Gramatické formalismy pro ZPJ II

Aleš Horák

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

Obsah:

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 ($\{\ldots\}$)
  - nebo seznamy příznakových struktur ($<\ldots>$)

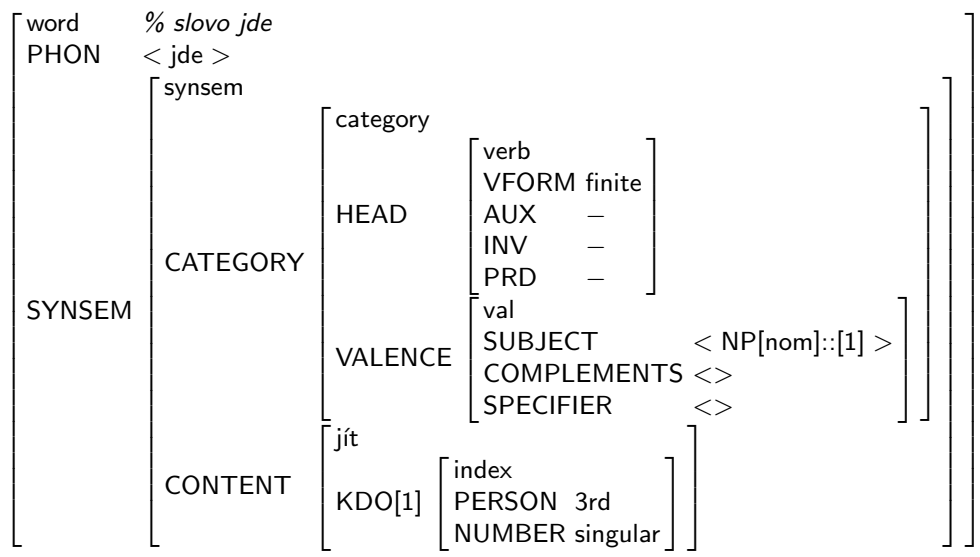
HPSG – lexikální hlava

- **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

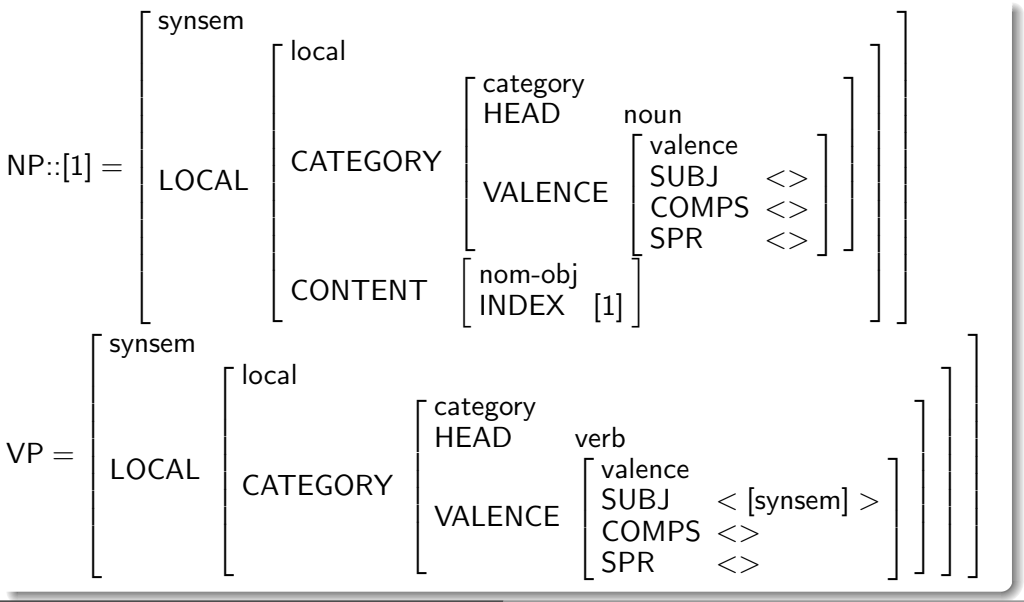
HPSG – struktury

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



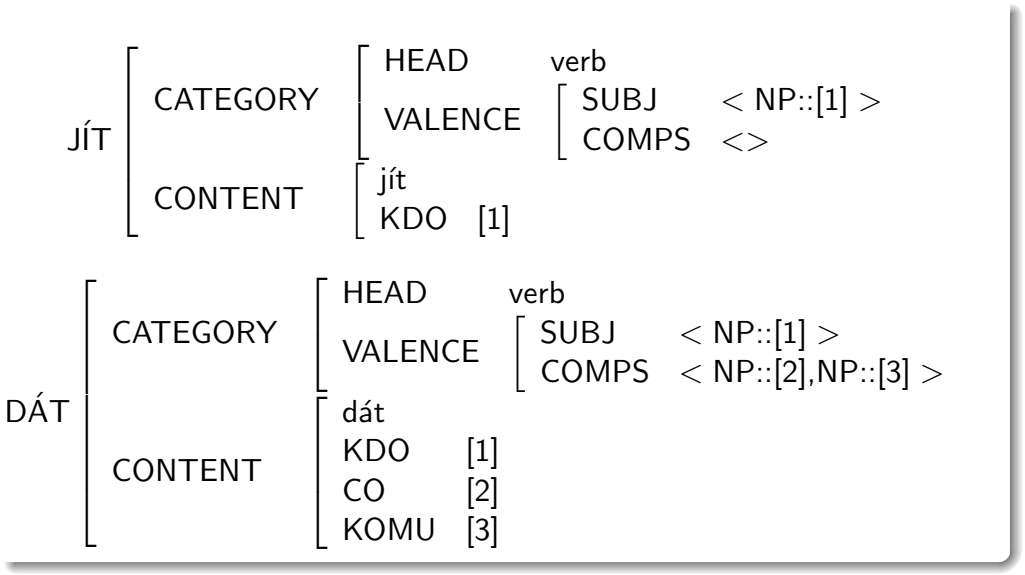
HPSG – syntaktické kategorie

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



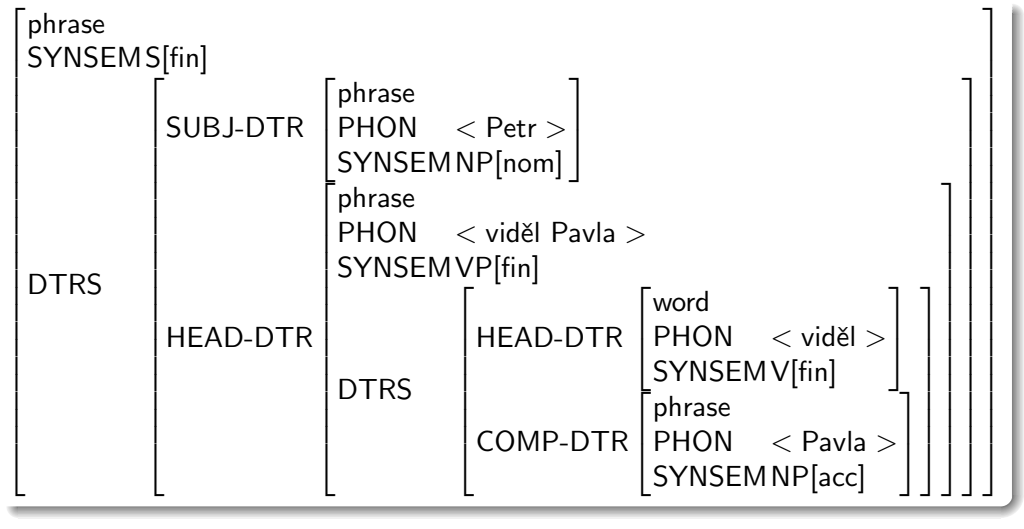
HPSG – lexikální položky

velké množství akcí je v **lexikonu**:



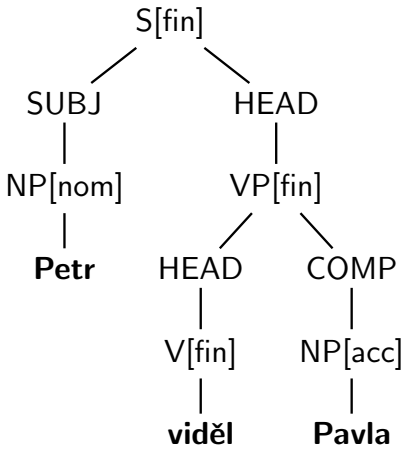
HPSG – fráze

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



HPSG – fráze – pokrač.

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



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

HPSG – 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

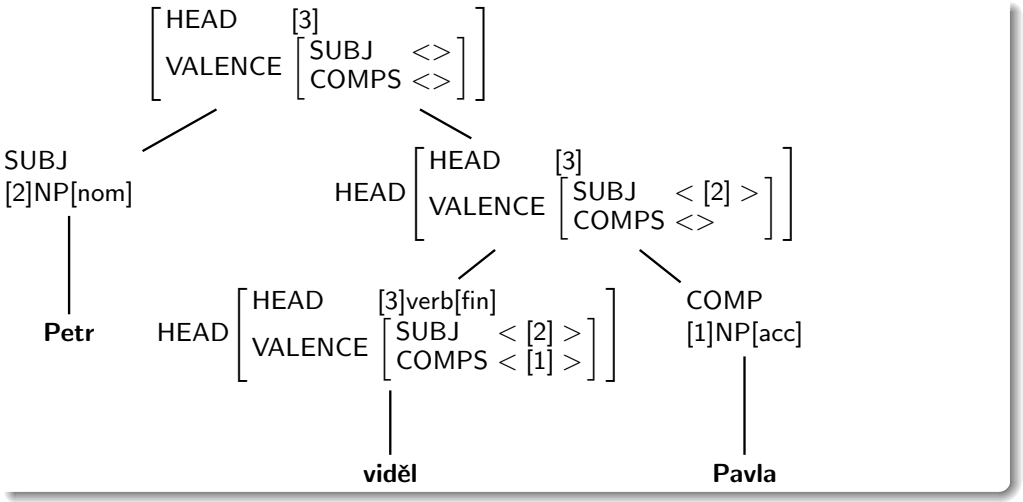
- každé vstupní **slovo** musí splňovat některou **lexikální položku**
- **fráze** musí splňovat **frázová omezení** (constraints):
  - **omezení přímé dominance** – každá fráze musí odpovídat jednomu ze schémat – schéma *head-subject*, schéma *head-specifier*, schéma *head-complement*, ...

```
     [SUBJ <>]              [SPR <>]              [COMPS <>]
        /\                     /\                    /\
   SUBJ/   \HEAD          SPR/   \HEAD          HEAD/   \COMPS
      /     \                /     \               /       \
 phrase     phrase      phrase     phrase        word      list of phrases
            [COMPS <>, SPR <>]    [COMPS <>]
```

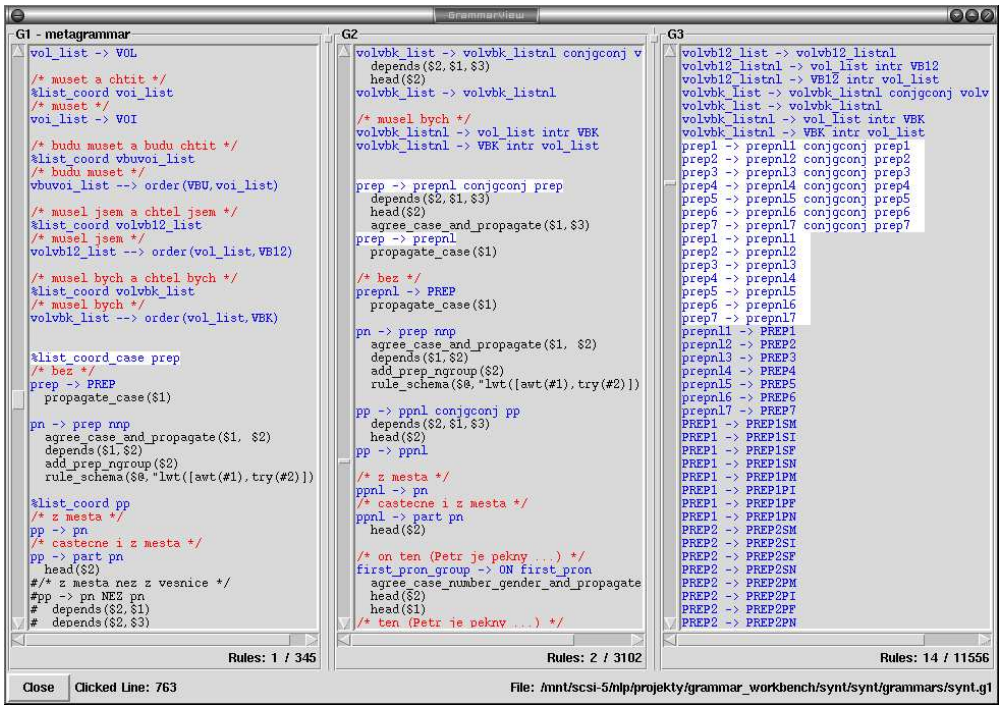
  - **omezení hlavových příznaků** – pro každou frázi, která má hlavu, musí být hlavové příznaky fráze shodné s hlavovými příznaky potomka, který je hlavou
  - **valenční omezení** – pro každý z valenčních příznaků (SUBJECT, COMPLEMENTS, ...) – hodnota příznaku na hlavové frázi musí odpovídat hodnotě na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

HPSG – dobře utvořené příznakové struktury

omezení ve větě ‘Petr viděl Pavla.’:



DEMO: **GG** – HPSG pro němčinu, DFKI Language Technology Lab, Saarbrücken
`http://hpsg.fu-berlin.de/~stefan/Babel/Interaktiv/beispiel.html`



◂ Zpět

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

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 – typy pravidel

- `->` normální CF pravidlo
- `-->` vložit `intersegment` mezi každé dva prvky
- `==>` + kontrola správného pořadí příklonek
- `===>` intersegmenty na začátku a konci RHS, spojky, ...

```
ss -> conj clause
/* budu muset číst */
futmod --> VBU VOI VI
/* byl bych býval */
cpredcondgr ==> VBL VBK VBLL
/* musím se ptát */
clause ===> VO R VRI
```

`clause` pravidla se zadávají pomocí **pravidlových vzorů**

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 – generativní konstrukty

skupina výrazů `%list_*` – produkují nová pravidla pro seznamy
(s oddělovači/bez oddělovačů, s různými testy na shody, ...)

```
/* (nesmim) zapomenout udelat - to forget to do */
%list_nocoord vi_list
vi_list -> VI

%list_coord_case np
%list_coord_case_number_gender left_modif
/* krasny velky pes a mala kocka - beautiful dog and small cat */
np -> left_modif np
```

koncovky `*_case`, `*_number_gender` and `*_case_number_gender` určují typ shody

Metagramatika – pravidlové vzory

pravidla pro slovesné skupiny – cca 40% všech pravidel metagramatiky
pravidlové vzory `%group` – definují časté skupiny konstrukcí v pravidlech

```
%group verbP={
  V:    verb_rule_schema($@,"(#1)")
        groupflag($1,"head"),
  VR R: verb_rule_schema($@,"(#1 #2)")
        groupflag($1,"head"),
}

%template clause ====> order(RHS)

/* ctu/ptam se - I am reading/I am asking */
clause %> group(verbP) vi_list
  verb_rule_schema($@,"#2")
  depends(getgroupflag($1,"head"), $2)
```

Metagramatika – pravidlové vzory – pokrač.

- předchozí příklad – skupina `verbP` = dvě skupiny preterminálů (`V` a `VR R`) s příslušnými akcemi
- při použití v `clause` vytvoří postupně dvě různé pravé strany
- `(get)groupflag` – odkaz na prvek uvnitř `%group`
- **vzor celého pravidla** – speciální pravidlová šipka `%>`
  `%template` definuje vzor každého pravidla s `%>`

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 fenomeny
- pravidla vyšších úrovní mohou být v průběhu analýzy **zapnuté**/**vypnuté**

```
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

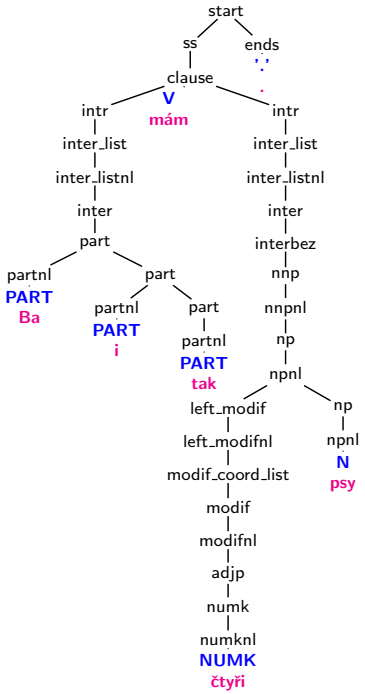
synt nabízí více možností zpracování výsledných struktur:

- **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

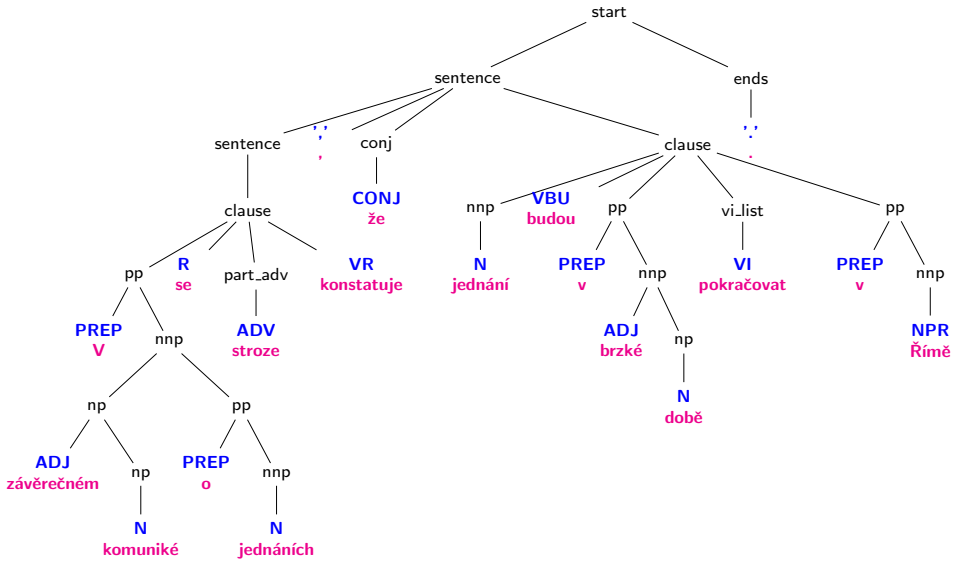
manuál ke **GDW** – Grammar Development Workbench
`http://nlp.fi.muni.cz/projekty/grammar_workbench/manual/`

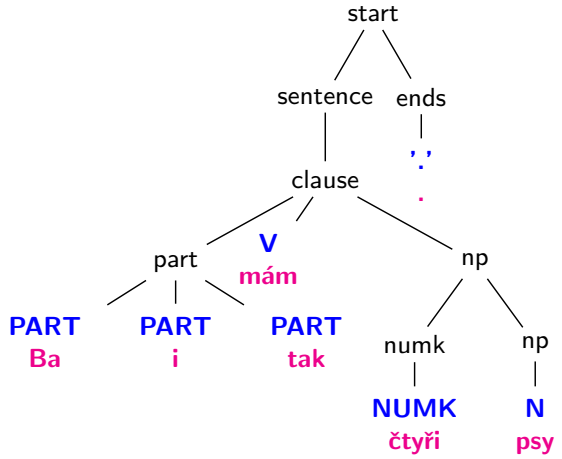
DEMO: **wwwsynt** – webové rozhraní k synt u
`http://nlp.fi.muni.cz/projekty/wwwsynt/`

▸ přeskočit příklady

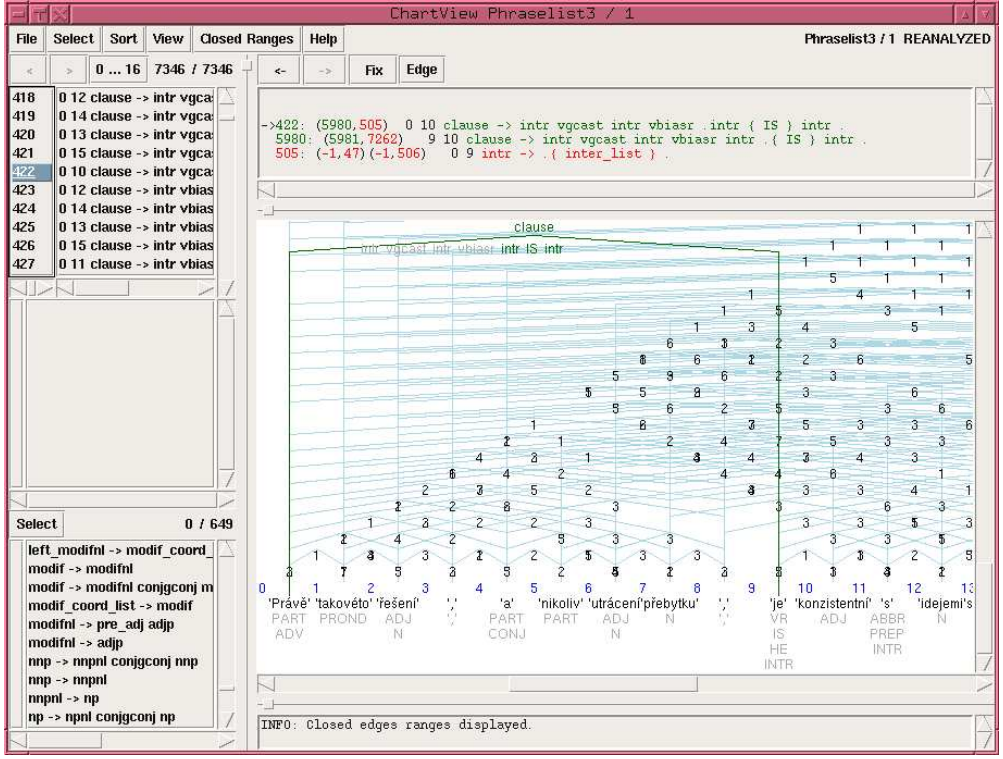


V závěrečném komuniké o jednáních se stroze konstatuje, že jednání budou v brzké době pokračovat v Římě.





◂ Zpět



◂ Zpět

np: *Tyto normy se však odlišují nejen v rámci různých národů a států, ale i v rámci sociálních skupin, a tak považuji 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.*

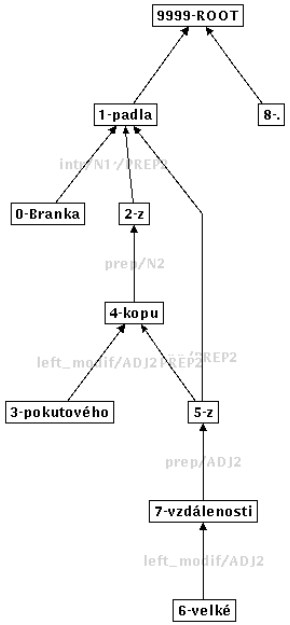
[0-5): byl býval věděl
[6-10): byl bych nechodil

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

[0-9): Muž , , vede kolo
[2-6): který stojí u cesty

◂ Zpět

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, gInPcld1, gInPc4d1, gInPc5d1, gInScld1wH, gInSc4d1wH, gInSc5d1wH, gMnPc4d1, gMnScld1wH, gMnSc5d1wH, gNnScld1, gNnSc4d1, gNnSc5d1}	k2eAgFnSc6d1
dlouhé	k2eA{gFnPcld1, gFnPc4d1, gFnPc5d1, gFnSc2d1, gFnSc3d1, gFnSc6d1, gInPcld1, gInPc4d1, gInPc5d1, gInScld1wH, gInSc4d1wH, gInSc5d1wH, gMnPc4d1, gMnScld1wH, gMnSc5d1wH, gNnScld1, gNnSc4d1, gNnSc5d1}	k2eAgFnSc6d1
ulici	klgFnSc3, klgFnSc4, klgFnSc6	klgFnSc6
stálo	k5eAaImAgNnSaIrD	kSeApNnStMmPaI
moderní	k2eA{gFnPcld1, gFnPc4d1, gFnPc5d1, gFnScld1, gFnSc2d1, gFnSc3d1, gFnSc4d1, gFnSc5d1, gFnSc6d1, gFnSc7d1, gInPcld1, gInPc4d1, gInPc5d1, gInScld1, gInSc4d1, gInSc5d1, gMnPcld1, gMnPc4d1, gMnPc5d1, gMnScld1, gMnSc5d1, gNnPcld1, gNnPc4d1, gNnPc5d1, gNnScld1, gNnSc4d1, gNnSc5d1}	k2eAgNnScld1, k2eAgNnSc4d1, k2eAgNnSc5d1
nablýskané	k2eA{gFnPcld1rD, gFnPc4d1rD, gFnPc5d1rD, gFnSc2d1rD, gFnSc3d1rD, gFnSc6d1rD, gInPcld1rD, gInPc4d1rD, gInPc5d1rD, gInScld1wHrD, gInSc4d1wHrD, gInSc5d1wHrD, gMnPc4d1rD, gMnScld1wHrD, gMnSc5d1wHrD, gNnScld1rD, gNnSc4d1rD, gNnSc5d1rD}	k2eAgNnScld1, k2eAgNnSc4d1, k2eAgNnSc5d1
auto	klgNnScl, klgNnSc4, klgNnSc5	klgNnScl, klgNnSc4, klgNnSc5

◂ Zpět

Systém synt – příklad logické analýzy

vyhodnocení `rule_schema` pro `np` 'pečené kuře'

```
4, 6, -npnl -> . left_modif  np .: k1gNnSc145
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:
```

$\lambda w_2 \lambda t_3 \lambda x_4([\textbf{pečený}_{w_2 t_3}, x_4] \wedge [\textbf{kuře}_{w_2 t_3}, x_4]) \ldots (o\iota)_{\tau\omega}$

```
And constrs: none added
Exi vars: none added
```


Systém 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:
```

$On \ldots \iota$

```
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:
```

$\lambda t_{12} \textbf{dnes}_{tt_{12}} \ldots (o\tau)$

```
frequency:
```

$\textbf{Onc} \ldots ((o(o\tau))\pi)_\omega$

```
verbal object:
```

$x_{15} \ldots (o(o\pi)(o\pi))$

```
present tense clause:
```

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

```
clause:
```

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