

Gramatické formalismy pro ZPJ II

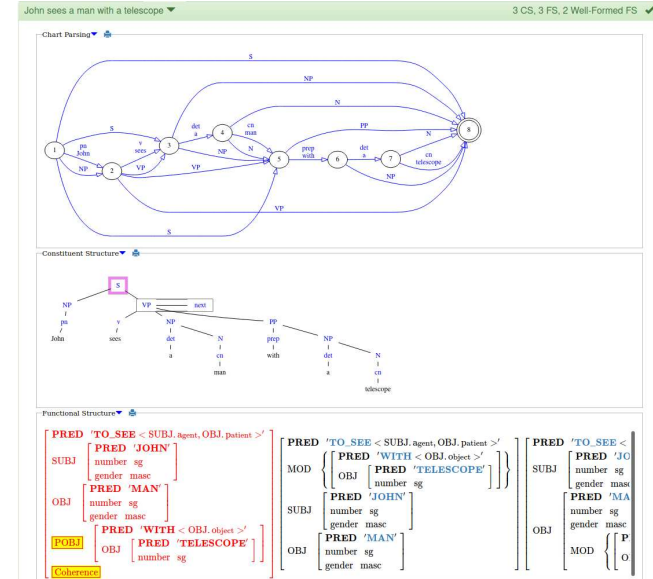
Aleš Horák

E-mail: hales@fi.muni.cz
http://nlp.fi.muni.cz/poc_lingv/

Obsah:

- ▶ Lexikální funkční gramatiky LFG
- ▶ HPSG – Head-driven Phrase Structure Grammar
- ▶ SET – pravděpodobnostní závislostní gramatika
- ▶ Metagramatika systému synt

Lexikální funkční gramatiky LFG

XLFG project – <http://xlfg.labri.fr>

Úvod do počítačové lingvistiky 7/12

1 / 57

Lexikální funkční gramatiky LFG

Úvod do počítačové lingvistiky 7/12

2 / 57

Lexikální funkční gramatiky LFG

Lexikální funkční gramatiky LFG

XLE web interface – <https://clarino.uib.no/iness/xle-web>

XLE-Web

XLE-Web documentation can be found here.

Grammar: English

Write a sentence (max. 600 characters), ending it with punctuation (. ? or !).
 Please observe orthographic conventions, such as capitalization of proper names.
 Please do not use XLE-Web for automatic parsing of larger amounts of text. Contact iness@sub.no instead.

Write a sentence

Parse sentence Packed representation Suppress CHECK Show discriminant weights
 Morphemes Tokens Generate | Prolog Show XLE messages Suppress complex categories Include non-top F-structures
 GIT update grammar Show unoptimal PREDs only Show discriminants c-structure f-structure

1+1 solutions, 0.017 CPU seconds, 2.924MB max mem, 137 subtrees unified

Discriminants **C-structure** **F-structure**

Selected solutions: 1 of 1

Úvod do počítačové lingvistiky 7/12

3 / 57

Lexikální funkční gramatiky LFG

- ▶ LFG, *Lexical Functional Grammar* – Kaplan a Bresnan, 1982

- ▶ dva typy syntaktických struktur

- vnější, **c-struktura** – viditelná hierarchická organizace slov do frází
- vnitřní, **f-struktura** – abstraktnější struktura gramatických funkcí, které tvoří hierarchii komplexních funkčních struktur

důvod:

- různé přirozené jazyky se významným způsobem odlišují v **organizaci fráze**, v pořadí a způsobech realizace gramatických funkcí
- abstraktnější, **funkcionální** organizace jazyků se odlišuje mnohem méně v mnoha jazycích se např. objevují gramatické funkce *podmětu*, *předmětu* atd.

Úvod do počítačové lingvistiky 7/12

4 / 57

Lexikální funkční gramatiky LFG – pokrač.

- ▶ **L** = vztahy mezi jazykovými formami, např. mezi aktivními a pasivními formami slovesa, jsou zobecněním struktury **lexikonu**, ne transformačními operacemi, derivujícími jednu formu z druhé
- ▶ **F** = **funkcionální teorie** – gramatické vztahy, jako je podmět, předmět atd., jsou základními konstrukty, a nejsou definovány pomocí konfigurace frázové struktury, nebo sémantických pojmů typu Agent a Patient
- ▶ v **LFG** – pro reprezentaci **funkcionální syntaktické informace** je vhodné definovat hierarchickou strukturu jazykových jednotek, avšak *vynucená linearizace* pořádku těchto struktur *není vhodná*

Syntaktické úrovně LFG

- ▶ dvě syntaktické úrovně:
 - **složková struktura** (*c-structure, constituent structure*) – zachycuje frázovou dominanci a prioritu a je reprezentována jako **strom** frázové struktury (CFG strom)
 - **funkcionální struktura** (*f-structure*) – zachycuje syntaktickou strukturu typu predikát-argumenty a je reprezentována *maticí dvojic atribut-hodnota*
nabízí jednotnou reprezentaci syntaktické informace abstrahující od detailů struktury fráze a lineárního pořádku
- f-struktura obsahuje soubor atributů:
 - **příznaky** – čas, rod, číslo, ...
 - **funkce** – PRED, SUBJ, OBJ, jejichž hodnoty mohou být jiné f-struktury
- ▶ vztah mezi c-strukturami (stromy) a odpovídajícími f-strukturami:

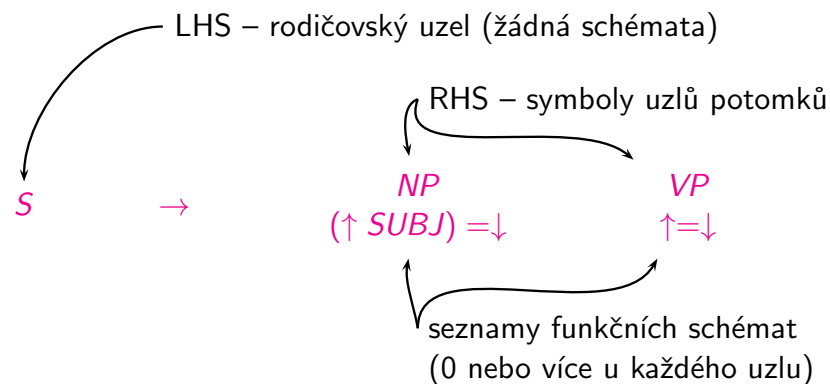
projekce $\phi : \{\text{uzly stromu c-struktury}\} \rightarrow \{\text{f-struktury}\}$

vyjádřená **funkčními schémata**

LFG – c-struktura

LFG pravidla:

- ▶ klasická CF pravidla
- ▶ plus **funkční schémata** – výrazy pracující se symboly na pravé straně pravidel (za \rightarrow , RHS)



LFG – pravidla

příklady:

$$S \rightarrow \quad \text{NP} \quad \text{VP}$$

$$(\uparrow \text{SUBJ}) = \downarrow \quad \uparrow = \downarrow$$

$$VP \rightarrow \quad \text{V} \quad (\text{NP})$$

$$\uparrow = \downarrow \quad (\uparrow \text{OBJ}) = \downarrow$$

$$NP \rightarrow (\text{DET}) \quad \text{N}$$

$$\uparrow = \downarrow \quad \uparrow = \downarrow$$

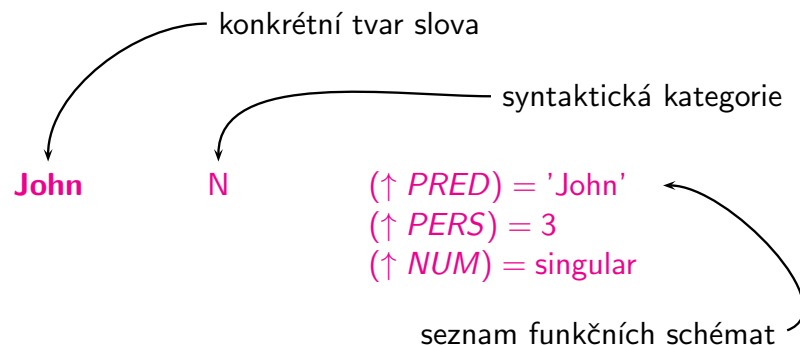
výrazy $(\uparrow \text{SUBJ}) = \downarrow$, $\uparrow = \downarrow$ a $(\uparrow \text{OBJ}) = \downarrow$ jsou *funkční schémata*

LFG – lexikon

lexikon také obsahuje funkční schémata

položka lexikonu:

1. konkrétní tvar slova
2. syntaktickou kategorii
3. seznam funkčních schémat



LFG – lexikon – pokrač.

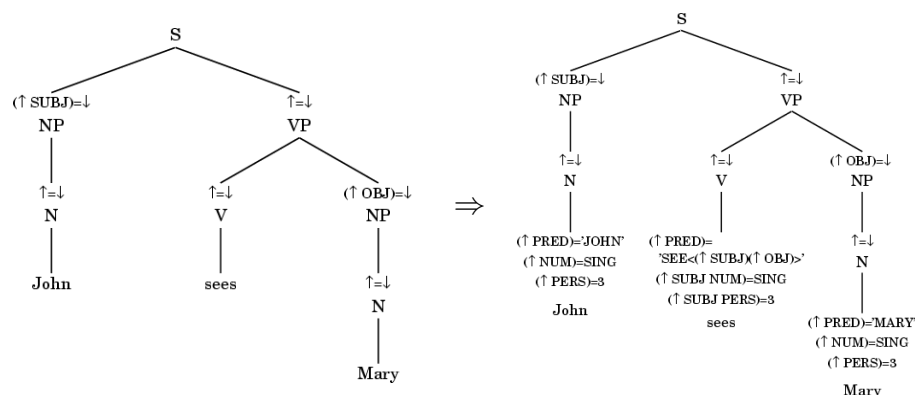
příklady:

John	N	(↑ PRED)	=	'JOHN'
		(↑ NUM)	=	SING
		(↑ PERS)	=	3
sees	V	(↑ PRED)	=	'SEE<(↑SUBJ)(↑OBJ)>'
		(↑ SUBJ NUM)	=	SING
		(↑ SUBJ PERS)	=	3
Mary	N	(↑ PRED)	=	'MARY'
		(↑ NUM)	=	SING
		(↑ PERS)	=	3

LFG – konstrukce c-struktury

informace v c-struktuře:

- ▶ hierarchická struktura větných členů
- ▶ funkční anotace (funkční schémata převedená do stromu) – po jejich interpretaci získáme výslednou f-strukturu



LFG – f-struktura

$$f_n \begin{bmatrix} A & f_m \begin{bmatrix} B & C \\ D & E \end{bmatrix} \\ F & G \\ H & I \end{bmatrix}$$

grafický zápis:

matice atribut-hodnota (*attribute-value matrix*, AVM) – levé sloupce jsou atributy, pravé sloupce hodnoty (symboly, podřazené f-struktury nebo sémantické formy)

funkční rovnice a f-struktury:

$$(f_p \text{ ATT}) = \text{VAL}$$

v f-struktuře f_p je řádek, kde

atribut je ATT

a jeho hodnota je VAL

funkční rovnice mohou být **splněny** nebo **nesplněny** (*true/false*)

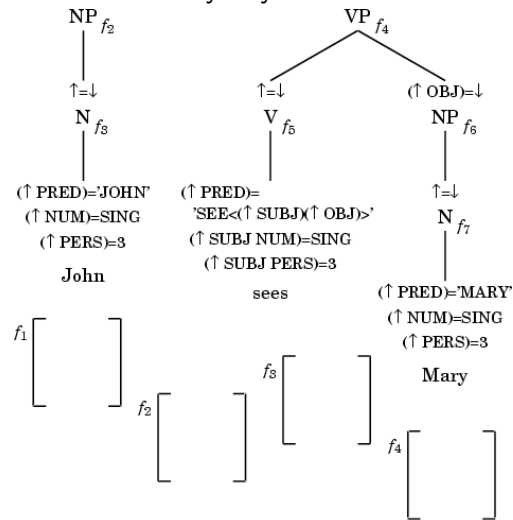
LFG – instanciacie hodnot

Instanciacie hodnot

1. doplňuje hodnoty metaproměnných \uparrow a \downarrow
2. transformuje schémata na **funkční rovnice** – výrazy získané z f-struktur

grafický zápis – f-struktura
v hranatých závorkách []

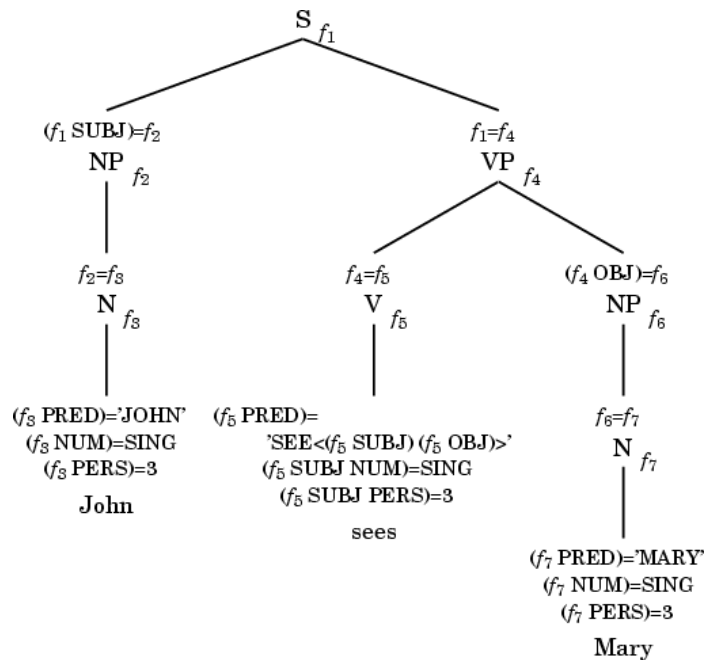
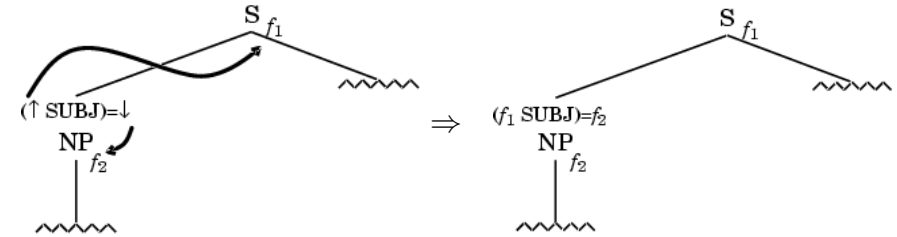
každý uzel c-struktury má
k sobě připojenou matici
f-struktury, které se označují
indexy f_i



LFG – doplnění hodnot metaproměnných

\uparrow a \downarrow (metaproměnné) se odkazují na f-struktury
je potřeba najít správné proměnné f_i na místa šipek

- ▶ \downarrow – metaproměnná **EGO** nebo **SELF** – odkazuje na f-strukturu uzlu nad schématem
- ▶ \uparrow – metaproměnná **MOTHER** – odkazuje na f-strukturu rodičovského uzlu vzhledem k uzlu nad schématem



LFG – funkční popis

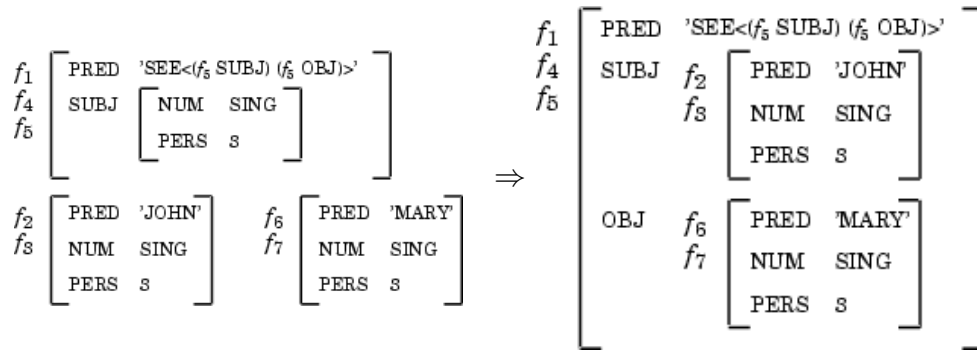
funkční popis = množina všech instanciováných funkčních rovnic stromu
vlastní konstrukce f-struktury pracuje pouze s tímto funkčním popisem
funkční popis předchází větě:

- | | |
|--|---|
| a. $(f_1 \text{ SUBJ}) = f_2$ | i. $(f_5 \text{ SUBJ NUM}) = \text{SING}$ |
| b. $f_3 = f_2$ | j. $(f_5 \text{ SUBJ PERS}) = f_3$ |
| c. $(f_3 \text{ PRED}) = \text{'JOHN'}$ | k. $(f_4 \text{ OBJ}) = f_6$ |
| d. $(f_3 \text{ NUM}) = \text{SING}$ | l. $f_6 = f_7$ |
| e. $(f_3 \text{ PERS}) = f_3$ | m. $(f_7 \text{ PRED}) = \text{'MARY'}$ |
| f. $f_1 = f_4$ | n. $(f_7 \text{ NUM}) = \text{SING}$ |
| g. $f_4 = f_5$ | o. $(f_7 \text{ PERS}) = f_3$ |
| h. $(f_5 \text{ PRED}) = \text{'SEE}<(f_5 \text{ SUBJ})(f_5 \text{ OBJ})>$ | |

LFG – konstrukce f-struktury

f-**struktura** se tvoří z **funkčního popisu** tak, aby všechny funkční rovnice byly **splněny**

výsledná f-**struktura** musí být **minimální** taková f-**struktura**

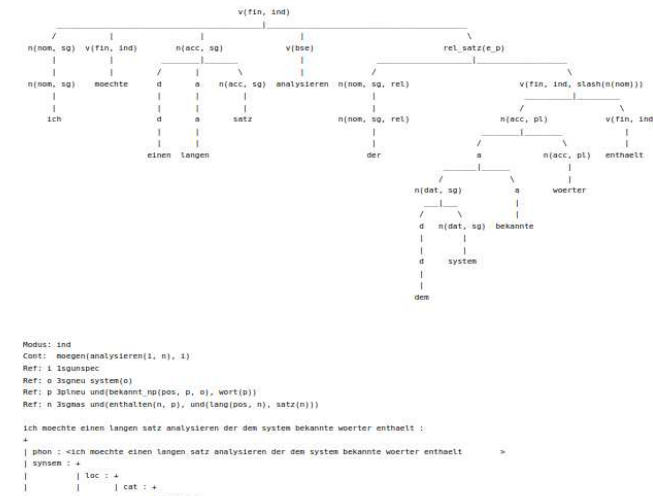


HPSG – Head-driven Phrase Structure Grammar

Das Babel-System: HPSG-Interaktiv

<https://hpsg.hu-berlin.de/~stefan/Babel/>

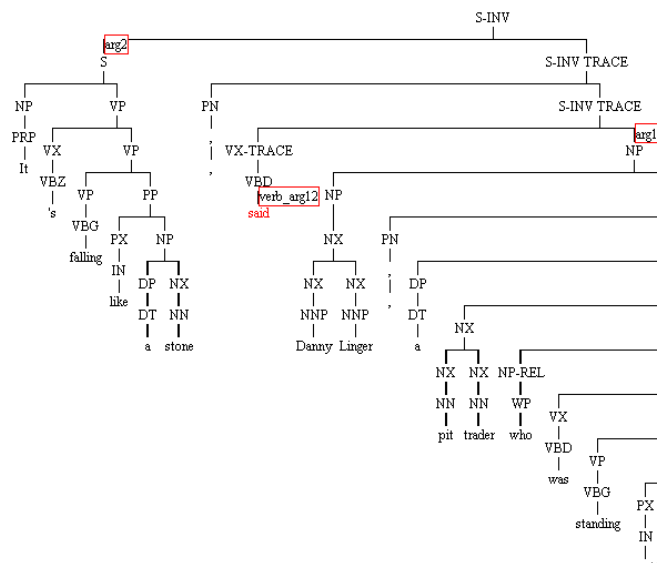
Babel interaktiv: „Ich moechte einen langen Satz analysieren, der dem System bekannte Woerter enthaelt.“
 Analyse für den Satz „Ich moechte einen langen Satz analysieren, der dem System bekannte Woerter enthaelt.“
 Wenn sie nur Teilphrasen analysieren wollen, geben Sie statt eines Satzzeichens ein „*“ am Ende ein!



HPSG – Head-driven Phrase Structure Grammar

English Resource Grammar <http://moin.delph-in.net/wiki/ErgTop>

Enju <https://mynlp.is.s.u-tokyo.ac.jp/enju/>



HPSG – Head-driven Phrase Structure Grammar

- ▶ HPSG, **Head-driven Phrase Structure Grammar** – Pollard & Sag, 1994
- ▶ navazuje na Gazdar, **Generalized Phrase Structure Grammar**, 1985
- ▶ **lexikalizovaná** teorie generativní gramatiky přirozeného jazyka
- ▶ **neterminály** CFG jsou nahrazeny **příznakovými strukturami**
- ▶ založená na **omezeních** (constraints)
- ▶ modeluje jazyk pomocí **deklarativních omezení** typovaných struktur. Pro vyhodnocení omezení se používá **unifikace** mezi příznakovými strukturami.
- ▶ **příznaky** jsou propojeny pomocí **strukturního sdílení**, tedy předáváním proměnných mezi podstrukturami dané struktury
- ▶ HPSG je **nederivační**, na rozdíl od jiných formalismů, kde jsou různé úrovně syntaktické struktury sekvenčně odvozovány pomocí transformačních operací

HPSG – Head-driven Phrase Structure Grammar – pokrač.

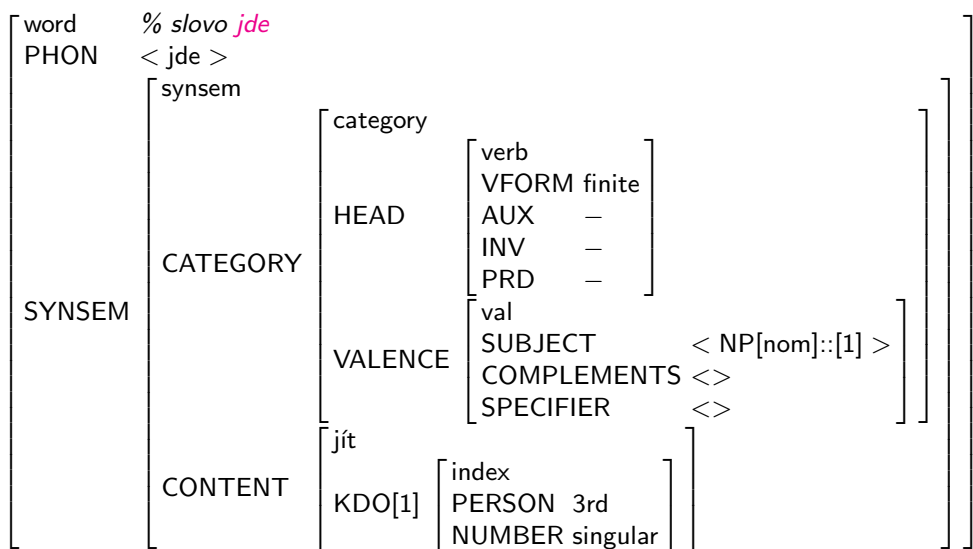
- ▶ gramatika je v HPSG modelována pomocí **uspořádaných příznakových struktur**, které korespondují s typy výrazů přirozeného jazyka a jejich částmi
- ▶ cílem teorie je detailní specifikace, které příznakové struktury jsou **přípustné**
- ▶ příznakové struktury definují **omezení**
hodnoty příznaků mohou být jednoho ze čtyř typů
 - atomy
 - příznakové struktury
 - množiny příznakových struktur ($\{...\}$)
 - nebo seznamy příznakových struktur ($\langle...\rangle$)

Lexikální hlava

- ▶ **slova** (lexikální položky) obsahují **hodně informací** – podle psycholingvistiky se podobá *zpracování v lidském mozku*
- ▶ **lexikální hlava** – základní prvek frázové struktury HPSG
lexikální hlava = jedno slovo, jehož položka specifikuje informace, které určují základní gramatické **vlastnosti fráze**, kterou hlava zastupuje
gramatické vlastnosti zahrnují:
 - morfologické informace (part-of-speech, POS)
N zastupuje NP, VP zastupuje S, V zastupuje VP
 - relace závislosti (např. valenční rámec slovesa)
- ▶ lexikální hlava obsahuje také klíčové **sémantické informace**, které sdílí se zastupovanou frází

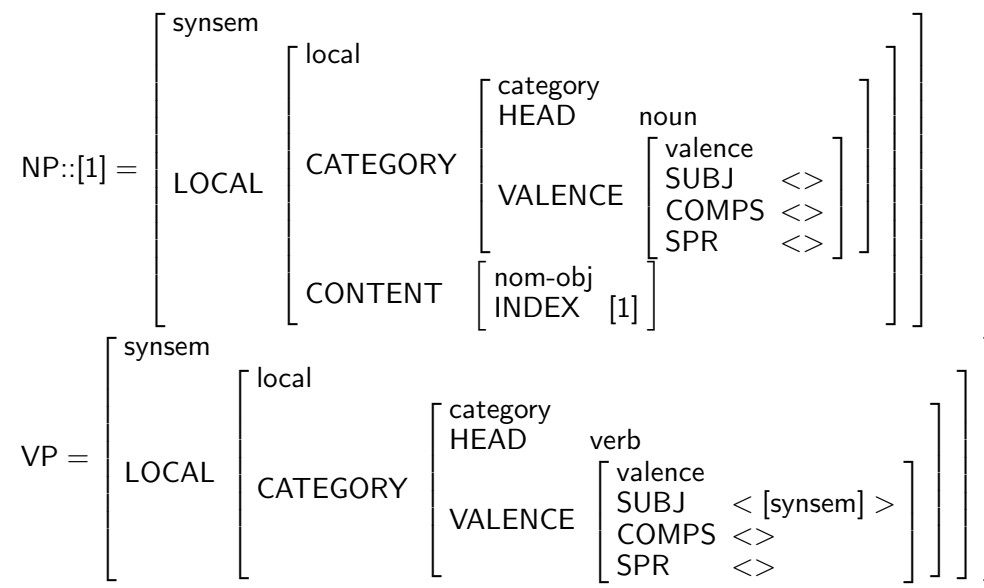
HPSG struktury

HPSG struktury jsou **typované příznakové struktury**
zapisují se pomocí AVM – **příznaky** velkými písmeny, **typy** malými



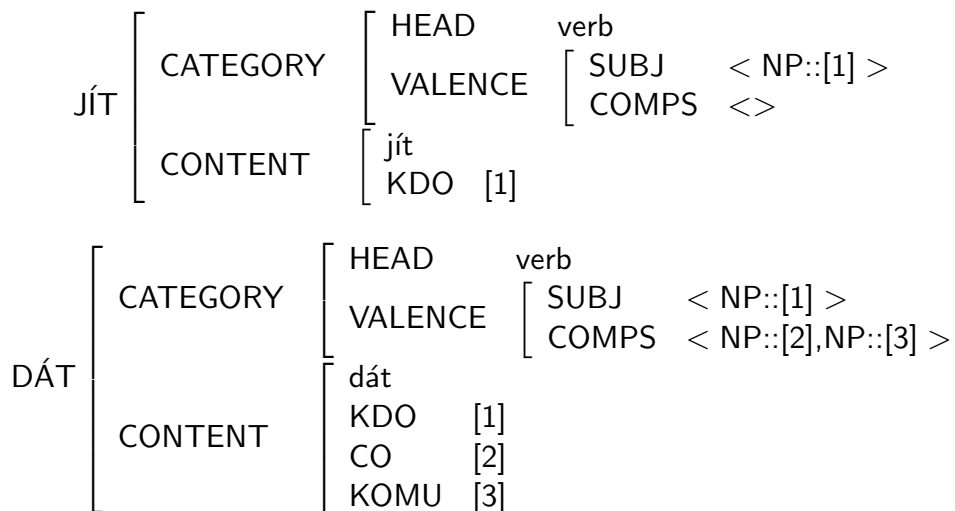
Syntaktické kategorie

symboly **syntaktických kategorií** – zkratky určitých příznakových popisů:



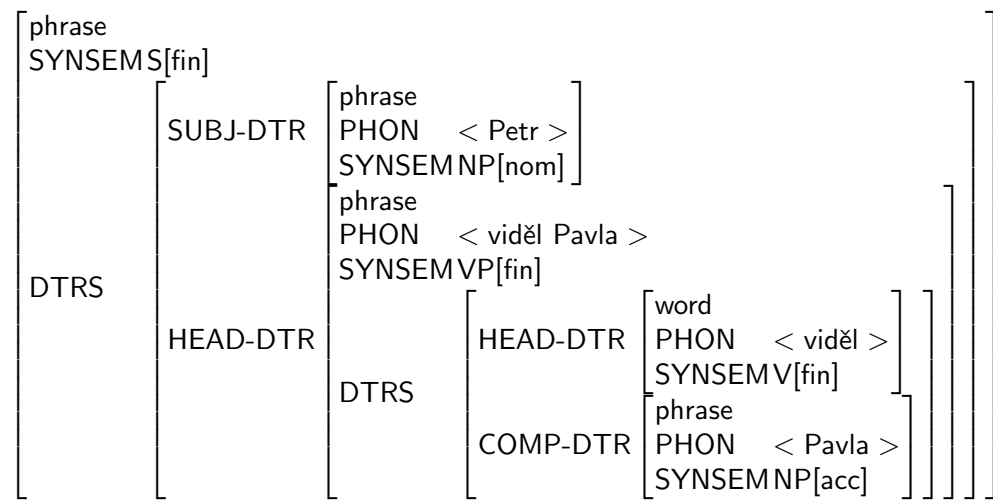
Lexikální položky

velké množství akcí je v **lexikonu**:



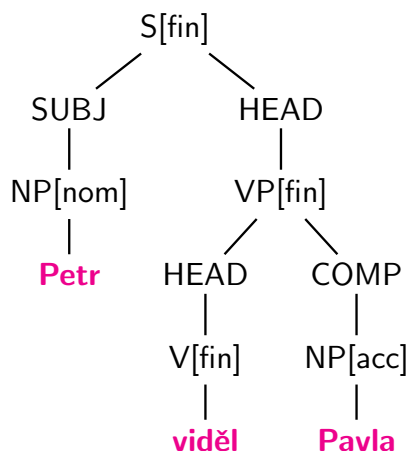
Fráze

reprezentace **frází** – v HPSG obdoba reprezentace **slov**
 navíc příznak **DAUGHTERS** – struktura členů fráze



Fráze – pokrač.

pro snazší čtení popisů frází používáme **stromový zápis**:



ve skutečnosti se ovšem jedná o **příznakovou strukturu**, ne strom!

Dobře utvořené příznakové struktury

dobře utvořené příznakové struktury musí splňovat **omezení daná gramatikou**

příznaková struktura je **dobře utvořená** ⇔:

- ▶ každý uzel splňuje **omezení geometrie příznaku**
- ▶ každá uzel vstupního slova splňuje **omezení některé lexikální položky**
- ▶ každý frázový uzel splňuje **frázová omezení** – *omezení přímé dominance* (immediate dominance, viz dále), *omezení hlavových příznaků* (head feature), *valenční omezení*, ...

omezení geometrie příznaku specifikují:

- ▶ s jakými **typy** se pracuje
- ▶ jaká je použitá **typová hierarchie** – který typ je podtypem jiného typu
- ▶ pro každý typ – jaké příznaky přísluší tomuto typu
- ▶ pro každý typ a každý příznak – jakých typů mohou být hodnoty tohoto příznaku

HPSG – deklarace typu

pro popis omezení geometrie příznaku se používají **typové deklarace**:

```
category: [HEAD: head, VALENCE: valence]

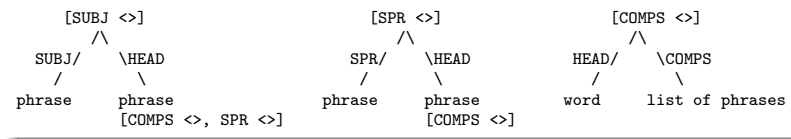
head # příznaková struktura složená z příznakových struktur
noun: [CASE: case]
verb: [VFORM: vform, AUX: boolean, INV: boolean]
prep: [PFORM: pform]
...

vform # jednoduchý příznak, forma slovesa – možné hodnoty:
fin # určitý tvar slovesa
inf # neurčitý tvar slovesa – infinitive
...

case # jednoduchý příznak, gramatický pád
nom # 1. pád, nominativ
acc # 4. pád, akuzativ
...
```

HPSG – dobře utvořená slova a fráze

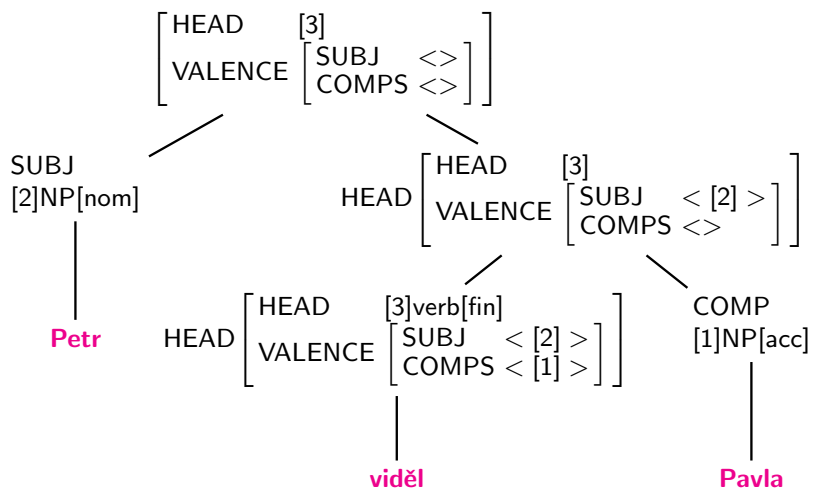
- ▶ každé vstupní **slovo** musí splňovat některou **lexikální položku**
- ▶ **fráze** musí splňovat **frázová omezení** (constraints):
 - **omezení přímé dominance** – každá fráze musí odpovídat jednomu ze schémat – schéma *head-subject*, schéma *head-specifier*, schéma *head-complement*, ...



- **omezení hlavových příznaků** – pro každou frázi, která má hlavu, musí být hlavové příznaky fráze shodné s hlavovými příznaky potomka, který je hlavou
- **valenční omezení** – pro každý z valenčních příznaků (SUBJECT, COMPLEMENTS, ...) – hodnota příznaku na hlavové frázi musí odpovídat hodnotě na potomku, který je hlavou, minus ty příznaky, které jsou splněny některým z nehlavových potomků

Dobře utvořené příznakové struktury

omezení ve větě 'Petr viděl Pavla.':



Syntaktický analyzátor SET

Syntactic Engineering Tool, autor Vojtěch Kovář

- ▶ důraz na **jednoduchost** v návrhu i v použití
- ▶ některé syntaktické jevy jsou lépe **rozpoznatelné** než jiné
- ▶ nejprve určíme **snadnější vztahy**, dále pokračujeme **složitějšími**

Principy:

- ▶ využití principů **parciální analýzy** pro analýzu úplnou
- ▶ pravidlový systém – množina **vzorků** (patterns)
- ▶ **pattern matching** – vyhledávání vzorků v textu

SET – jazyk pro definici pravidel

Každé **pravidlo** obsahuje dvě části – **šablonu** a **akce**

- ▶ **šablona** určuje, **co** se v textu má hledat
- ▶ **akce** určují, jaké **syntaktické vztahy** mají být vyznačeny
- ▶ a morfologické **shody**
- ▶ **pravděpodobnostní ohodnocení** nalezených vzorků – délka, pravděpodobnost pravidla

Příklady pravidel:

```
prep ... noun      AGREE 0 2 c MARK 2 DEP 0 PROB 500
verb ... comma conj ... verb ... bound      MARK 2 7 <relclause>
```

SET – příklady pravidel

Podmínka pro **jedno slovo**:

```
(lemma world)
(word and|or|so)
(tag k[123] .*c2)
```

Podmínka pro **více slov**:

```
noun ... noun2

$C1 (word and) $C2
MATCH $C1(tag) $C2(tag)
k1 k1
k2 k2
END
```

SET – příklady pravidel

Alias:

```
CLASS vpart (word by|bychom|byste|bych|bys)
CLASS noun (tag k1)
CLASS noun2 (tag k1c2)
```

Akce:

- ▶ **MARK** – vyznačuje závislosti a frázové prvky
- ▶ **DEP** – doplnění MARK, udává závislost
- ▶ **HEAD** – doplnění MARK, udává hlavu frázového prvku
- ▶ **AGREE** – požadavek na shodu (**g/n/c**)
- ▶ **PROB** – udává pravděpodobnostní váhu pravidla

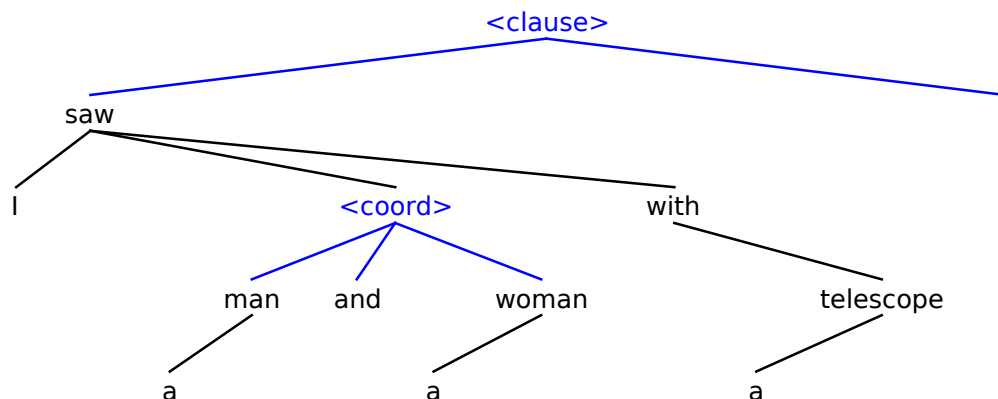
SET – výstup analýzy

hybridní stromy – kombinují **závislostní** a **složkové** prvky

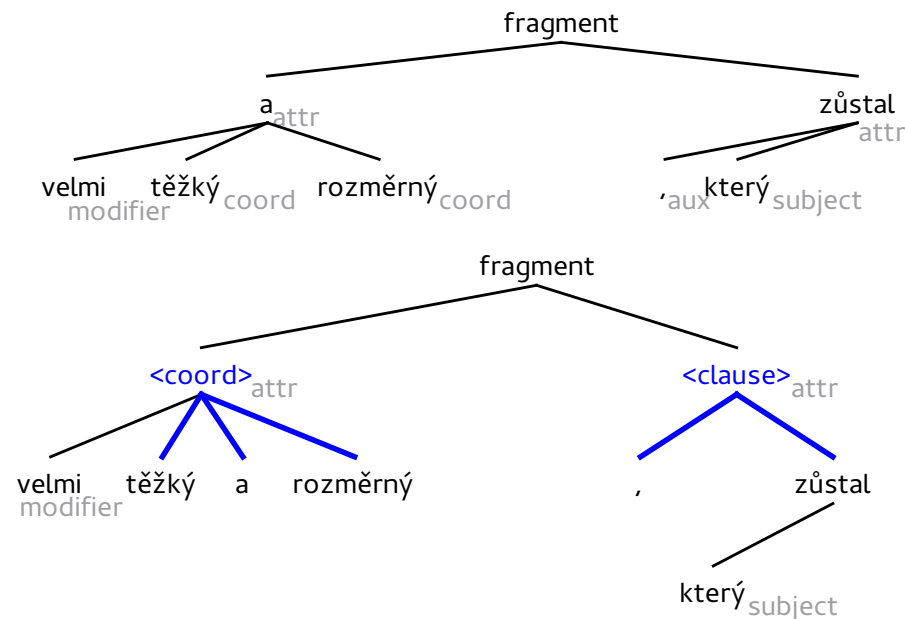
- ▶ **čitelnější** pro člověka
- ▶ rozlišování složkových a závislostních jevů je **výhodou** při analýze
- ▶ možnost **převodu** do čistě závislostního i čistě složkového formátu

Na výstupu analýzy je vždy **jediný strom**, možnost výpisu **všech nalezených vzorků** – zachycení možné víceznačnosti

Hybridní strom – příklad



Hybridní a závislostní strom



SET – implementace

Technické detaily

- ▶ implementace v jazyce Python
- ▶ **objektový model** věty, pravidel a syntaktických vztahů
- ▶ ucelený **soubor pravidel** pro analýzu syntaxe **češtiny**
- ▶ gramatiky pro **angličtinu**, **slovenštinu**
- ▶ specializované gramatiky pro **extrakce informací**, **opravy chyb** (interpunkce), ...
- ▶ 3000 řádků kódu, **70 pravidel**

Funkce:

- ▶ analýza **morfologicky označovaného textu**
- ▶ výstup ve formě různých typů **stromů**, **frází** a **kolokací**
- ▶ reprezentace **víceznačnosti**
- ▶ grafická **vizualizace** výstupu

SET – přesnost a rychlost

Rychlost:

- ▶ asymptoticky $O(R N^2 \log(R N^2))$
- ▶ v praxi 0.14 sekundy na větu

Přesnost závislostního výstupu (vzhledem k PDT, SET v0.3):

Testovací sada	Přesnost – průměr	Přesnost – medián
PDT e-test	76,14 %	78,26 %
BPT2000	83,02 %	87,50 %
PDT50	92,68 %	94,99 %

<http://nlp.fi.muni.cz/projekty/set/>

Metagramatika systému synt

3 formy (meta)gramatiky: [ukázka](#)

▶ metagramatika (G1)

- ▶ pravidla s kombinatorickými konstrukty + globální omezení pořadí
- ▶ akce (= gramatické testy + kontextové akce)
- ▶ česká lingvistická tradice – závislostní struktury, kontrola shody, pravidla pro pořadí slov, ...

▶ generovaná gramatika (G2)

- ▶ bezkontextová pravidla
- ▶ akce

▶ expandovaná gramatika (G3)

- ▶ jen bezkontextová pravidla

The screenshot shows three windows of a grammar editor. Each window contains a list of rules for a specific grammar type (G1, G2, G3). The rules are written in a specific syntax, often including comments in /* */ and using various symbols like \$1, \$2, \$3 for variables. The G1 window shows rules for basic grammatical structures, G2 shows more complex rules with dependencies and propagation, and G3 shows a large set of rules for a more detailed grammar.

[← Zpět](#)

Metagramatika – kombinatorické konstrukty

kombinatorické konstrukty se používají pro generování variant pořadí daným terminálů a neterminálů

hlavní kombinatorické konstrukty:

- ▶ **order()** generuje všechny možné permutace zadaných komponent
- ▶ **first()** argument musí být na prvním místě
- ▶ **rhs()** doplní všechny pravé strany svého argumentu

```
/* budu se ptát */
```

```
clause ==> order(VBU,R,VRI)
```

```
/* který ... */
```

```
relclause ==> first(relprongr) rhs(clause)
```

Metagramatika – globální omezení pořadí

globální omezení pořadí zakazuje některé kombinace pořadí preterminálů

%enclitic – které preterminály jsou brány jako **příklonky**

%order – zajišťuje dodržení precedence zadaných preterminálů

```
/* jsem, bych, se */
```

```
%enclitic = (VB12, VBK, R)
```

```
/* byl — četl, ptal, musel */
```

```
%order VBL = {VL, VRL, VOL}
```

Metagramatika – úrovně pravidel

- ▶ používá se pro **ohodnocení** výstupních stromů pro jejich **třídění**
- ▶ doplněk trénování na **stromových korpusech** (6.000 vět)
- ▶ zadané **lingvistou** – specialistou na vývoj gramatiky
- ▶ **základní úroveň** – 0, **vyšší úrovně** – méně frekventované fenomény
- ▶ pravidla vyšších úrovní mohou být v průběhu analýzy **zapnuté/vyprnuté**

```
3:np -> adj_group
  propagate_case_number_gender($1)
```

Gramatika G2 – kontextové akce

- ▶ gramatické **testy na shody** – pád, rod, číslo
- ▶ **testy na zanoření vedlejších vět** – test_comma
- ▶ akce pro specifikaci **závislostních hran**
- ▶ akce **typové kontroly** logických konstrukcí

```
np -> adj_group np
  rule_schema($@, "lwtx(awtx(#1) and awtx(#2))")
  rule_schema($@, "lwtx([[awt(#1),#2],x])")
```

rule_schema – schéma pro tvorbu logické konstrukce ze subkonstrukcí

projdou jenom kombinace, které **typově vyhovují** danému schématu

Expandovaná gramatika G3

- ▶ překlad testů na shody do CF pravidel
- ▶ v češtině – 7 gramatických pádů, dvě čísla a 4 rody → 56 možných variant pro plnou shodu mezi dvěma prvky

počty pravidel

metagramatika G1	253
gramatika G2	3091
expandovaná gramatika G3	11530

Výstupy syntaktické analýzy

synt nabízí více možností zpracování výsledných struktur:

- ▶ **syntaktické stromy** (varianty: technická/lingvistická, uspořádané/neuspořádané) [▶ ukázka](#)
- ▶ struktura **chart** – komprimovaný *les* všech stromů [▶ ukázka](#)
- ▶ **závislostní graf** – graf všech závislostí vytvořených akcemi [▶ ukázka](#)
- ▶ seznamy **frází** v dané větě, získané přímo ze struktury *chart* [▶ ukázka](#)
- ▶ částečné **zjednoznačnění morfologických značek** na vstupu [▶ ukázka](#)
- ▶ převod na **logické konstrukce TIL** [▶ ukázka](#)

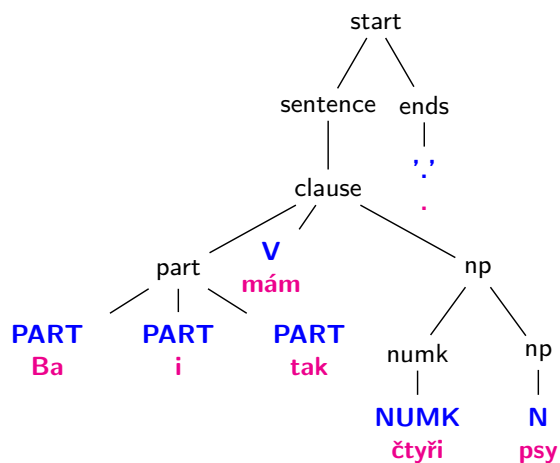
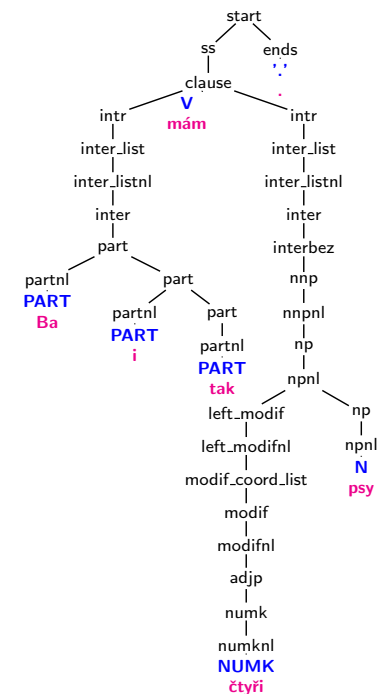
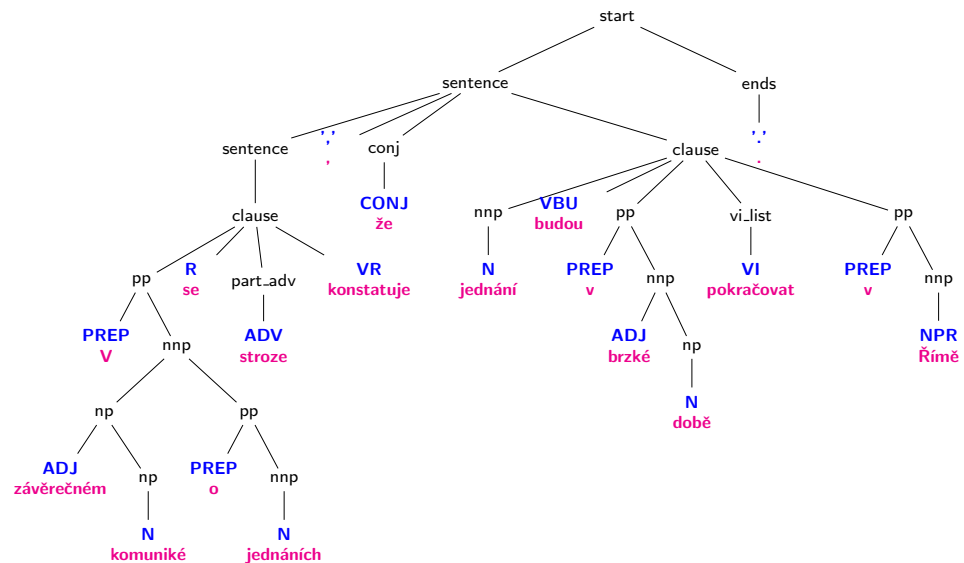
manuál ke **GDW** – Grammar Development Workbench

http://nlp.fi.muni.cz/projekty/grammar_workbench/manual/

DEMO: **wwwsynt** – webové rozhraní k syntu

<http://nlp.fi.muni.cz/projekty/wwwsynt/>

V závěrečném komuniké o jednáních se stroze konstatuje, že jednání budou v brzké době pokračovat v Římě.



ChartView Phraselist3 / 1

File Select Sort View Closed Ranges Help Phraselist3 / 1 REANALYZED

0 ... 16 7346 / 7346

418	0 12	clause -> intr vgca
419	0 14	clause -> intr vgca
420	0 13	clause -> intr vgca
421	0 15	clause -> intr vgca
422	0 10	clause -> intr vgca
423	0 12	clause -> intr vbias
424	0 14	clause -> intr vbias
425	0 13	clause -> intr vbias
426	0 15	clause -> intr vbias
427	0 11	clause -> intr vbias

Select 0 / 649

- left_modifnl -> modif_coord
- modif -> modifnl
- modif -> modifnl conjconj m
- modif_coord_list -> modif
- modifnl -> pre_adj adjp
- modifnl -> adjp
- nnp -> nnpnl conjconj nnp
- nnp -> nnpnl
- nnpnl -> np
- np -> nnpnl conjconj np

422: (5980,505) 0 10 clause -> intr vgca intr vbias intr { IS } intr .
 5980: (5981,7262) 9 10 clause -> intr vgca intr vbias intr { IS } intr .
 505: (-1,47) (-1,506) 0 9 intr -> { inter_list } .

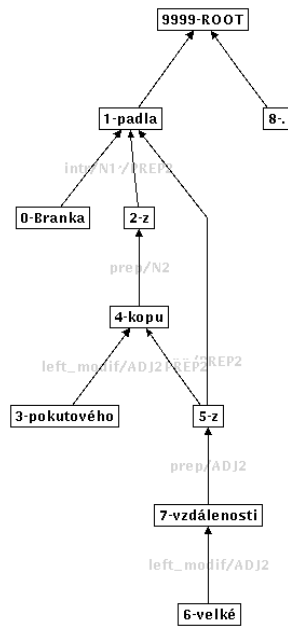
0 1 2 3 4 5 6 7 8 9 10 11 12 13

'Právě' takovému 'řešení' 'a' 'nikoliv' 'utrácení' přebytku 'je' 'konzistentní' 's' 'idejemi's

PART PROND ADJ N PART CONJ PART ADJ N VR ADJ ABBR PREP N INTR

INFO: Closed edges ranges displayed.

Branka padla z pokutového kopu z velké vzdálenosti.



← Zpět

slovo	před	po
Na	k7{c4, c6}	k7c6
krásné	k2eA{gFnPcld1, gFnPc4d1, gFnPc5d1, gFnSc2d1, gFnSc3d1, gFnSc6d1, glnPcld1, glnPc4d1, glnPc5d1, glnScld1wH, glnSc4d1wH, glnSc5d1wH, gMnPc4d1, gMnScld1wH, gMnSc5d1wH, gNnScld1, gNnSc4d1, gNnSc5d1}	k2eAgFnSc6d1
dlouhé	k2eA{gFnPcld1, gFnPc4d1, gFnPc5d1, gFnSc2d1, gFnSc3d1, gFnSc6d1, glnPcld1, glnPc4d1, glnPc5d1, glnScld1wH, glnSc4d1wH, glnSc5d1wH, gMnPc4d1, gMnScld1wH, gMnSc5d1wH, gNnScld1, gNnSc4d1, gNnSc5d1}	k2eAgFnSc6d1
ulici	klgFnSc3, klgFnSc4, klgFnSc6	klgFnSc6
stálo	k5eAalmAgNnSalrD	kSeApNnStMmPal
moderní	k2eA{gFnPcld1, gFnPc4d1, gFnPc5d1, gFnScld1, gFnSc2d1, gFnSc3d1, gFnSc4d1, gFnSc5d1, gFnSc6d1, gFnSc7d1, glnPcld1, glnPc4d1, glnPc5d1, glnScld1, glnSc4d1, glnSc5d1, gMnPcld1, gMnPc4d1, gMnPc5d1, gMnScld1, gMnSc5d1, gNnPcld1, gNnPc4d1, gNnPc5d1, gNnScld1, gNnSc4d1, gNnSc5d1}	k2eAgNnScld1, k2eAgNnSc4d1, k2eAgNnSc5d1
nablýskané	k2eA{gFnPcld1rD, gFnPc4d1rD, gFnPc5d1rD, gFnSc2d1rD, gFnSc3d1rD, gFnSc6d1rD, glnPcld1rD, glnPc4d1rD, glnPc5d1rD, glnScld1wHrD, glnSc4d1wHrD, glnSc5d1wHrD, gMnPc4d1rD, gMnScld1wHrD, gMnSc5d1wHrD, gNnPcld1rD, gNnPc4d1rD, gNnPc5d1rD}	k2eAgNnScld1, k2eAgNnSc4d1, k2eAgNnSc5d1
auto	klgNnSc1, klgNnSc4, klgNnSc5	klgNnSc1, klgNnSc4, klgNnSc5

← Zpět

np: Tyto normy se však odlišují nejen v rámci různých národů a států, ale i v rámci sociálních skupin, a tak považují dřívější pojetí za dosti široké a nedostačující.

[0-2) Tyto normy

[2-3) se

[6-12) v rámci různých národů a států

[15-19) v rámci sociálních skupin

[23-30) dřívější pojetí za dosti široké a nedostačující

vp: Kdybych to byl býval věděl, byl bych sem nechodil.

[0-5): byl býval věděl

[6-10): byl bych nechodil

clause: Muž, který stojí u cesty, vede kolo.

[0-9): Muž , , vede kolo

[2-6): který stojí u cesty

← Zpět

Metagramatika systému synt synt – příklad logické analýzy

System synt – příklad logické analýzy

vyhodnocení rule_schema pro np 'pečené kuře'

4, 6, -npnl -> . left_modif np .: klgNnSc145

agree_case_number_gender_and_propagate OK

rule_schema: 2 nterms, 'lwtx(awtx(#1) and awtx(#2))'

And constrs, Abstr and Exi vars are just gathered

1 (1x1) constructions:

$$\lambda w_2 \lambda t_3 \lambda x_4 ([\text{pečený}_{w_2 t_3}, x_4] \wedge [\text{kuře}_{w_2 t_3}, x_4]) \dots (ol)_{\tau w}$$

And constrs: none added

Exi vars: none added

System synt – příklad logické analýzy – pokrač.

vyhodnocení **verb_rule_schema** pro celou **clause**

verb_rule_schema: 3 groups

no acceptable subject found: supplying an inexplicit one

inexplicit subject: k3xPgMnSc1,k3xPgInSc1: *On...l*

Clause valency list: jíst <v>#1:(1)hA-#2:(2)hPTc1, ...

Verb valency list: jíst <v>#2:hH-#1:hPTc4ti

Matched valency list: jíst <v>#2:(1)hH-#1:(2)hPTc4ti

time span: λt_{12} **dnes** _{t_{12}} ... ($o\tau$)

frequency: **Onc** ... (($o(o\tau)$)) π _{ω}

verbal object: x_{15} ... ($o(o\pi)$)($o\pi$)

present tense clause:

$$\lambda w_{17} \lambda t_{18} (\exists i_{10}) (\exists x_{15}) (\exists i_{16}) ([\text{Does}_{w_{17} t_{18}}, \text{On}, [\text{Imp}_{w_{17}}, x_{15}]] \wedge [\text{večeře}_{w_{17} t_{18}}, i_{10}] \wedge [\text{pečený}_{w_{17} t_{18}}, i_{16}] \wedge [\text{kuře}_{w_{17} t_{18}}, i_{16}] \wedge x_{15} = [\text{jíst}, i_{16}]_{w_{17}} \wedge [[\text{k}_{w_{17} t_{18}}, i_{10}]_{w_{17}}, x_{15}]) \dots \pi$$

clause:

$$\lambda w_{19} \lambda t_{20} [\text{P}_{t_{20}}, [\text{Onc}_{w_{19}}, \lambda w_{17} \lambda t_{18} (\exists i_{10}) (\exists x_{15}) (\exists i_{16}) ([\text{Does}_{w_{17} t_{18}}, \text{On}, [\text{Imp}_{w_{17}}, x_{15}]] \wedge [\text{večeře}_{w_{17} t_{18}}, i_{10}] \wedge [\text{pečený}_{w_{17} t_{18}}, i_{16}] \wedge [\text{kuře}_{w_{17} t_{18}}, i_{16}] \wedge x_{15} = [\text{jíst}, i_{16}]_{w_{17}} \wedge [[\text{k}_{w_{17} t_{18}}, i_{10}]_{w_{17}}, x_{15}])], \lambda t_{12} \text{dnes}_{t_{12}}] \dots \pi$$

[← Zpět](#)