

# Syntaktická a logická analýza češtiny v analyzátoru synt

Aleš Horák

E-mail: [hales@fi.muni.cz](mailto:hales@fi.muni.cz)  
<http://nlp.fi.muni.cz/projekty/wwwsynt/>

Obsah:

- Motivace
- Syntaktický analyzátor synt
- Logická analýza v systému synt
- Shrnutí

# Motivace

Cíl = počítačový systém dovede **komunikovat v přirozeném jazyce**

Obor **Počítačová lingvistika**:

- od 60. let 20. stol., *Computational linguistics*, **NLP** (*Natural Language Processing*)
- spojení **umělé inteligence** (informatiky) a **lingvistiky** – jako jedna z **kognitivních věd**
- zkoumá problémy **analýzy** či **generování** textů nebo mluveného slova, které vyžadují určitou (ne absolutní) míru porozumění přirozenému jazyku strojem
- tvoří **jazykové modely** – pojmy **algoritmus**, **datová struktura**, **(formální) gramatika**, ...

# Komunikace v přirozeném jazyce

SITUACE

*Mluvčí* → *Promluva* → *Posluchač*

$KS = \{Ml, Po, Pr, Loc, Tm, Var\}$

**komunikace** = cílená výměna informace pomocí produkce a vnímání  
(sdílených) pokynů

**přirozený jazyk** = systém strukturovaných pokynů

průběh promluvy je možné rozdělit na **fáze**: záměr, generování, syntéza a  
vnímání, analýza, zjednoznačnění, zahrnutí      (Russell a Norvig, 2003)

# Komunikační fáze – příklad

záměr

 $Vědět(Po, \neg Na\_živu(Krtek_1, S_3))$ 

generování

"Krtek je mrtvý."

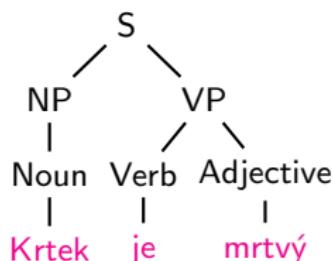
syntéza

[k r t e k j e m r t v i:]

MLUVČÍ

vnímání

analýza

syntaktická  
analýza:"Krtek je  
mrtvý."sémantická  
interpretace:  
 $\neg Na\_živu(Krtek, Ted')$   
 $Unavený(Krtek, Ted')$ pragmatická  
interpretace:  
 $\neg Na\_živu(Krtek_1, S_3)$   
 $Unavený(Krtek_1, S_3)$ 

zjednoznačnění

 $\neg Na\_živu(Krtek_1, S_3)$ 

POSLUCHAČ

zahrnutí

 $Tell(KB, \neg Na\_živu(Krtek_1, S_3))$

# Obsah

1 Motivace

2 Syntaktický analyzátor synt

- Metody analýzy syntaxe
- Reálná syntaktická analýza češtiny
- Syntaktický analyzátor synt
- Metagramatika systému synt
- Jazykové zdroje
- Výstupy syntaktické analýzy

3 Logická analýza v systému synt

4 Shrnutí

# Metody analýzy syntaxe

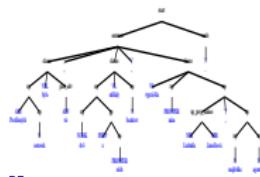
**syntaktická analýza** – struktura větných frází

popisuje, jak vypadá **gramaticky správná věta**, většinou pomocí:

- pravidel gramatiky
- pravděpodobnostního ohodnocení

při vývoji se využívají data z **rozsáhlých textových korpusů**

**syntaktický analyzátor** – nástroj, který analyzuje vstup (nejčastěji **větu**) a pro *správnou větu* dá na výstup informaci o její **struktuře** (např. derivační stromy)

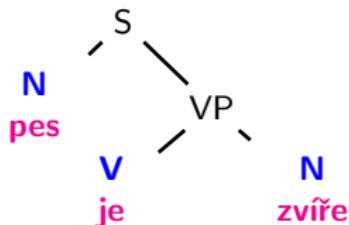


# Metody analýzy syntaxe – reprezentace větné struktury

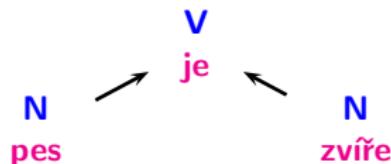
**větná struktura** – strukturovaný popis větných členů

- povrchová struktura (*surface structure*)

derivační/složkový strom jako výsledek bezkontextové (CF) analýzy



- závislostní struktura (*dependency structure*)  
zobrazuje závislosti mezi větnými členy



# Reálná syntaktická analýza češtiny

- silná víceznačnost – někdy až obrovské množství (milióny) možných syntaktických stromů

*Obehnat Šalounův pomník mistra Jana Husa na pražském Staroměstském náměstí živým plotem z hustých keřů s trny navrhuje občanské sdružení Společnost Jana Jesenia.*

- velice rozsáhlé gramatiky (tisíce až desítky tisíc pravidel)
- existují efektivní algoritmy pro takové gramatiky  
např. tabulkový analyzátor (*chart parser*), beží v  $O(n^3)$ , stovky slov/sekundu

# Syntaktický analyzátor synt

- navržen pro analýzu **jazyků s volným slovosledem** (čeština)
- dlouhodobý projekt **Centra ZPJ** (vede Aleš Horák – Andrej Gardoň, Jiří Golembiovský, Dana Hlaváčková, Lukáš Holík, Miloš Jakubíček, Vladimír Kadlec, Vojtěch Kovář, Jan Michelfeit, Pavel Smrž, Radek Vykydal)
- pracuje s formalismem **metagramatiky** – bezkontextová pravidla + kontextové podmínky a akce
- **rychlý algoritmus** ( $0.07s/věta$ ) pro kontextovou část, umí efektivně pracovat s **exponenciálním množstvím stromů**
- výsledné stromy jsou v polynomiálním čase **ohodnoceny** a setříděny – na výstup jde jen **N nejlepších**
- systém obsahuje i prototypové části pro **výstup závislostí** a **logickou analýzu věty**

# Metagramatika systému synt

3 formy (meta)gramatiky:

- **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

počty pravidel

metagramatika G1	253
gramatika G2	3091
expandovaná gramatika G3	11530

▶ ukázka

# Jazykové zdroje

synt využívá data z následujících systémů a zdrojů:

- [ajka](#) – morfologická analýza (slovní tvary, lemmata)
- [VerbaLex](#) – valenční slovník českých sloves:
  - syntaktické informace o valenci (pád, předložka, ...)
  - odkazy na hypero/hyponymické třídy českého WordNetu
  - sémantické třídy sloves (VerbNet)
  - >10,000 slovesných lemmat
- [manatee](#) – korpusový manažer

▶ ukázka

# 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** – webové rozhraní k syntu

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

# Obsah

1 Motivace

2 Syntaktický analyzátor synt

3 Logická analýza v systému synt

- Sémantická a logická analýza
- Logická analýza přirozeného jazyka
- Transparentní intenzionální logika
- Tvorba logické konstrukce v syntu
- synt – příklad logické analýzy
- Dolphin – báze znalostí v TIL

4 Shrnutí

# Sémantická a logická analýza

**sémantická analýza** – význam výrazů přirozeného jazyka a jejich kombinací  
hodně závisí na zvolené **sémantické reprezentaci**

**logická analýza věty** – strukturní část sémantické analýzy

**Princip kompozicionality:**

Význam složeného tvrzení je funkcí významu jednotlivých komponent.

(je určován, je odhadnutelný, každá složka hraje význam?)  
omezení PK: idiomy, ustrnulé metafore, kolokace, klišé

# Logická analýza přirozeného jazyka

logická analýza PJ – analýza významu výrazů (vět) PJ  
přirozený jazyk = nástroj pojmového uchopení reality

pro úplnou logickou analýzu PJ potřebujeme logický systém, který:

- umí pracovat s pojmy
- umí pracovat s potencialitami a časem – *intenzionalita*

# Logická analýza přirozeného jazyka – pojem

co je **pojem**?

- **pojem  $\neq$  výraz** – např. výrazy v různých jazycích často reprezentují stejný pojem ( $\text{pojem}(\text{"prvočíslo"}) \equiv \text{pojem}(\text{"prime number"})$ )
- **pojem  $\neq$  představa** – představa je *subjektivní*, pojem je *objektivní*
- pojmy mohou **identifikovat** různé objekty: jedno *individuum* (např. Petr, prezident ČR), *třídu* objektů (např. červený, šelma), *pravdivostní hodnotu* (např. v Brně prší), ...
- **pojem = logická konstrukce** (procedura, např. “**planeta**” = třída nebeských těles s určitými charakteristikami – obíhá po oběžné dráze kolem stálice, není zdrojem světla, ...)

# Logická analýza přirozeného jazyka – intenzionalita

- **intenze** – objekty typu funkcí, jejichž hodnoty závisí na světě a čase  
(samotné intenze tak na světě a čase *nezávisí*)
- **extenze** – ostatní objekty (na světě a čase nezávislé)

Varšava

hlavní město Polska

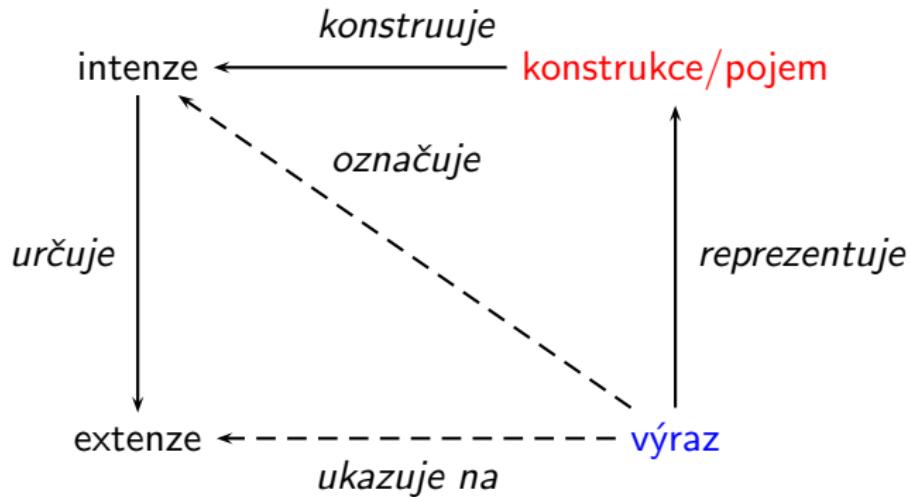
Varšava

– jméno individua, jasně identifikovatelné a odlišitelné

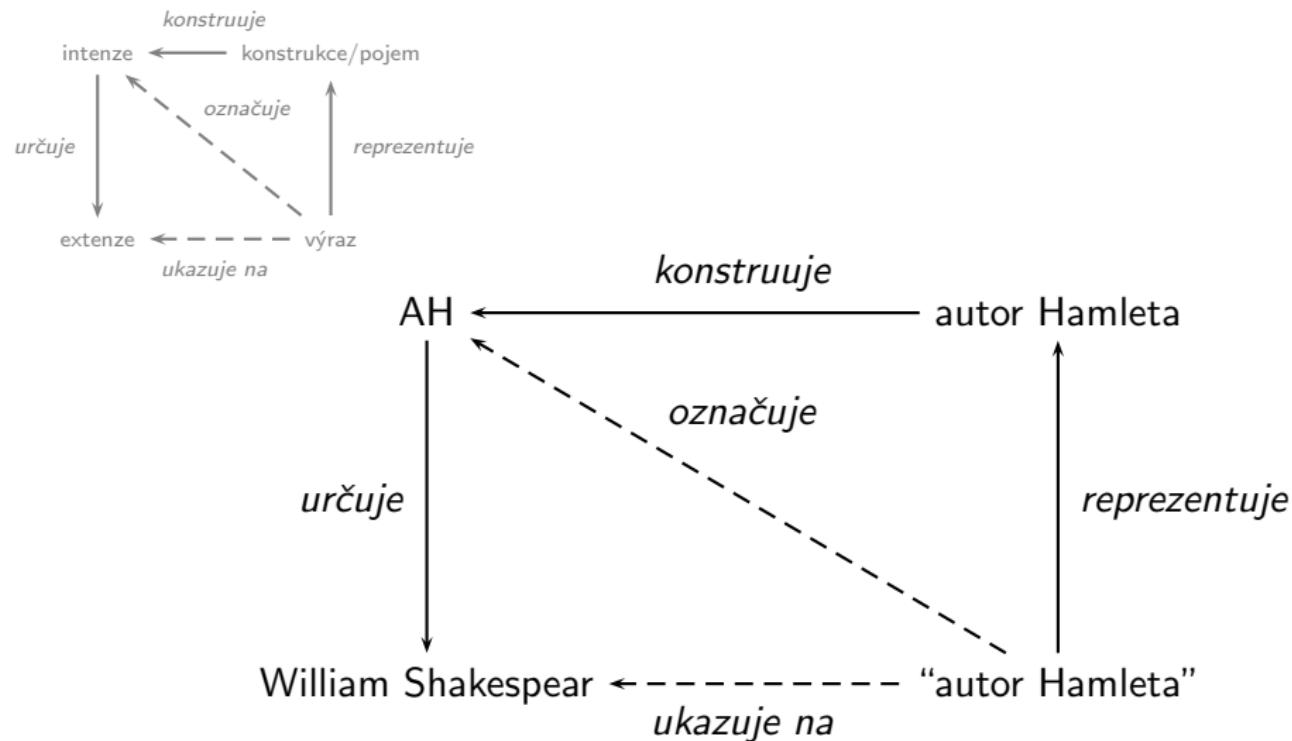
hlavní město Polska

– individuová role, momentálně identifikuje Varšavu, ale dříve to byl i Krakov

# Rozšířený vztah výrazu a významu u intenzí



# Rozšířený vztah výrazu a významu u intenzí



# Transparentní intenzionální logika

- *Transparent Intensional Logic*, TIL
- logický systém speciálně navržený pro zachycení **významu výrazů** PJ
- autor **Pavel Tichý**: *The Foundations of Frege's Logic*, de Gruyter, Berlin, New York, 1988.
- Tichý vychází z myšlenek – *Gottlob Frege* (1848 – 1925, logik) a *Alonzo Church* (1903 – 1995, teorie typů)
- vlastnosti:
  - rozvětvená **typová hierarchie** (s typy vyšších řadů) – typová báze 4 typů (**individua**, **pravdivostní hodnota**, **čas/kontinuum**, **možné světy**)
  - **temporální**
  - **intenzionální** (intenze × extenze)
  - potenciality v podobě **možných světů**
  - **vyvozování** pomocí **kalkulu sekventů**
- implementace v syntu – algoritmus NTA

# Tvorba logické konstrukce v syntu

## Lexikon základních konstrukcí:

```
#T_PROP (((oi)t)w) /* definice typu:  $(oi)_{\tau\omega} = \text{vlastnost individu}\iota$  */
červený
0(červený/T_PROP) /* trivializace objektu,  ${}^0\text{červený} \dots (oi)_{\tau\omega}$  */
jablko
/* připsání vlastnosti,  $(\exists i \dots \iota)[{}^0\text{jablko}_{wt} i]$  */
:exists: V(i/i):V(i) :and: [awt(0(jablko/T_PROP)),V(i)]
```

## 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 z podkonstrukcí projdou jenom kombinace, které **typově vyhovují** danému schématu

# Systém synt – příklad logické analýzy

vyhodnocení **rule\_schema** pro **np** 'velkou kytici'

```

4, 6, +npln -> . left_modif np .: k1gFnSc4
agree_case_number_gender_and_propagate OK
rule_schema: 2 nterms, 'lwtx(awtx(#1) and awtx(#2))'
error: appl_arr: arg type does NOT match (i!=((oi)t)w))
...
rule_schema: 2 nterms, 'lwtx([[awt(#1),#2],V(x)])'
And constrs, Abstr and Exi vars are just gathered
1 (1x2) constructions:
 $\lambda w_{10} \lambda t_{11} \lambda x_{12} [[\text{velký-4}_{w_{10} t_{11}}, \text{kytice-5}], x_{12}] \dots (o\iota)_{\tau\omega}$ 
And constrs: none added
Exi vars: none added

```

# Systém synt – příklad logické analýzy – pokrač.

vyhodnocení **verb\_rule\_schema** pro celou větu (**clause**)

*Petr dal včera babičce velkou kyticí.*

verb\_rule\_schema: 4 groups, verb #2, possible subjects: #1

group 1: 0, 1, +NPR -> .'Petr' . k1gMnSc1, **Petr-0**...

group 2: 2, 3, +ADV -> .'včera' . k6xTeAd1, temp:**včera-2**...((o $\tau$ ) $\tau$ )

group 3: 3, 4, +N -> .'babičce' . k1gFnSc3, **babička-3**...( $o\iota$ ) $\tau\omega$

group 4: 4, 6, +npnl -> . left\_modif np . k1gFnSc4,

$\lambda w_{10} \lambda t_{11} \lambda x_{12} [[\text{velký-4}_{w_{10} t_{11}}, \text{kytice-5}], x_{12}] \dots (o\iota)_{\tau\omega}$

Clauses val.list: dát <v>#1:(0)hPTc1-#2:(2)hPTc3-#3:(3)hPTc4, ...

Verbs val.list: dát <v>#1:hPTc3ti-#2:hPTc4ti

Matched val.list: dát(0) <v>#1:(2)hPTc3ti-#2:(3)hPTc4

time span:  $\lambda t_{23} \mathbf{včera-2}_{tt_{23}} \dots (o\tau)$

frequency: **Onc**...((o(o $\tau$ )) $\pi$ ) $\omega$

verbal object:  $x_{26} \dots (o(o\tau)(o\pi))$

present tense clause:  $\lambda w_{29} \lambda t_{30} (\exists x_{26})(\exists i_{27})(\exists i_{28})([\mathbf{Does}_{w_{29} t_{30}}, \mathbf{Petr-0}, \dots$

clause:

$\lambda w_{31} \lambda t_{32} [\mathbf{P}_{t_{32}}, [\mathbf{Onc}_{w_{31}}, \lambda w_{29} \lambda t_{30} (\exists x_{26})(\exists i_{27})(\exists i_{28})([\mathbf{Does}_{w_{29} t_{30}}, \mathbf{Petr-0},$   
 $[\mathbf{Imp}_{w_{29}}, x_{26}]] \wedge [\mathbf{babička-3}_{w_{29} t_{30}}, i_{27}] \wedge [[\mathbf{velký-4}_{w_{29} t_{30}}, \mathbf{kytice-5}], i_{28}] \wedge$   
 $x_{26} = [\mathbf{dát-1}, i_{27}, i_{28}]_{w_{29}})], \lambda t_{23} \mathbf{včera-2}_{tt_{23}}] \dots \pi$

# Dolphin – báze znalostí v TIL

systém **Dolphin** (Andrej Gardoň):

- prototypová implementace báze znalostí v TIL
- v současnosti základní inference bez možných světů a časů (and, or, neg a specifikátor)
- organizace dat
  - **jazyková** vrstva
  - **objektová** vrstva
- interaktivní módy:
  1. **učící mód** – kontroluje konzistenci, volné proměnné ukládá jako nové objekty, ukládá nová fakta
  2. **dotazovací mód** – spouští vyhledávání pro základní typy otázek (*yes/no, what*)

▶ ukázka

# Obsah

1 Motivace

2 Syntaktický analyzátor synt

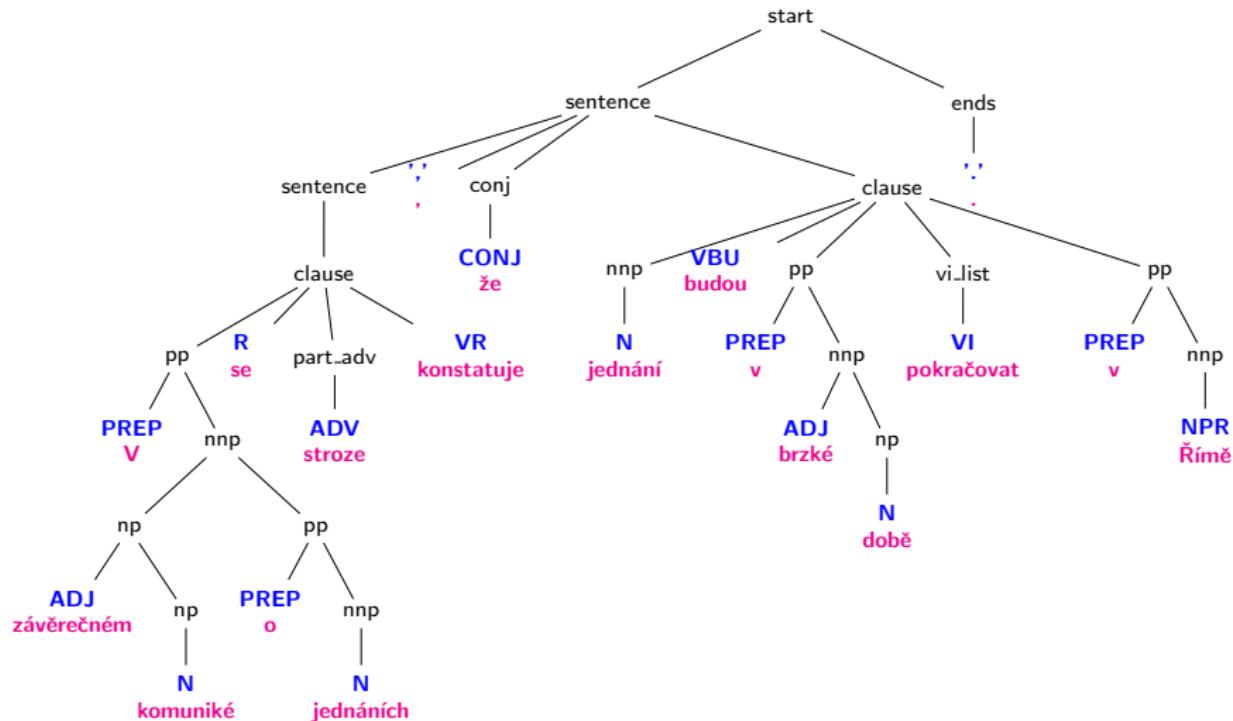
3 Logická analýza v systému synt

4 Shrnutí

# Shrnutí

- **synt** je navržen a implementován jako efektivní nástroj pro syntaktickou analýzu silně víceznačných gramatik
- **metagramatika** syntu implementuje pravidla podle české lingvistické tradice
- **synt** umožňuje pracovat s velkým množstvím potenciálních analýz – v současnosti se implementují nové techniky na vylepšení ohodnocení stromů
- **synt** umožňuje práci i přímo s komprimovanou strukturou všech stromů (*chart*) – graf závislostí, vyhledání podstruktur a frází, částečné zjednoznačnění značek vstupu
- implementace algoritmu NTA – logická analýza české věty v TIL, překlad vstupu v přirozeném jazyce do logických konstrukcí
- **synt** → základ pro komunikaci s počítačem v přirozeném jazyce

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



G1 - metagrammar

```

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

/* budu muset a budu chtit */
%list_coord vbuvoi_list
/* budu muset */
vbuvoi_list --> order(VBU, voi_list)

/* musel jsem a chtel jsem */
%list_coord volvbk12_list
/* musel jsem */
vovbk12_list --> order(vol_list, VB12)

/* musel bych a chtel bych */
%list_coord volvbk_list
/* musel bych */
vovbk_list --> order(vol_list, VBK)

%list_coord_case prep
/* bez */
prep -> PREP
  propagate_case($1)

pn -> prep nnp
  agree_case_and_propagate($1, $2)
  depends($1, $2)
  add_prep_ngroup($2)
  rule_schema($0, "lwt([awt(#1), try(#2)])")

%list_coord pp
/* z mesta */
pp -> pn
/* castecne i z mesta */
pp -> part pn
  head($2)
#/* z mesta nez z vesnice */
#pp -> pn NEZ pn
#  depends($2, $1)
#  depends($2, $3)

```

Rules: 1 / 345

G2

```

volvbk_list -> volvbk_listnl conjgconj v
  depends($2, $1, $3)
  head($2)
vovbk_list -> volvbk_listnl

/* musel bych */
vovbk_listnl -> vol_list intr VB12
vovbk_listnl -> VB12 intr vol_list
vovbk_list -> volvbk_listnl conjgconj v
vovbk_listnl -> vol_list intr VBK
vovbk_listnl -> VBK intr vol_list

prep -> prepnl conjgconj 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 nnp
  agree_case_and_propagate($1, $2)
  depends($1, $2)
  add_prep_ngroup($2)
  rule_schema($0, "lwt([awt(#1), try(#2)])")

pp -> ppnl conjgconj 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 pekny ...) */
first_pron_group -> ON first_pron
  agree_case_number_gender_and_propagate
    head($2)
    head($1)
/* ten (Petr je pekny ...) */

```

Rules: 2 / 3102

G3

```

volvbk12_list -> volvbk12_listnl
volvbk12_listnl -> vol_list intr WB12
volvbk12_listnl -> WB12 intr vol_list
vovbk_list -> volvbk_listnl conjgconj v
vovbk_list -> volvbk_listnl conjgconj v
vovbk_listnl -> vol_list intr VBK
vovbk_listnl -> VBK intr vol_list
prep1 -> prepnl1 conjgconj prep1
prep2 -> prepnl2 conjgconj prep2
prep3 -> prepnl3 conjgconj prep3
prep4 -> prepnl4 conjgconj prep4
prep5 -> prepnl5 conjgconj prep5
prep6 -> prepnl6 conjgconj prep6
prep7 -> prepnl7 conjgconj 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 -> PREP1PF
PREP1 -> PREP1PN
PREP2 -> PREP2SM
PREP2 -> PREP2SI
PREP2 -> PREP2SF
PREP2 -> PREP2SN
PREP2 -> PREP2PM
PREP2 -> PREP2PI
PREP2 -> PREP2PF
PREP2 -> PREP2PN

```

Rules: 14 / 11556

File: /mnt/scsi-5/nlp/projekty/grammar\_workbench/synt/synt/grammars/synt.g1

**Close** Clicked Line: 763

◀ Zpět

- A (121)
- B (480)
- C (156)
- Č (136)
- D (1236)
- Ď (3)
- E (19)
- F (43)
- G (9)
- H (181)
- CH (100)
- I (24)
- J (44)
- K (906)
- L (312)
- M (495)
- N (1655)
- Ň (2)
- O (1106)
- P (5683)
- R (1196)
- Ř (51)
- S (1155)
- Š (83)
- T (168)
- Ŧ (7)
- U (1288)
- V (2716)
- X (1)
- Z (1825)
- Ž (43)

- hmoždit se<sub>1</sub>
- hnát<sub>1</sub>
- hnát<sub>n2</sub>
- hnát<sub>n4</sub>
- hnát se<sub>1</sub>
- hněvat se<sub>2</sub>
- hněvat se<sub>n3</sub>
- hnísat<sub>1</sub>
- hnísť<sub>1</sub>
- hnísť<sub>2</sub>
- hnít<sub>1</sub>
- hnoint<sub>1</sub>
- hnout<sub>1</sub>
- hnout<sub>2</sub>
- hnout se<sub>n2</sub>
- **hodit**<sub>1</sub>
- hodit<sub>1</sub>
- hodit se<sub>1</sub>
- hodit se<sub>2</sub>
- hodit se<sub>3</sub>
- hodit se<sub>4</sub>
- hodit se<sub>n5</sub>
- hodnotit<sub>2</sub>
- hodnotit<sub>n2</sub>

# hodit<sup>pf</sup><sub>1</sub> metat<sup>impf</sup><sub>1</sub>

## házet<sup>impf</sup><sub>1</sub>

**definition:** prudkým pohybem uvést do letu určitým směrem

**class:** throw-17.1

**passive:** yes

**[1]** házet<sub>1</sub>, hodit<sub>1</sub>, metat<sub>1</sub> ~

-frame: **AG**<person:1><sub>kdo1</sub> **VERB**<sup>obl</sup> **OBJ**<object:1><sub>co4, čím7</sub> **PAT**<person:>

-example: husité metali kameny na nepřitele (impf)

-example: žárlivá manželka metá po manželovi vázou (impf)

-synonym:

-use: prim

-reflexivity: no

**[2]** házet<sub>1</sub>, hodit<sub>1</sub>, metat<sub>1</sub> ~

-frame: **AG**<person:1><sub>kdo1</sub> **VERB**<sup>obl</sup> **OBJ**<object:1><sub>co4</sub>

-example: chlapec hodil sněhovou kouli (pf)

-example: lovec metá oštěp (impf)

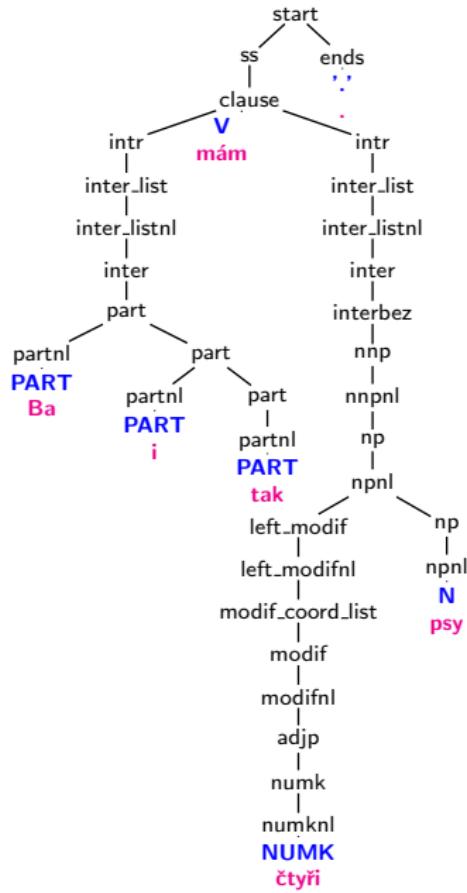
-synonym:

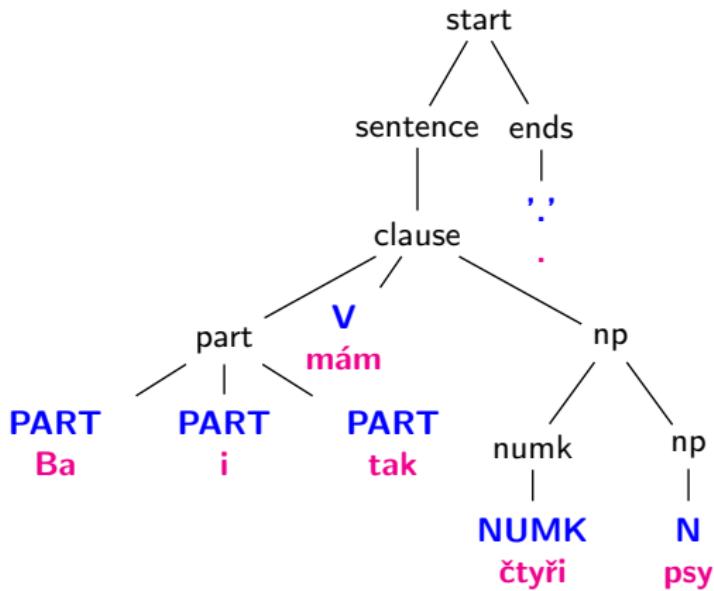
-use: prim

-reflexivity: no

**[3]** házet<sub>1</sub>, hodit<sub>1</sub>, metat<sub>1</sub> ~

-frame: **AG**<person:1><sub>kdo1</sub> **VERB**<sup>obl</sup> **OBJ**<object:1><sub>co4</sub> **LOC**<location:1><sub>do:</sub>





◀ Zpět

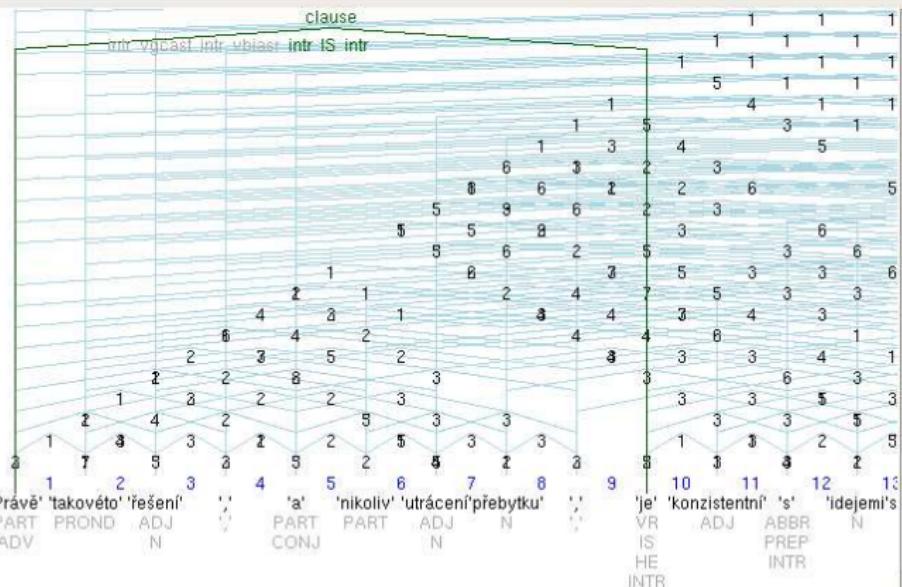
File Select Sort View Closed Ranges Help

< > 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 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

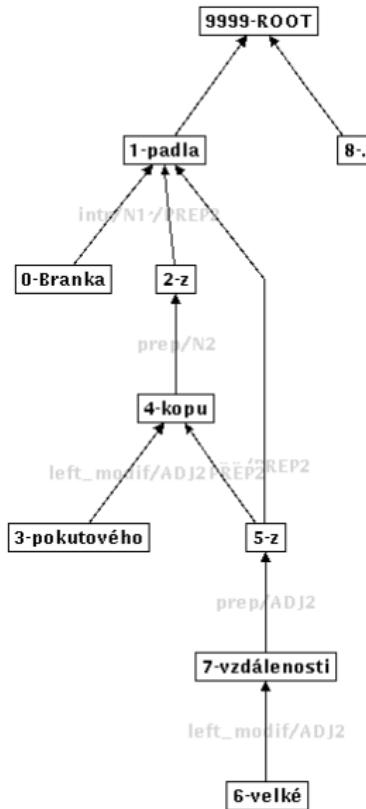
<- >- Fix Edge

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



INFO: Closed edges ranges displayed.

*Branka padla z pokutového kopu z velké vzdálenosti.*



**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

<b>slovo</b>	<b>před</b>	<b>po</b>
Na	k7{c4, c6}	k7c6
krásné	k2eA{gFnPcl1, gFnPc4d1, gFnPc5d1, gFnSc2d1, gFnSc3d1, gFnSc6d1, gInPcl1, gInPc4d1, gInPc5d1, gInScl1wH, gInSc4d1wH, gInSc5d1wH, gMnPc4d1, gMnScl1wH, gMnSc5d1wH, gNnScl1, gNnSc4d1, gNnSc5d1}	k2eAgFnSc6d1
dłouhé	k2eA{gFnPcl1, gFnPc4d1, gFnPc5d1, gFnSc2d1, gFnSc3d1, gFnSc6d1, gInPcl1, gInPc4d1, gInPc5d1, gInScl1wH, gInSc4d1wH, gInSc5d1wH, gMnPc4d1, gMnScl1wH, gMnSc5d1wH, gNnScl1, gNnSc4d1, gNnSc5d1}	k2eAgFnSc6d1
ulici	kIgFnSc3, kIgFnSc4, kIgFnSc6	kIgFnSc6
stálo	k5eAalmAgNnSalrD	kSeApNnStMmPal
moderní	k2eA{gFnPcl1, gFnPc4d1, gFnPc5d1, gFnScl1, gFnSc2d1, gFnSc3d1, gFnSc4d1, gFnSc5d1, gFnSc6d1, gFnSc7d1, gInPcl1, gInPc4d1, gInPc5d1, gInScl1, gInSc4d1, gInSc5d1, gMnPcl1, gMnPc4d1, gMnPc5d1, gMnScl1, gMnSc5d1, gNnPcl1, gNnPc4d1, gNnPc5d1, gNnScl1, gNnSc4d1, gNnSc5d1}	k2eAgNnScl1, 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	kIgNnScl, kIgNnSc4, kIgNnSc5	kIgNnScl, kIgNnSc4, kIgNnSc5

Sémantická síť po vstupu 3 vět: *Toto je jablko.*, *Tato kostka je červená.* a *To jablko je červené.*

