Netgraph – a Tool for Searching in the Prague Dependency Treebank 2.0

Defence of the Doctoral Thesis, Prague, September 3rd, 2008

Author: Mgr. Jiří Mírovský
Charles University in Prague, Institute of Formal and Applied Linguistics

Supervisor: Prof. RNDr. Jan Hajič, Dr.
Charles University in Prague, Institute of Formal and Applied Linguistics

Opponents: RNDr. Roman Ondruška, Ph.D.
SUN Microsystems
Ing. Alexandr Rosen, Ph.D.
Charles University in Prague, Institute of Theoretical and Computational Linguistics

The Starting Point

Three sides existed whose connection needed to be solved:

- The Prague Dependency Treebank
- Netgraph 1.0
- Users without programming skills

The Work

The doctoral thesis consists of these main parts:

- Analysis of the Requirements
- Designing the Query Language
- Usage of the Query Language
- Comparison to Other Search Systems
- Implementation in Netgraph

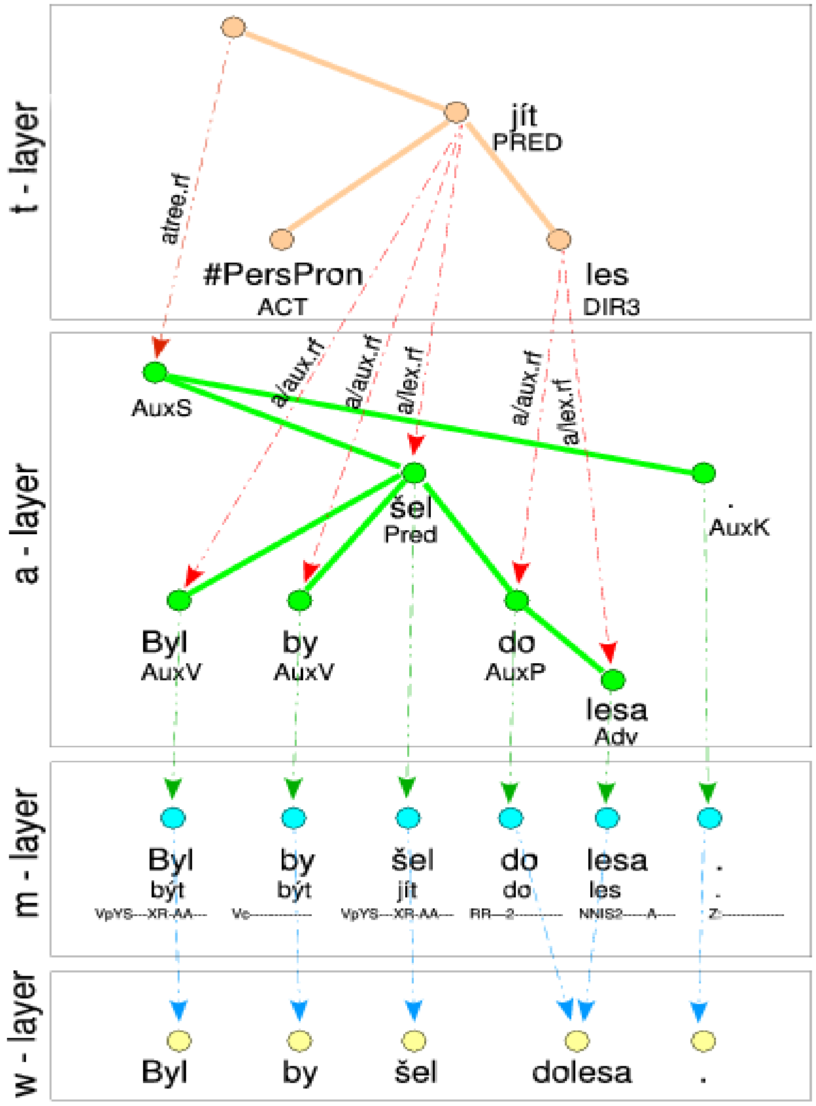
The Work

In this presentation, I will focus on these parts:

- Analysis of the Requirements
- Designing the Query Language
- Usage of the Query Language
- Comparison to Other Search Systems
- Implementation in Netgraph

Prague Dependency Treebank 2.0

Layers in PDT 2.0



Linguistic Requirements
Valency

To study valency, the query language should be able to:

- control a presence of a particular type of son
- control a non-presence of a son
- control number of sons

Linguistic Requirements
Coordination etc.

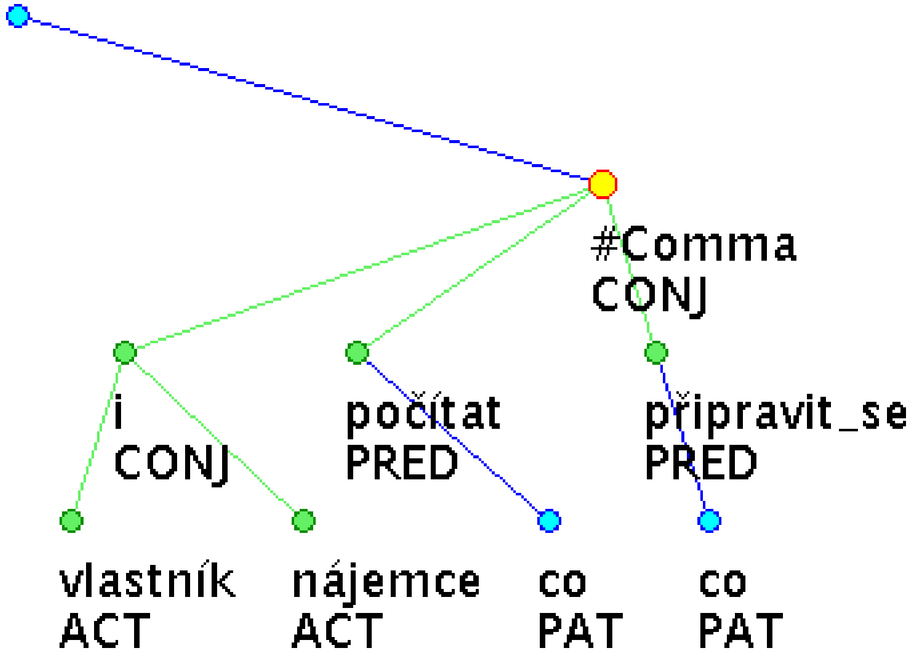
Tree dependency is not always linguistic dependency. We need:

➡ to skip a node (etc. coordination, apposition)

Linguistic Requirements
Complex Example of Coordination

Czech: S čím mohou vlastníci i nájemci počítat, na co by se měli připravit?

English (lit.): What can owners and tenants expect, what they should get ready for?



Linguistic Requirements
Coordination etc.

Tree dependency is not always linguistic dependency. We need:

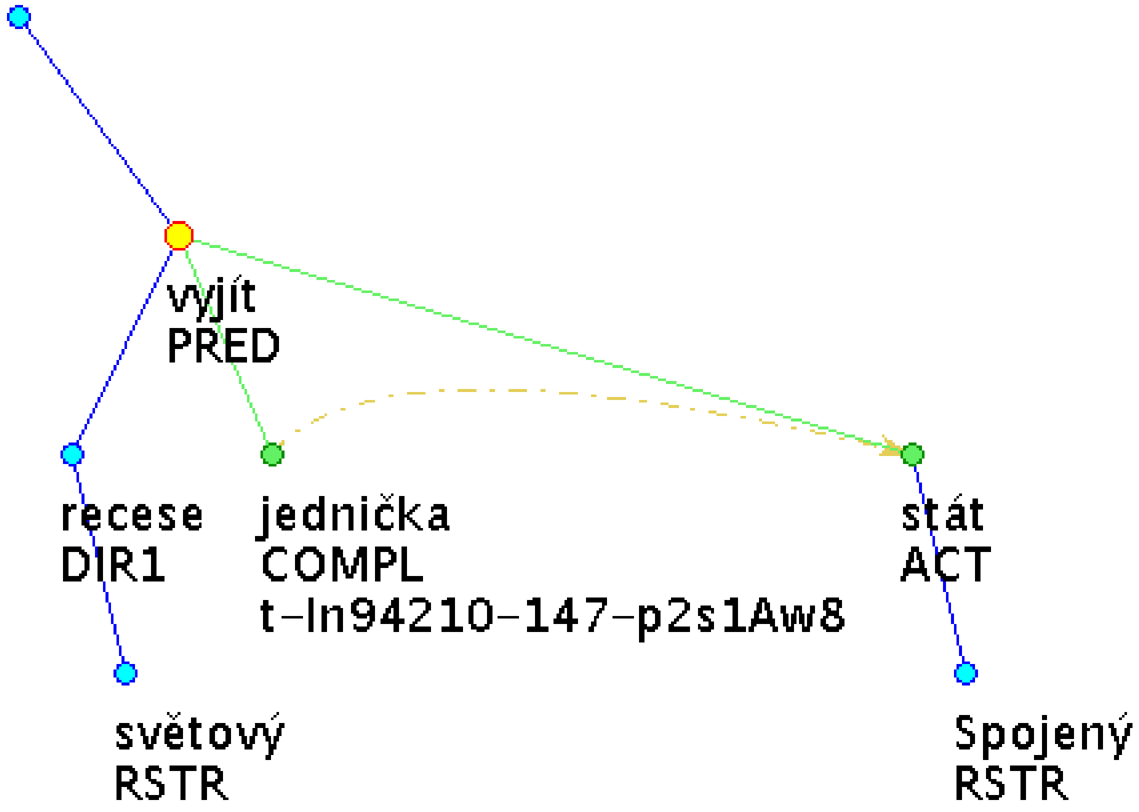
- to skip a node (etc. coordination, apposition)

- even better: to set a linguistic dependency

Linguistic Requirements
Predicative Complement

Czech: Ze světové recese vyšly jako jednička Spojené státy.

English: The United States emerged from the world recession as number one.



Linguistic Requirements
Predicative Complement

The dual dependency is represented by means of a **reference** to another node (attributes `compl.rf` and `id`). We need:

➡ to match values unknown at the time of creating the query

Linguistic Requirements *...other phenomena*

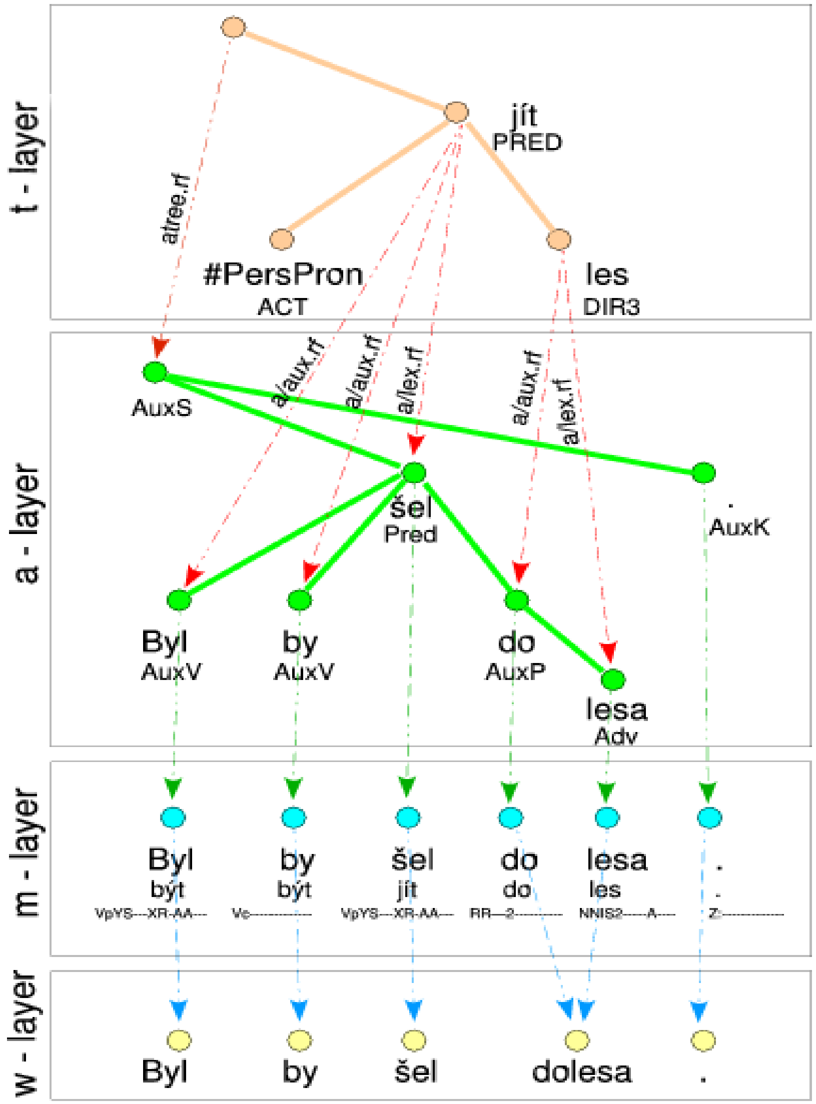
- *Topic – Focus, Focus proper* – combination of references, non-existence of a node and transitive closure of dependency; relation “<”
- *Rhematizers* – closest left son, closest left brother
- *(Non-)projectivity* – multiple-tree query to combine several one-tree queries representing different orientations of non-projective edges

Linguistic Requirements
...other phenomena

- *Idioms etc.* – searching in the linear form of the sentence (with regular expressions)
- *Agreement* – reference to only a part of a value of an attribute of another node (e.g. the fifth position of the morphological tag for case)
- *Word order* – measuring the horizontal distance between words

Linguistic Requirements

Layers in PDT 2.0



Linguistic Requirements
Accessing Lower Layers

Queries across the layers of annotation:

- A PATient expressed with a preposition k and a noun in the dative
- A PATient more dynamic than an ACTor but on the left side from it in the sentence

⇨ We need to have means of accessing the lower layers.

Linguistic Requirements
Summary

Evaluation of a node

- multiple attributes evaluation
- alternative values
- alternative nodes (alternative evaluation of the whole set of attributes)
- wild cards (regular expressions)
- negation, relations other than “equal to”

Linguistic Requirements
Summary

Dependencies between nodes
(vertical relations)

- direct, transitive (existence, non-existence)
- vertical distance (from root, from one another)
- number of sons (zero for leaves)

Linguistic Requirements
Summary

Horizontal relations

- precedence, immediate precedence
- negation of it
- horizontal distance

Secondary relations

- secondary dependencies, coreferences

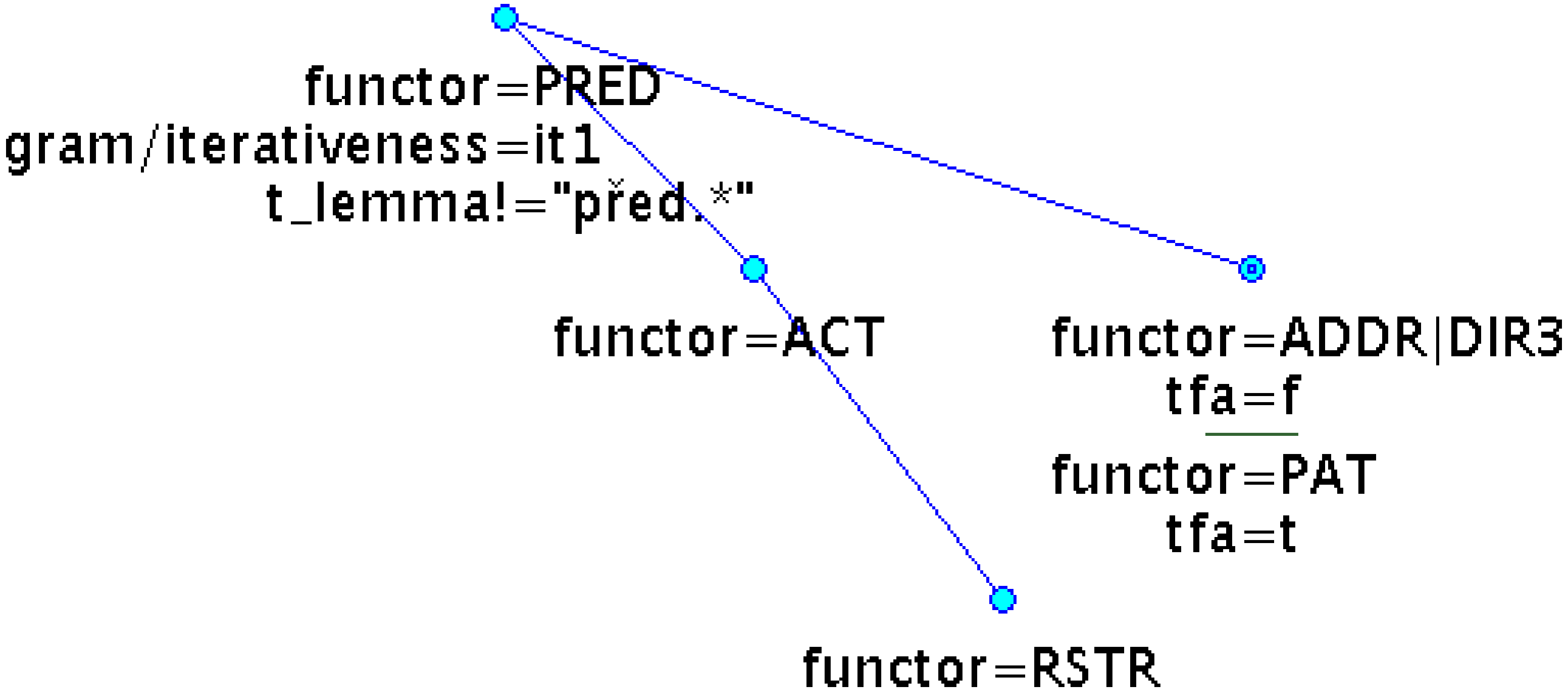
Linguistic Requirements
Summary

Other features

- multiple-tree queries
- accessing several layers of annotation at the same time
- searching in the linear form of the sentence

Netgraph Query Language

The Basics



functor=PRED
gram/iterativeness=it1
t_lemma!="před.*"

functor=ACT

functor=ADDR|DIR3
tfa=f

functor=PAT
tfa=t

functor=RSTR

Netgraph Query Language

Main additions to the query language of Netgraph 1.0:

- Meta-attributes
- References to attributes of other nodes
- Multi-tree queries
- Hidden nodes for a multilayer access

Netgraph Query Language
Meta-Attributes

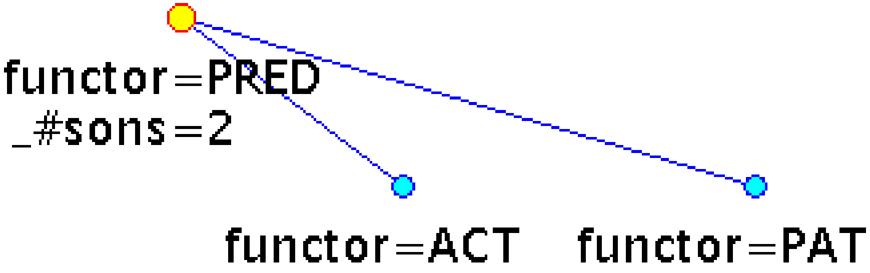
Attributes not present in the corpus,
treated like normal attributes:

- **_transitive** *(transitive edge)*
- **_optional** *(optional node(s))*
- **_#sons** *(number of sons)*
- **_#hsons** *(number of hidden sons)*
- **_#descendants** *(number of nodes in the subtree)*

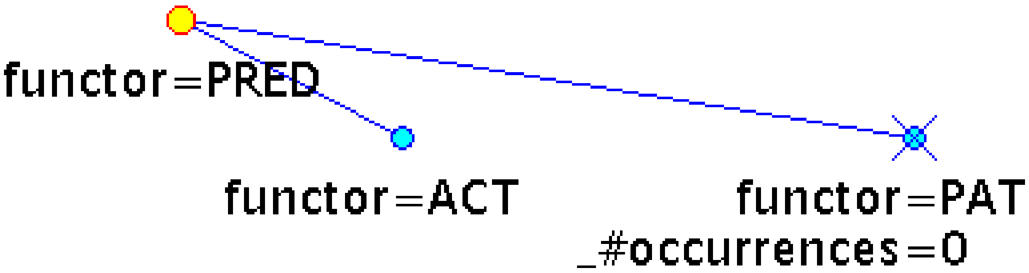
Netgraph Query Language
Meta-Attributes

- **_#lbrothers** *(number of left brothers)*
- **_#rbrothers** *(number of right brothers)*
- **_depth** *(distance from the root)*
- **_#occurrences** *(exact number of a particular type of sons/descendants)*
- ***_name*** *(label of a node for references)*
- ***_sentence*** *(linear form of the sentence)*

Using the Query Language
Valency

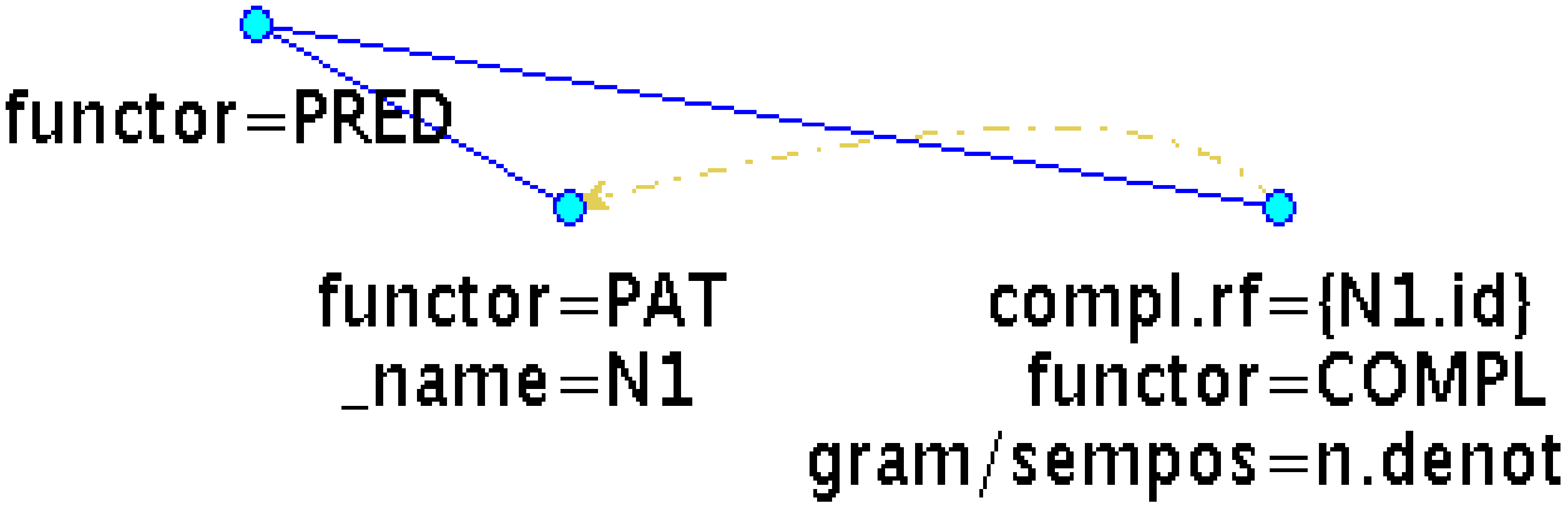


one ACTor, one PATient,
no other sons



at least one son (ACTor),
no PATient

Using the Query Language
Predicative Complement



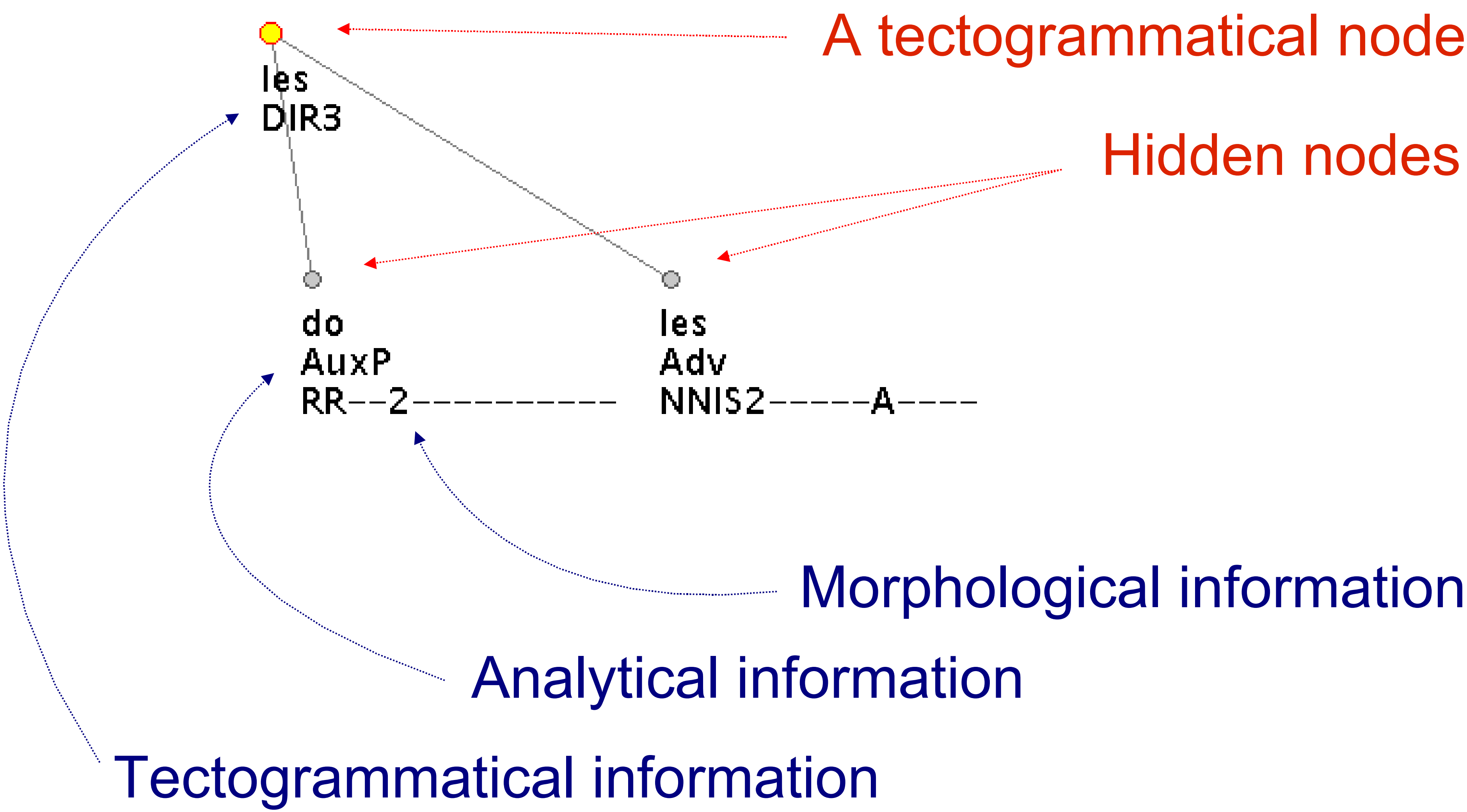
functor=PRED

functor=PAT
_name=N1

compl.rf={N1.id}
functor=COMPL
gram/sempos=n.denot

- a nominal predicative COMPLement with second dependency on a PATient

Netgraph Query Language
Hidden Nodes



A tectogrammatical node

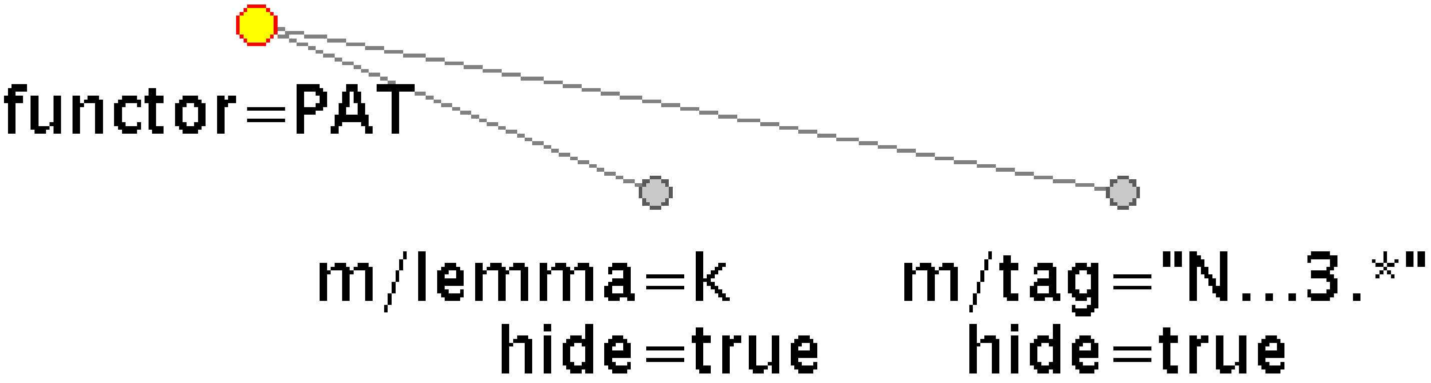
Hidden nodes

Morphological information

Analytical information

Tectogrammatical information

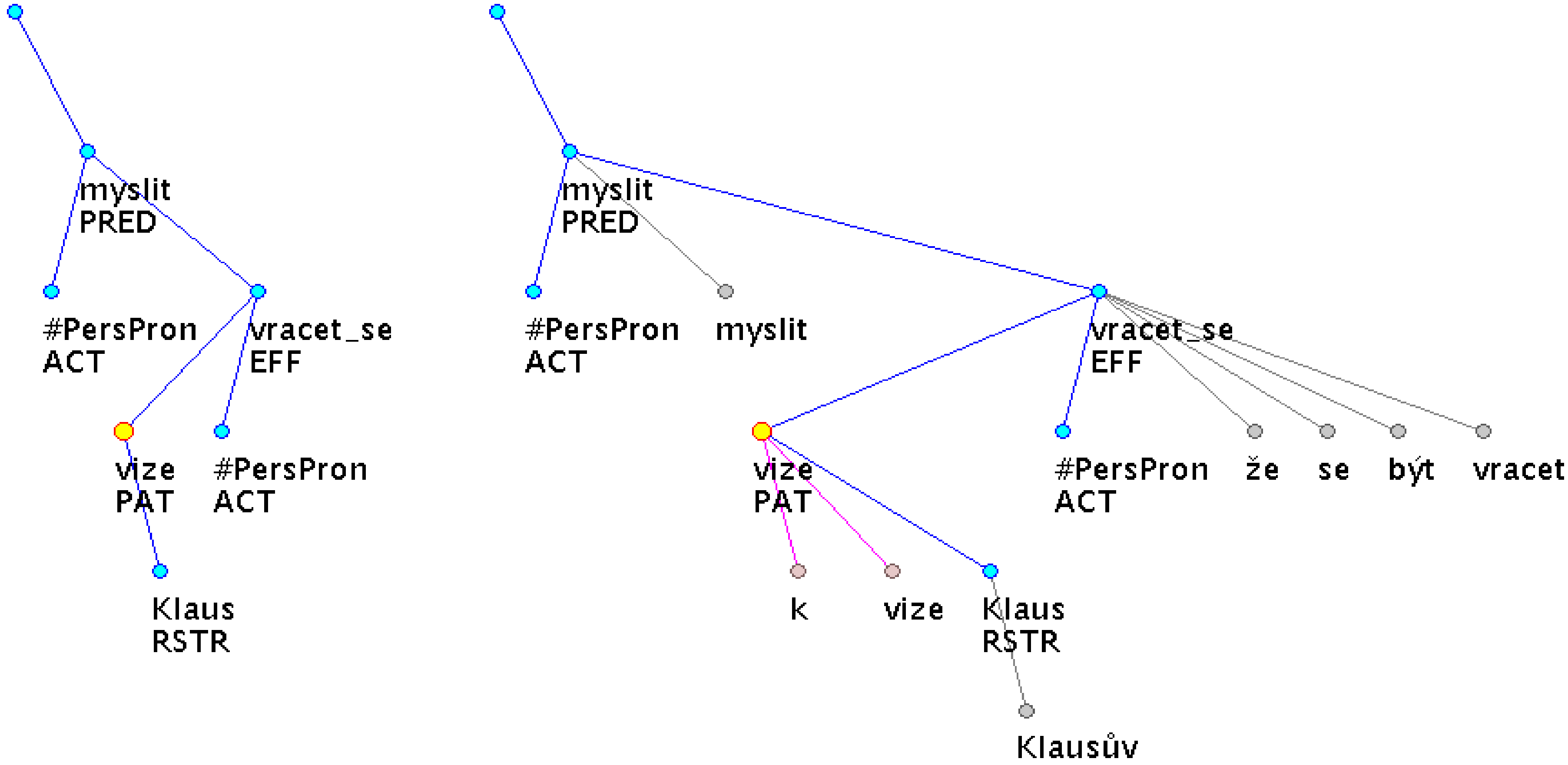
Using the Query Language
Hidden Nodes – A Query



- a PATient expressed with a preposition k and a noun ("N...3.*") in the dative ("N...3.*")

Using the Query Language
Hidden Nodes – A Result Tree

Czech: Myslím, že ke Klausově vizi se budeme vracet.
English: I think that we will get back to Klaus's vision.



Comparison to Other Tools

A detailed comparison to the query languages of the three following search tools has been done:

- Netgraph > TGrep (Pito 1994)
- Netgraph ~ TGrep2 (Rohde 2005)
- Netgraph ~ TigerSearch (Brants et al. 2002)

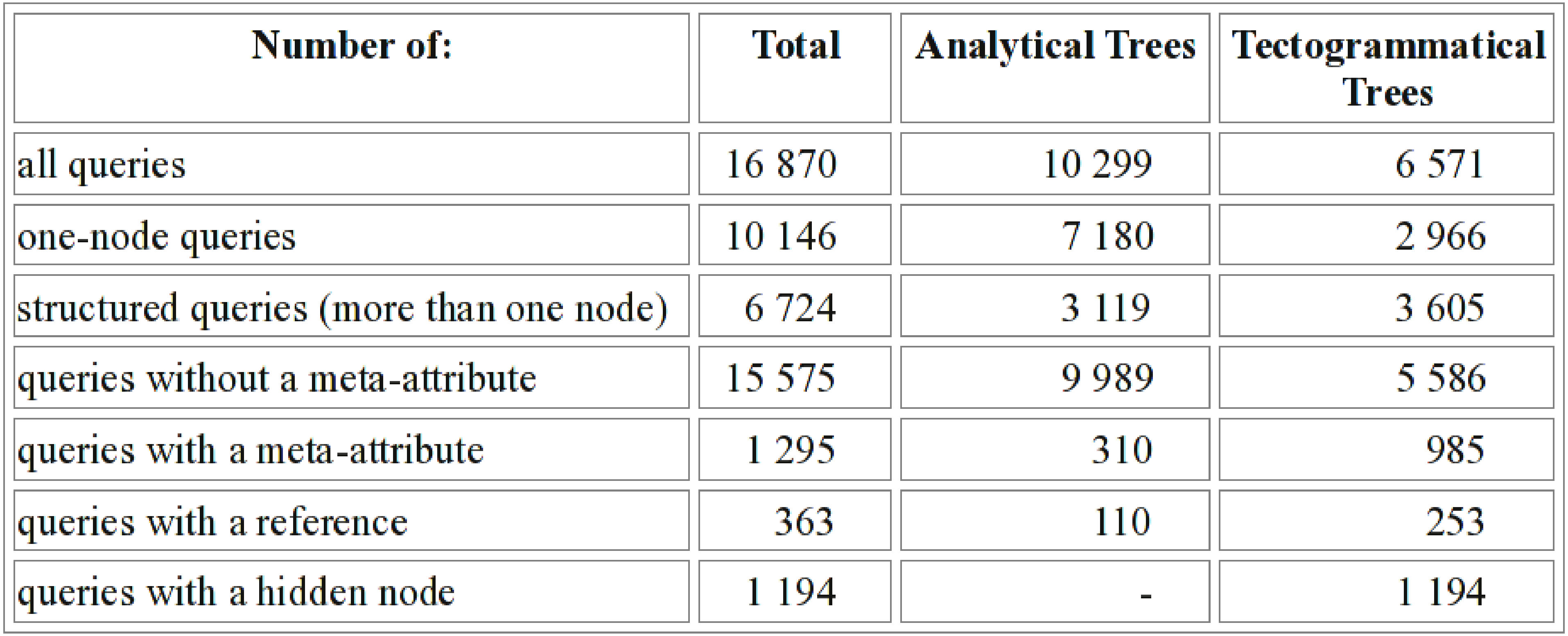
Netgraph As a Tool

Main extensions to the tool since version 1.0:

- Graphical creation of the query
- Chained queries
- Inverted matching
- Displaying context trees
- Removing trees from the result

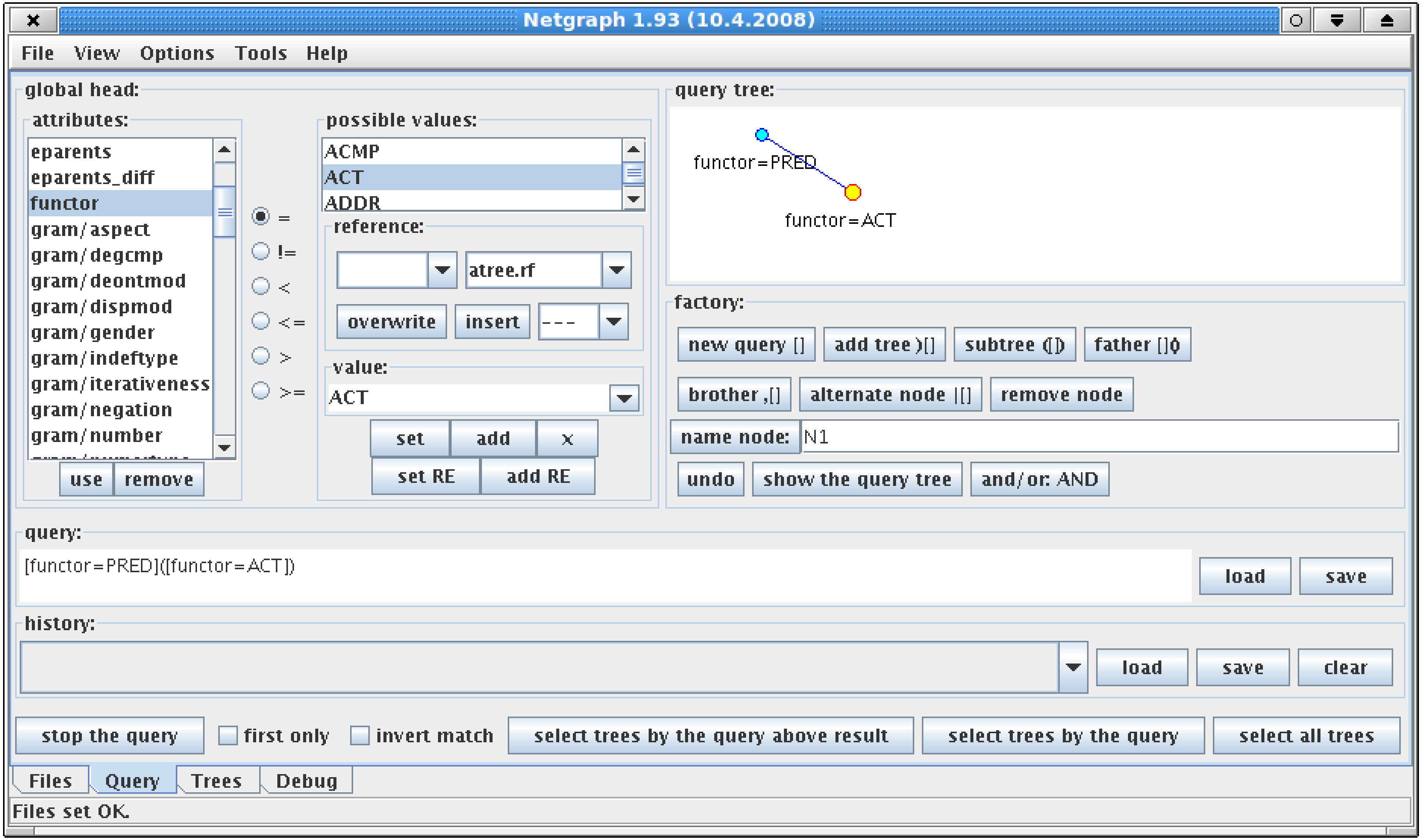
Usage of the Query Language

Analytical trees: October 2002 – March 2008
Tectogrammatical trees: February 2005 – March 2008



Number of:	Total	Analytical Trees	Tectogrammatical Trees
all queries	16 870	10 299	6 571
one-node queries	10 146	7 180	2 966
structured queries (more than one node)	6 724	3 119	3 605
queries without a meta-attribute	15 575	9 989	5 586
queries with a meta-attribute	1 295	310	985
queries with a reference	363	110	253
queries with a hidden node	1 194	-	1 194

Netgraph 1.93



Netgraph 1.93 (10.4.2008)

global head:

attributes: functor

possible values: ACT

● =

reference: atree.rf

value: ACT

query tree:

functor=PRED

functor=ACT

name node: N1

and/or: AND

query:

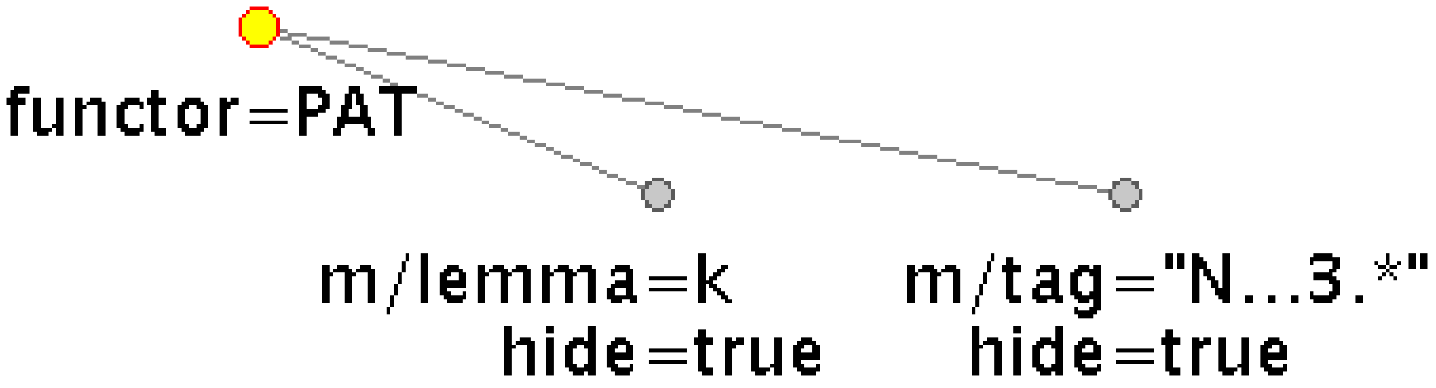
[functor=PRED]([functor=ACT])

history:

☐ first only ☐ invert match

Files set OK.

Using the Query Language
Hidden Nodes – A Query



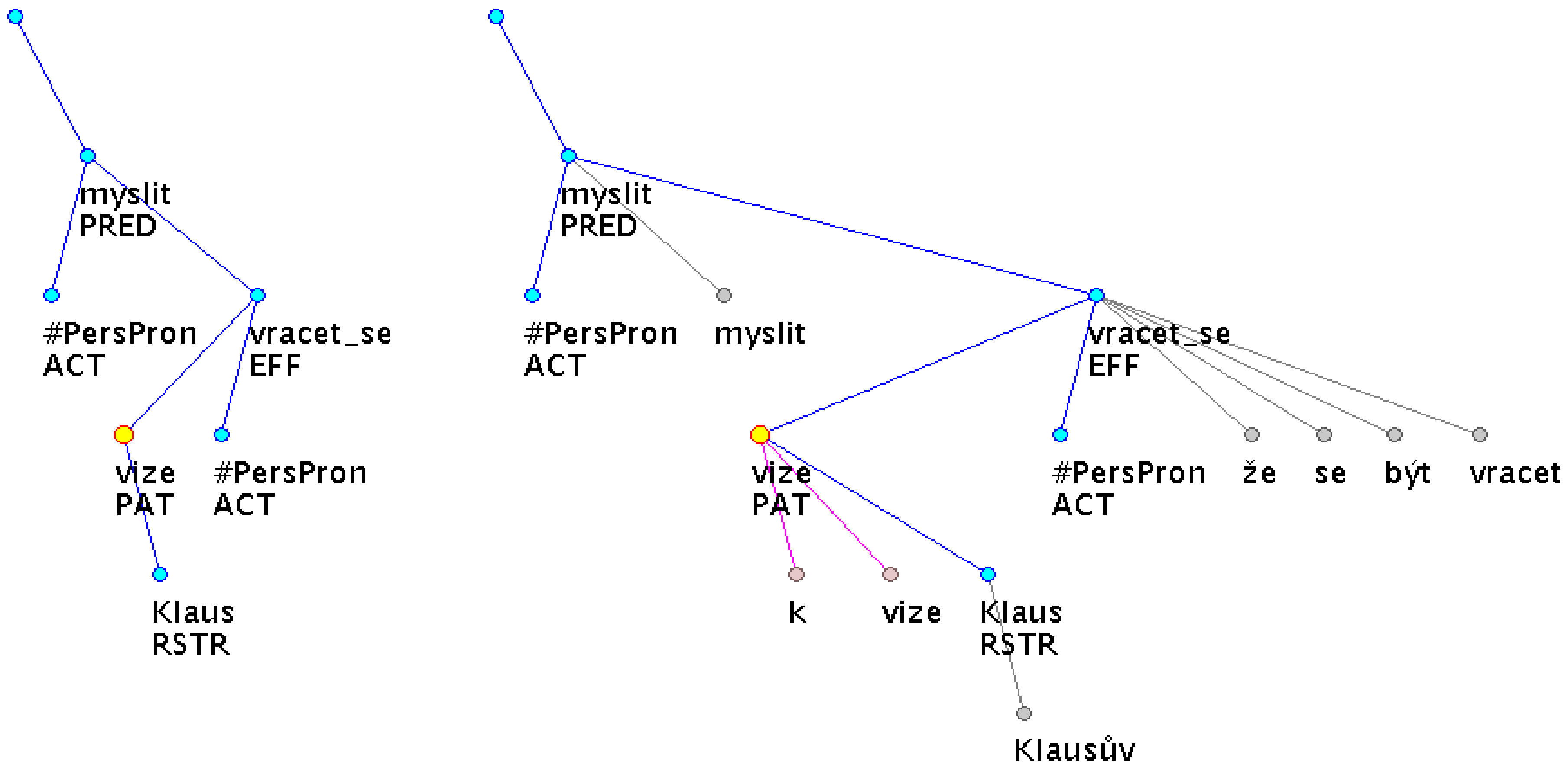
functor=PAT

m/lemma=k
hide=true

m/tag="N...3.*"
hide=true

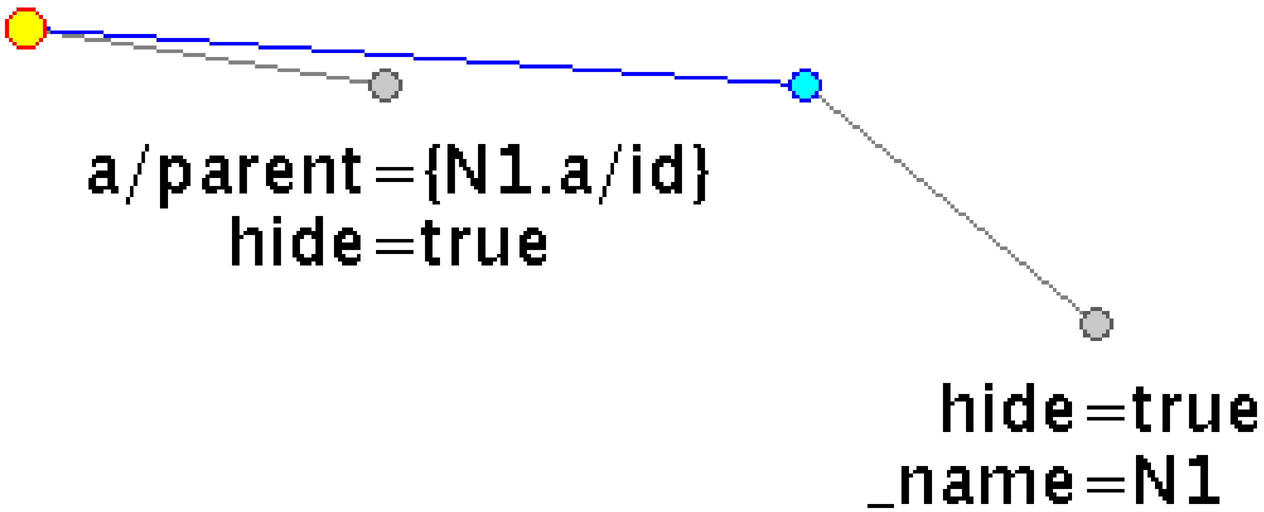
- a PATient expressed with a preposition k and a noun ("N...3.*") in the dative ("N...3.*")

Using the Query Language
Hidden Nodes – A Result Tree



Using the Query Language

Hidden Nodes – A Query



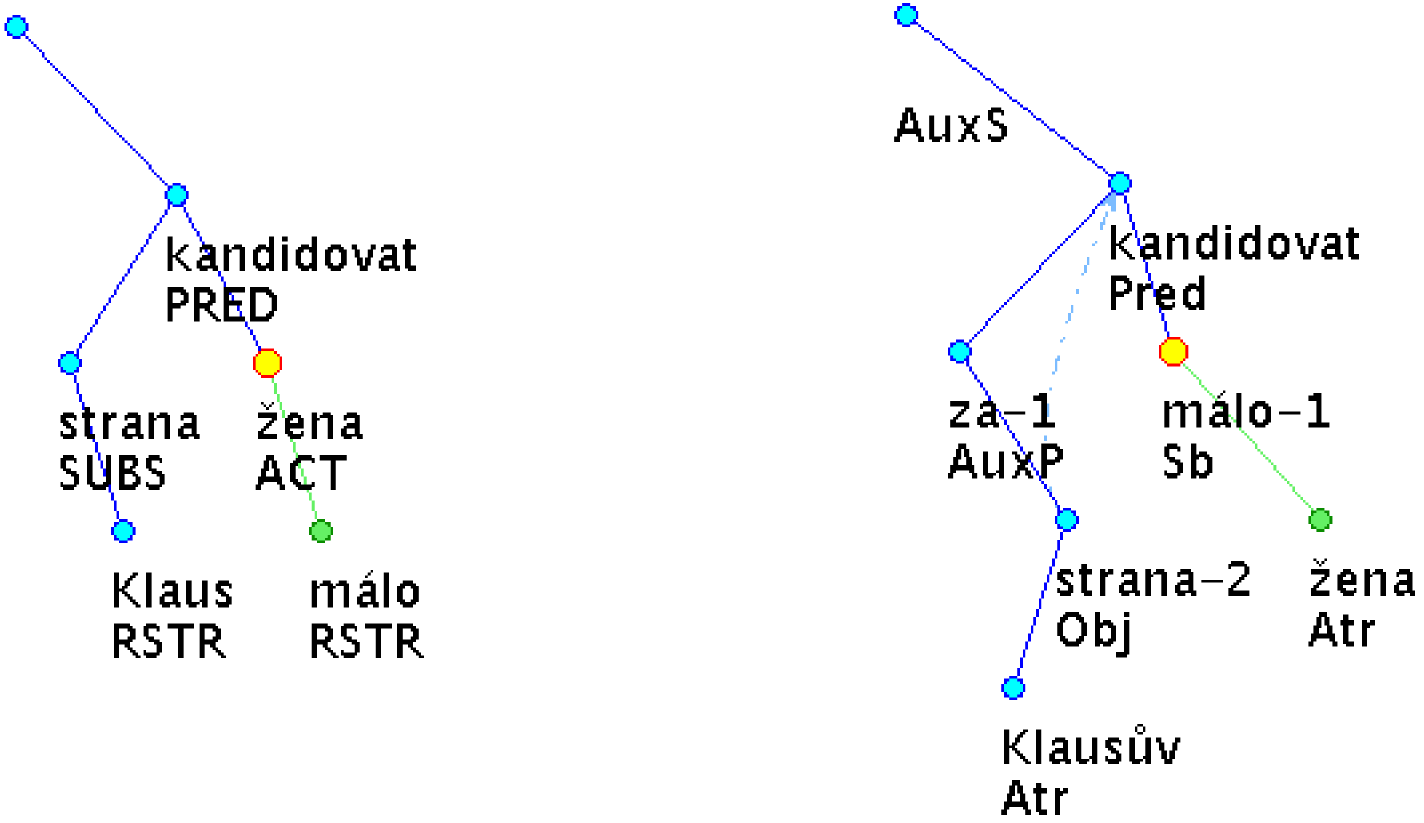
a/parent={N1.a/id}
hide=true

hide=true
_name=N1

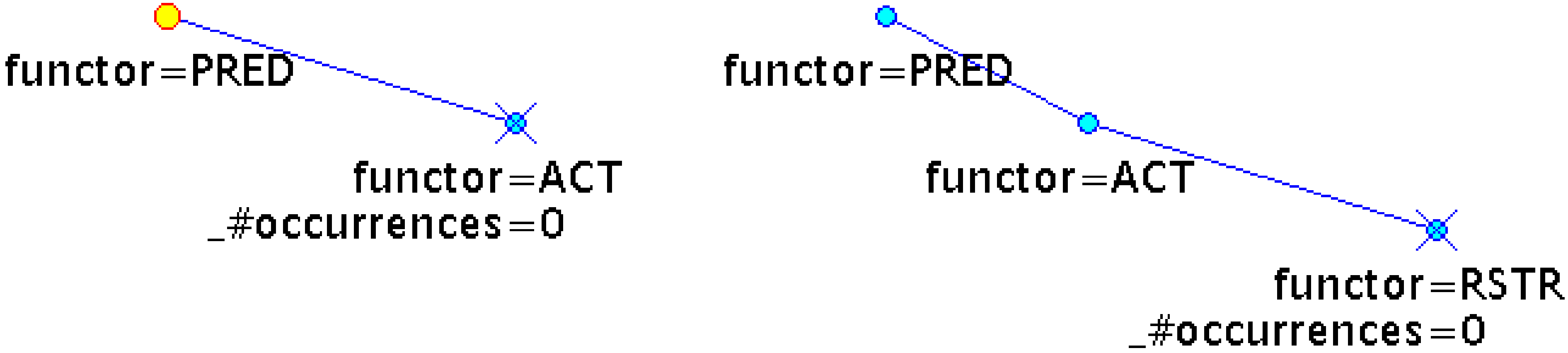
- an opposite dependency on the different layers

Using the Query Language

Hidden Nodes – A Result Tree

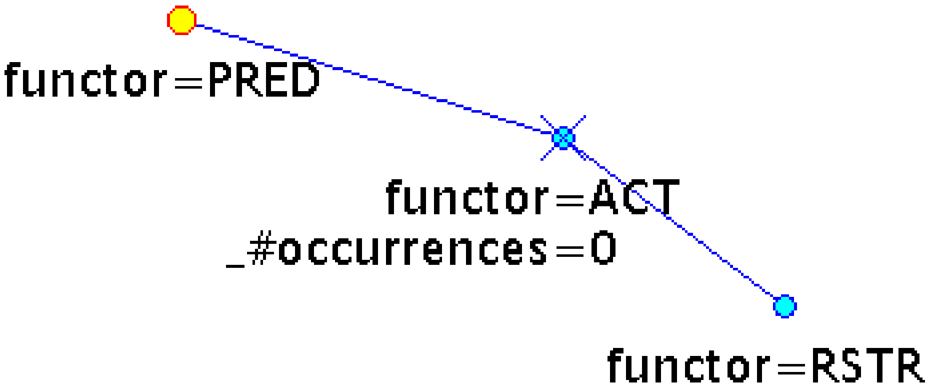


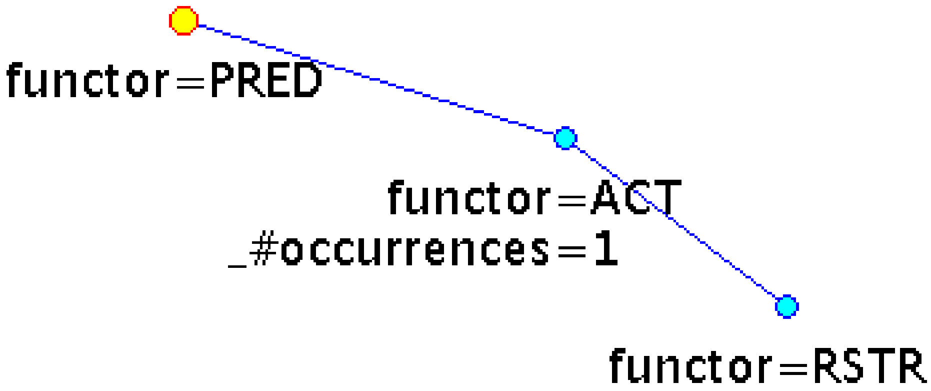
Improvements to the Query Language
Structured Negation Today



- a PREDicate that does not govern
an ACTor that governs a RSTR
(a multi-tree query with relation AND)

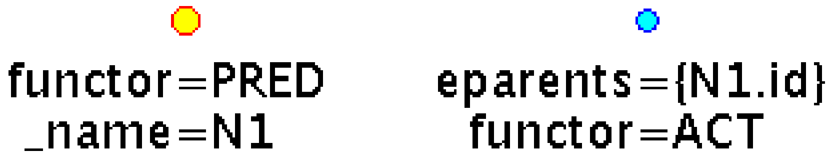
Improvements to the Query Language
Structured Negation – a Possible Future





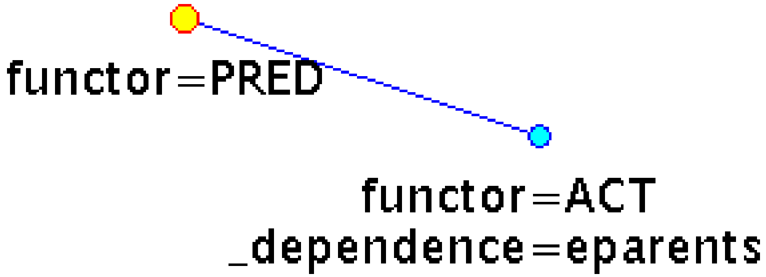
- unclear meaning of the second query

Improvements to the Query Language
Changing the Meaning of the Dependency



functor=PRED
_name=N1

eparents={N1.id}
functor=ACT

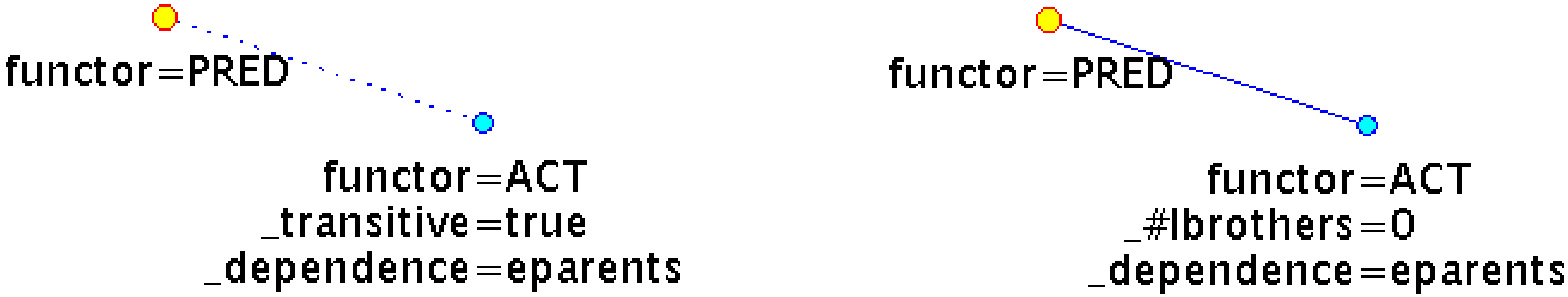


functor=PRED

functor=ACT
_dependence=eparents

- a new meta-attribute _dependence might change the meaning of the dependency

Improvements to the Query Language
Changing the Meaning of the Dependency



- combinations of meta-attributes have to be carefully thought over