results: > 80%
- Valency
  - frame assignment > 70%

To take home...

- What is PDT
  - Dependency-based treebank project
    - Czech (other languages in the works)
  - ~ 1mil. words
    - sufficient size for ML experiments
  - 4 layers of annotation
    - token, morphology, syntax, deep syntax/semantics++)
    - independent and full information at all levels, but...
    - interlinked (for the development of parsers/generators)
  - Valency dictionary integrated (links from data)

Some (more) pointers

- http://ufal.mff.cuni.cz/pdt2.0
  - Current version of PDT, all three levels, 1.9/1.5/0.8 Mw
- http://ufal.mff.cuni.cz/REST/CAC/CAC.html
  - The Czech Academic Corpus, v 1.0
- http://www.ldc.upenn.edu
  - LDC2001T10 (PDT v1.0), LDC2004T23 (PADT 1.0), LDC2004T25 (PCEDT 1.0)
- http://www.clsp.jhu.edu: Workshop 2002
  - Using TL for MT Generation