

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
- ▶ Metagramatika systému synt

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 – lexikální hlava

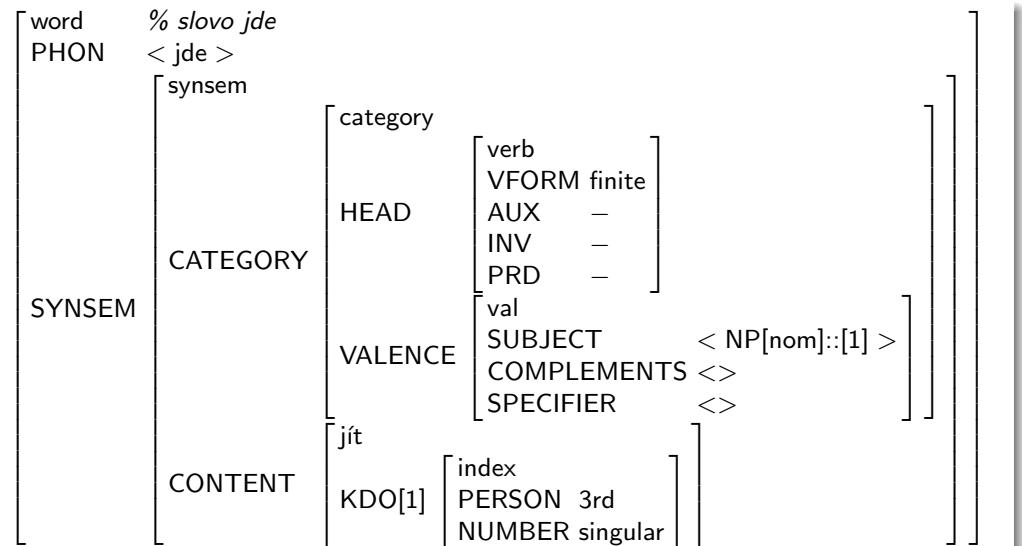
- ▶ gramatika je v HPSG modelována pomocí **uspořádaných příznakových struktur**, které korespondují s typy výrazů přirozeného jazyka a jejich částmi
- ▶ cílem teorie je detailní specifikace, které příznakové struktury jsou **přípustné**
- ▶ příznakové struktury definují **omezení** hodnoty příznaků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

- ▶ **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

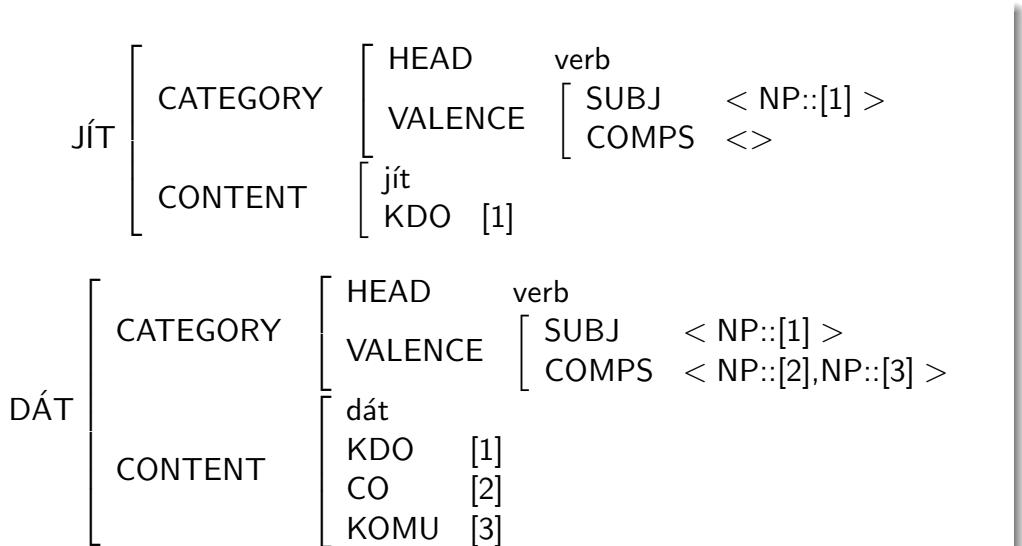
HPSG struktury jsou **typované příznakové struktury**

zapisují se pomocí AVM – **příznaky** velkými písmeny, **typy** malými



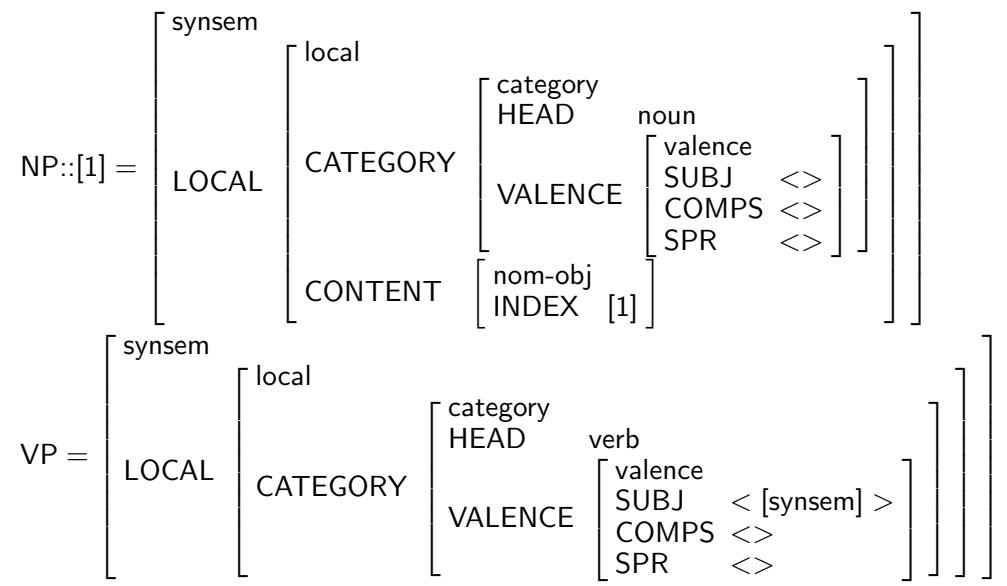
HPSG – lexikální položky

velké množství akcí je v **lexikonu**:



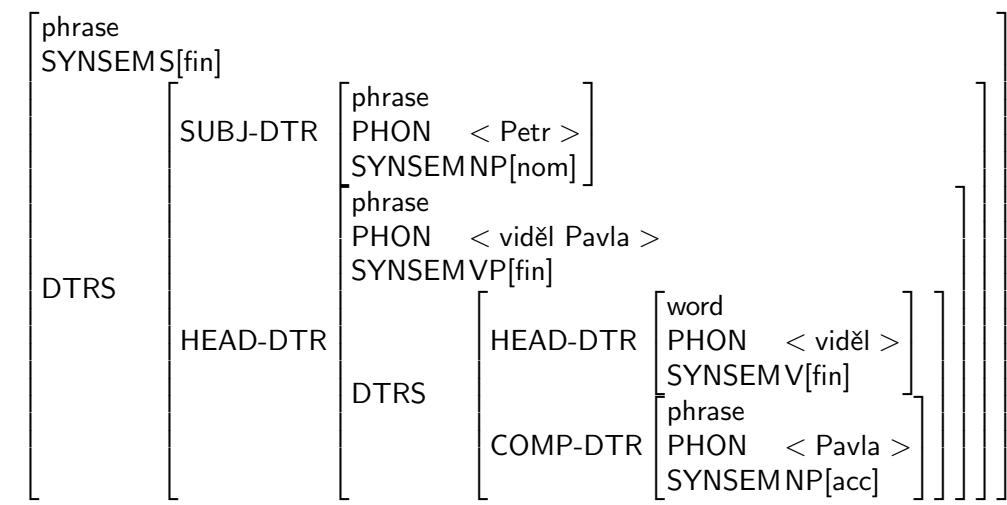
HPSG – syntaktické kategorie

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



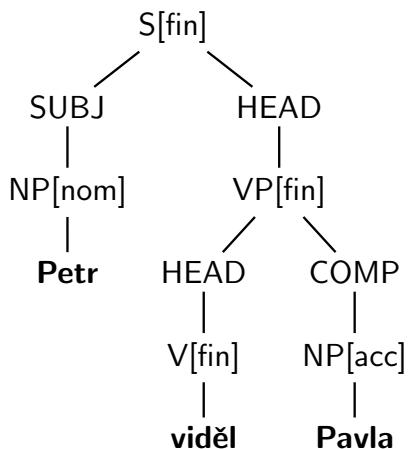
HPSG – fráze

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

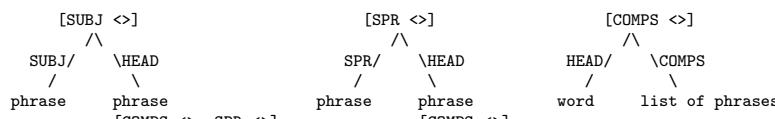
...

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*, ...

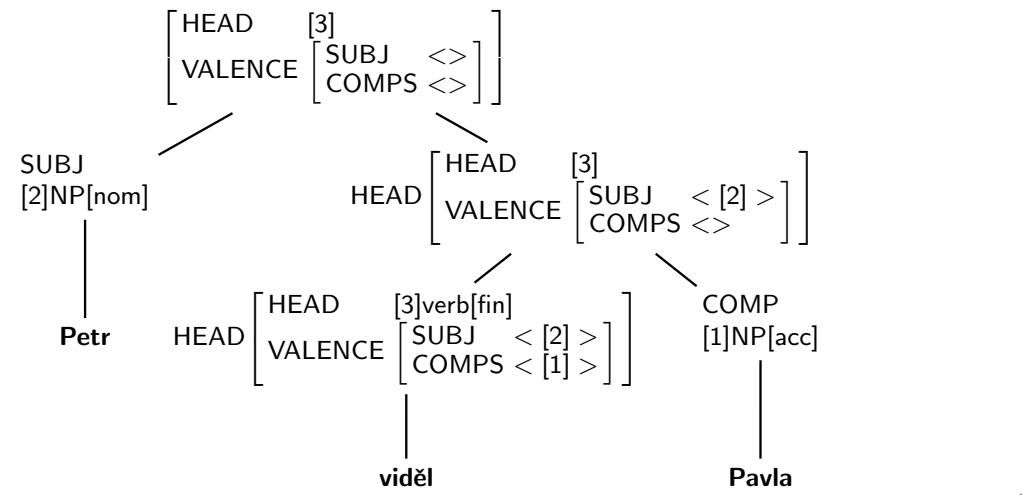


- **omezení hlavových příznaků** – pro každou frázi, která má hlavu, musí být hlavové příznaky fráze shodné s hlavovými příznaky potomka, který je hlavou

- **valenční omezení** – pro každý z valenčních příznaků (SUBJECT, COMPLEMENTS, ...) – hodnota příznaku na hlavové frázi musí odpovídat hodnotě na potomku, který je hlavou, míinus ty příznaky, které jsou splněny některým z nehlavových potomků

HPSG – dobré 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>

Úvod do počítačové lingvistiky 7/11

13 / 34

```

G1 - metagrammar
vol_list -> VOL
/* muset a chtit */
list_coord voi_list
/* muset */
voi_list -> VOI

/* budu muset a bude chtit */
list_coord vbvoui_list
/* budu muset */
vbvoui_list --> order(WBU,voi_list)

/* musel jsem a chtel jsem */
list_coord volvhb12_list
/* musel jsem */
volvhb12_list --> order(vol_list,VB12)

/* musel bych a chtel bych */
list_coord volvhk_list
/* musel bych */
volvhk_list --> order(vol_list,VB12)

G2
volvhk_list -> volvhk_listnl conjconj v
depends($2,$1,$3)
head($2)
volvhk_list -> volvhk_listnl

/* musel bych */
depends($2,$1,$3)
head($2)
volvhk_listnl -> vol_list intr VBK
volvhk_listnl -> VBK intr vol_list

prep -> prepnl conjconj prep
depends($2,$1,$3)
head($2)
agree_case_and_propagate($1,$3)
prep -> prepnl
propagate_case($1)

/* bez */
prepnl -> PREP
propagate_case($1)

pn -> prep np
agree_case_and_propagate($1, $2)
depends($1,$2)
add_prep_nrgroup($2)
rule_schema($8,"lvt([awt(#1),try(#2)])")

pp -> ppnl conjconj PP
depends($2,$1,$3)
head($2)
pp -> ppnl

/* z mesta */
ppnl -> pn
/* castecne i z mesta */
ppnl -> part pn
head($2)

/* on ten (Petr je pekný ...) */
first_pron_group -> ON first_pron
agree_case_number_gender_and_propagate
depends($2,$1)
# depends($2,$3)
/* ten (Petr je pekný ...) */

G3
volvhb12_list -> volvhb12_listnl conjconj v
depends($2,$1,$3)
head($2)
volvhb12_listnl -> vol_list intr VB12
volvhb12_listnl -> VB12 intr vol_list
volvhk_list -> volvhk_listnl conjconj volv
volvhk_listnl -> volvhk_listnl
volvhk_listnl -> vol_list intr VBK
volvhk_listnl -> VBK intr vol_list
prep1 -> prepnl1 conjconj prep1
prep2 -> prepnl2 conjconj prep2
prep3 -> prepnl3 conjconj prep3
prep4 -> prepnl4 conjconj prep4
prep5 -> prepnl5 conjconj prep5
prep6 -> prepnl6 conjconj prep6
prep7 -> prepnl7 conjconj prep7
prep1 -> prepnl1
prep2 -> prepnl2
prep3 -> prepnl3
prep4 -> prepnl4
prep5 -> prepnl5
prep6 -> prepnl6
prep7 -> prepnl7
prepnl1 -> PREP1
prepnl2 -> PREP2
prepnl3 -> PREP3
prepnl4 -> PREP4
prepnl5 -> PREP5
prepnl6 -> PREP6
prepnl7 -> PREP7
PREP1 -> PREP1SM
PREP1 -> PREP1SI
PREP1 -> PREP1SF
PREP1 -> PREP1SN
PREP1 -> PREP1PM
PREP1 -> PREP1PI
PREP1 -> PREP1PP
PREP1 -> PREP1PN
PREP2 -> PREP2SM
PREP2 -> PREP2SI
PREP2 -> PREP2SF
PREP2 -> PREP2SN
PREP2 -> PREP2PM
PREP2 -> PREP2PI
PREP2 -> PREP2PP
PREP2 -> PREP2PN

/* budu se ptát */
clause ==> order(VBU,R,VRI)

/* který ... */
relclause ==> first(relprongr) rhs(clause)
  
```

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

Úvod do počítačové lingvistiky 7/11

14 / 34

Metagramatika systému synt

Kombinatorické konstrukty

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 – generativní konstrukty

skupina výrazů %list_* – produkují nová pravidla pro seznamy (s oddělovačí/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 – globální omezení pořadí

globální omezení pořadí zakazuje některé kombinace pořadí preterminálů
 %enclitic – které preterminály jsou brány jako **příklonky**
 %order – zajišťuje dodržení precedenze zadaných preterminálů

```
/* jsem, bych, se */
%enclitic = (VB12, VBK, R)
/* byl — četl, ptal, musel */
%order VBL = {VL, VRL, VOL}
```

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($0,"(#1)")
          groupflag($1,"head"),
    VR R: verb_rule_schema($0,"(#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($0,"#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 fenomény
- ▶ 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]))")
```

## 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 |

rule\_schema – schéma pro tvorbu logické konstrukce ze subkonstrukcí  
projdou jenom kombinace, které **typově vyhovují** danému schématu

## 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

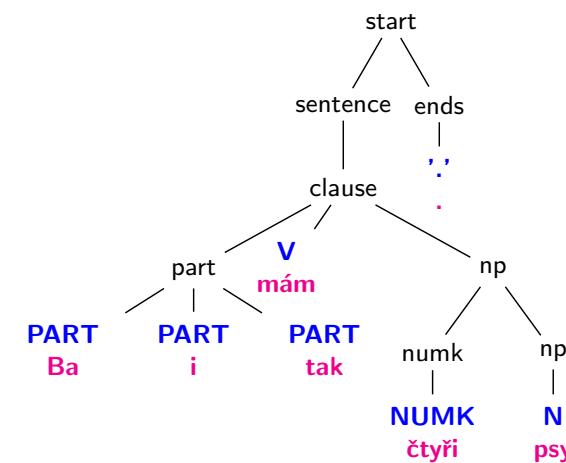
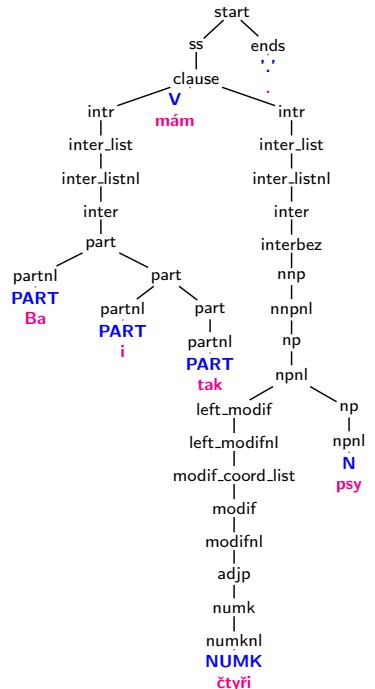
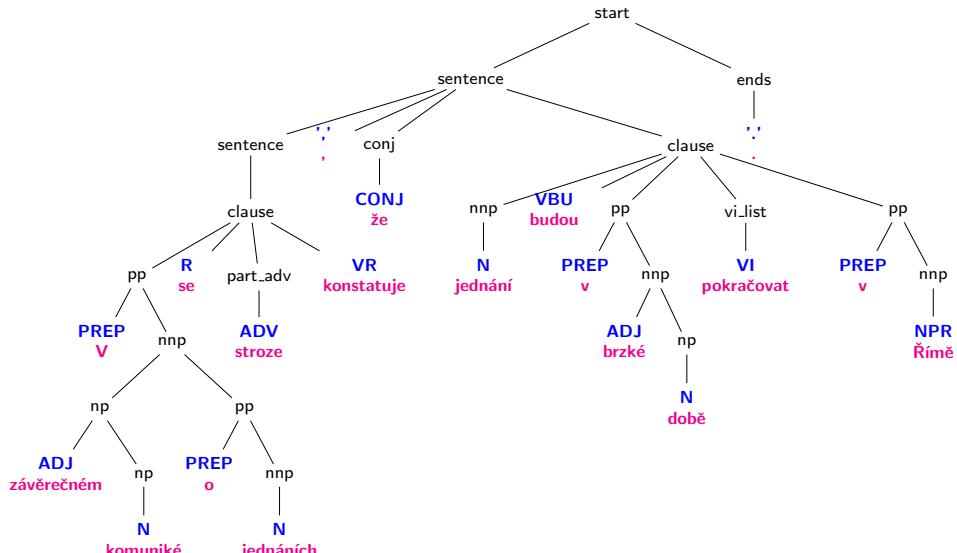
[http://nlp.fi.muni.cz/projekty/grammar\\_workbench/manual/](http://nlp.fi.muni.cz/projekty/grammar_workbench/manual/)

DEMO: [wwwsynt](http://wwwsynt.cz) – webové rozhraní k syntu

<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.*



ChartView Phraselist3 / 1

File Select Sort View Closed Ranges Help

Phraselist3 / 1 REANALYZE

< > 0 ... 16 7346 / 7346 Fix Edge

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 clauses -> intr vbia  
 424 0 14 clause -> intr vbia  
 425 0 13 clause -> intr vbia  
 426 0 15 clause -> intr vbia  
 427 0 11 clause -> intr vbia

->422: (5980,505) 0 10 clause -> intr vgcast intr vbiastr . intr ( IS ) intr .  
 5980: (5981,7262) 9 10 clause -> intr vgcast intr vbiastr intr ( IS ) intr .  
 505: (-1,47)(-1,506) 0 9 intr -> .( inter\_list ) .

INFO: Closed edges ranges displayed.

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

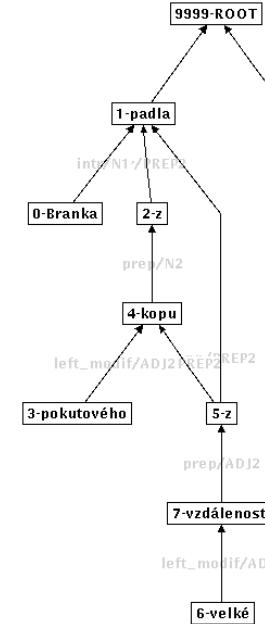
[6-10): býl bých nechodil

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

[0-9) · Muž vede kolo

[2-6]: který stojí u cesty

*Branka padla z pokutového kopu z velké vzdálenosti*



◀ Zp

| slovo      | před                                                                                                                                                                                                                                                                               | po                                       |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Na         | k7{c4, c6}                                                                                                                                                                                                                                                                         | k7c6                                     |
| krásné     | k2eA{gFnPcl1d, gFnPc4d1, gFnPc5d1, gFnSc2d1, gFnSc3d1, gFnSc6d1, gInPcl1d, gInPc4d1, gInPc5d1, gInScl1wH, gInSc4d1wH, gInSc5d1wH, gMnPc4d1, gMnScl1wH, gMnSc5d1wH, gNnScl1d, gNnSc4d1, gNnSc5d1}                                                                                   | k2eAgFnSc6d1                             |
| dlouhé     | k2eA{gFnPcl1d, gFnPc4d1, gFnPc5d1, gFnSc2d1, gFnSc3d1, gFnSc6d1, gInPcl1d, gInPc4d1, gInPc5d1, gInScl1wH, gInSc4d1wH, gInSc5d1wH, gMnPc4d1, gMnScl1wH, gMnSc5d1wH, gNnScl1d, gNnSc4d1, gNnSc5d1}                                                                                   | k2eAgFnSc6d1                             |
| ulici      | klgFnSc3, klgFnSc4, klgFnSc6                                                                                                                                                                                                                                                       | klgFnSc6                                 |
| stálo      | k5eAaImAgNnSaIrD                                                                                                                                                                                                                                                                   | kSeApNnStMmPaI                           |
| moderní    | k2eA{gFnPcl1d, gFnPc4d1, gFnPc5d1, gFnScl1d, gFnSc2d1, gFnSc3d1, gFnSc4d1, gFnSc5d1, gFnSc6d1, gFnSc7d1, gInPcl1d, gInPc4d1, gInPc5d1, gInScl1d, gInSc4d1, gInSc5d1, gMnPcl1d, gMnPc4d1, gMnPc5d1, gMnScl1d, gMnSc5d1, gNnPcl1d, gNnPc4d1, gNnPc5d1, gNnScl1d, gNnSc4d1, gNnSc5d1} | k2eAgNnScl1d, k2eAgNnSc4d1, k2eAgNnSc5d1 |
| nablýskané | k2eA{gFnPcl1rD, gFnPc4d1rD, gFnPc5d1rD, gFnSc2d1rD, gFnSc3d1rD, gFnSc6d1rD, gInPcl1rD, gInPc4d1rD, gInPc5d1rD, gInScl1wHrD, gInSc4d1wHrD, gInSc5d1wHrD, gMnPc4d1rD, gMnPc5d1wHrD, gMnSc5d1wHrD, gNnScl1rD, gNnSc4d1rD, gNnSc5d1rD}                                                 | k2eAgNnScl1, k2eAgNnSc4d1, k2eAgNnSc5d1  |
| auto       | klgNnScl, klgNnSc4, klgNnSc5                                                                                                                                                                                                                                                       | klgNnScl, klgNnSc4, klgNnSc5             |

## Systém synt – příklad logické analýzy

vyhodnocení rule\_schema pro np 'pečené kuře'

```
4, 6, -npln -> . 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:

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

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* ...  $\iota$

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

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

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

time span:  $\lambda t_{12} \mathbf{dnes}_{tt_{12}} \dots (\sigma\tau)$

frequency: **Onc**...((o(o\sigma))\pi) $_{\omega}$

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

present tense clause:

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

clause:

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