

1 Implementace gramatiky

1.1 Proč implementovat gramatiku?

- praktické, komerční účely
- ověřování hypotéz o jazyce

Cílem je obvykle najít:

- abstraktní reprezentaci daných jazykových výrazů (analýza), nebo
- jazykové výrazy k dané abstraktní reprezentaci (syntéza, generování)

Každopádně hledáme takové objekty našeho modelu, které splňují zadání. Prakticky ale nelze nejdříve vytvořit všechny možné objekty, které gramatika připouští, a pak mezi nimi hledat.

1.2 MoMo

Výukový program, na základě gramatiky kontroluje správnost objektů, které uživatel musí sám zkonstruovat.

- <http://www.sfs.uni-tuebingen.de/~fr/teaching/ss04/>
- vhodné pro seznámení s formalismem, nejbližší teoretické podobě HPSG, ale nevytváří struktury, pouze kontroluje jejich správnost podle gramatiky. Je třeba nejprve „namalovat“ graf (*interpretaci*), jehož správnost se ověřuje vzhledem k signatuře a teorii. Pro rozsáhlejší struktury pomalé.
- Gramatika 0 s několika interpretacemi, k otevření souboru je třeba si MoMo nainstalovat nebo zkusit webovou verzi: <http://utkl.ff.cuni.cz/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/gram0a.mmp>.

Interpretace je možné si prohlédnout i na obrázcích, bez instalace MoMo:

- slovo *pase*:
<http://utkl.ff.cuni.cz/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase.jpg>
- věta *pepa pase*:
<http://utkl.ff.cuni.cz/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pepapase.jpg>
- slovo *pase*, Momo hlásí: Signature check failed. The feature-structure is not total. Attributes (CAT) are missing from the node number 0.
http://utkl.ff.cuni.cz/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase_bad_signature1.jpg
- slovo *pase*, Momo hlásí: Signature check failed. Feature structure check failed. The node number CAT belong to more than one substructures with the root nodes.
http://utkl.ff.cuni.cz/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase_bad_signature2.jpg
- slovo *pase*, Momo hlásí: Signature check failed. The feature-structure is not well-typed. The attribute DTRS of the node number 0 has an incorrect value. The feature-structure is not total. Attributes (CAT) are missing from the node number 0.
http://utkl.ff.cuni.cz/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase_bad_signature3.jpg

- slovo *pase*, Momo hlásí: Signature check was successful but satisfaction check failed. (*pase* je zde jako *noun*)
http://utkl.ff.cuni.cz/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase_bad_description.jpg
- věta *Pepa pase*, MoMo hlásí: Satisfaction failed (místo *pase* je zde *e_list*)
http://utkl.ff.cuni.cz/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pepapase_satisfaction_failed.jpg
- věta *Pepa pase*, MoMo hlásí: Modelling failed (*pase* je tady jako *noun*)
http://utkl.ff.cuni.cz/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pepapase_modelling_failed.jpg

1.3 Trale

Parser: na vstupu je gramatika a řetězec slov, k nimž se co nejefektivnějším způsobem hledá odpovídající *popis* objektů, které splňují omezení daná oběma vstupy. Z výsledku nás obvykle zajímá hlavně reprezentace významu.

Generátor: na vstupu je gramatika a reprezentace významu, k nimž se co nejefektivnějším způsobem hledá odpovídající *popis* objektů, které splňují omezení daná oběma vstupy. Z výsledku nás obvykle zajímá hlavně řetězec slov.

Trale:

- je parser (a generátor), který umožňuje implementovat gramatiku HPSG bez větších úprav
- nedokáže však pracovat s každou teoretickou gramatikou HPSG bez úprav
- představuje kompromis mezi efektivitou výpočtu a věrností lingvistice
- ukázky gramatik v systému Trale viz webové stránky kursu

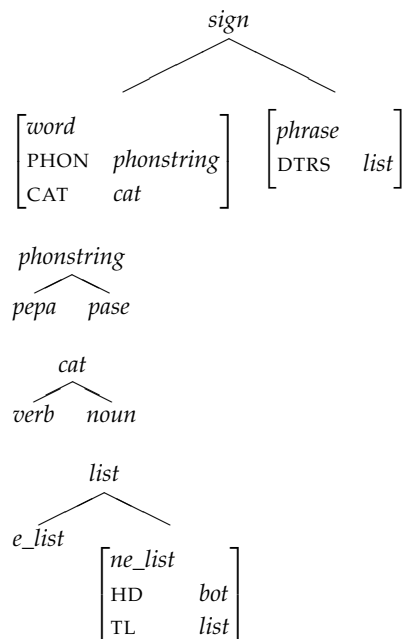
1.3.1 Co je v Trale-u jinak?

- složková (bezkontextová) pravidla
- technické specifikace (názvy souborů)
- pokyny pro zobrazení struktur
- prostředky pro úspornější zápis gramatiky (slovník, makra, seznamy)

1.3.2 Ukázky gramatiky ve formátu Trale

lze stáhnout z adresy <http://utkl.ff.cuni.cz/~rosen/VYUKA/DEKL/GRAMATIKY07/>

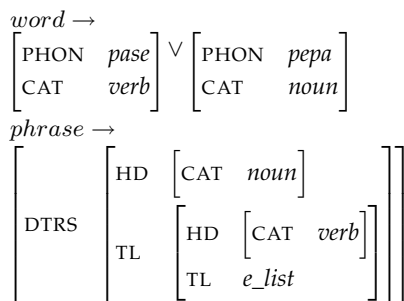
(1) *Papírová signatura gramatiky 0*



(2) *Signatura gramatiky 0 ve formátu Trale*

```
type_hierarchy
bot
  sign
    phrase dtrs:list
    word phonology:phonstring cat:cat
  phonstring
    pepa
    pase
  cat
    noun
    verb
  list
    ne_list hd:bot tl:list
    e_list
.
```

(3) *Papírová teorie gramatiky 0*



(4) *Teorie gramatiky 0 ve formátu Trale, bez složkových pravidel a bez využití mechanismu pro slovníková hesla*

```
% Lexicon
word *> phonology:pepa,
```

```
cat:noun;
phonology:pase,
cat:verb.
```

```
phrase *> dtrs:[(cat:noun),
               (cat:verb)].
```

(5) *Teorie gramatiky 0, verze 0a ve formátu Trale, bez využití mechanismu pro hodnoty atributů PHON. Signatura zůstává stejná jako u původní verze.*

```
% Lexicon
pepa ---> (word,
          phonology:pepa,
          cat:noun).
pase ---> (word,
          phonology:pase,
          cat:verb).
```

```
% Phrase Structure Rule
npvp rule
  (phrase,
   dtrs:[Subj,Head])
====>
  cat> (Subj,
        cat:noun),
  cat> (Head,
        cat:verb).
```

(6) *Signatura gramatiky 0, verze 0b ve formátu Trale*

```
type_hierarchy
bot
  sign phon:list
  phrase dtrs:list
  word cat:cat
  cat
    noun
    verb
  list
    ne_list hd:bot tl:list
    e_list
.
```

(7) *Teorie gramatiky 0, verze 0b ve formátu Trale, bez využití mechanismu pro hodnoty atributů PHON, s implicitně definovanými hodnotami phonstring*

```
% slovník
pepa ---> (word,
          phon:[(a_ pepa)],
          cat:noun).
pase ---> (word,
          phon:[(a_ pase)],
          cat:verb).
```

```
% pravidla
nv rule (phrase,
        phon:[SubjPhon,HeadPhon],
        dtrs:[Subj,Head]) ====>
  cat> (Subj,
        word,
        phon:[SubjPhon],
        cat:noun),
  cat> (Head,
        word,
```

```
phon: [HeadPhon],
cat: verb).
```

- (8) Teorie gramatiky 0, verze 0c ve formátu Trale, s využitím mechanismu pro hodnoty atributů PHON, s implicitně definovanými hodnotami *phonstring*. Signatura zůstává stejná jako u verze 0b.

```
:- tree_extensions.
:- multifile if/2.

% slovník
pepa ~~> (word, cat:noun).
pase ~~> (word, cat:verb).

% pravidla
nv ##
    phrase
===>
    cat> (word, cat:noun),
    cat> (word, cat:verb).
```

1.3.3 Instalace

- instalační soubory a dokumentace: <http://www.ale.cs.toronto.edu/docs/>, k této verzi je třeba mít nainstalovaný Sicstus Prolog
- verze zkompileovaná pro Linux, Sicstus Prolog není třeba: <http://www.cl.uni-bremen.de/Software/Trale/>
- verze na bootovacím CD, lze spustit na každém PC, které umožňuje zavedení systému z CD mechaniky (lze nastavit v BIOSu): <http://www.cl.uni-bremen.de/Software/Grammix/>

1.3.4 Spuštění systému Trale

1. umístěte svou gramatiku (soubory *signature* a *theory*) do jednoho adresáře
2. v příkazové řádce se přesuňte do tohoto adresáře (příkazem *cd*)
3. `$ trale -gsu`
– tímto příkazem spustíte Trale, význam parametrů můžete zjistit příkazem `trale -help`
4. `| ?-`
– na příkazové řádce se objeví prologovský prompt a otevře se okénko grafického rozhraní GRALE

1.3.5 Kompilace gramatiky

1. `| ?- c.`
– tímto příkazem gramatiku zkompilejte (za písmenem „c“ musí být tečka)
2. kompilátor odhalí případné syntaktické chyby v gramatice (soubory *signature* a *theory*), ty je třeba opravit a gramatiku znovu zkompileovat

1.3.6 Testování gramatiky, syntaktická analýza příkladů

– je možné jen s úspěšně zkompilevanou gramatikou

- `| ?- rec.`
– tak otevřete další řádek, který začíná promptem `| :`
– na něj můžete napsat slovo, výraz nebo větu, které chcete analyzovat, tedy např. `| : děti spí ;` pak stisknete klávesu Enter (bez tečky)

- je-li zadán řetězec slov podle Vaší gramatiky v pořádku, objeví se na dalších řádcích např. toto:

```
STRING:
0 děti 1 spí 2
```

```
ANOTHER?
```

v okně GRALE se objeví řádek

(FS) děti spí

a v novém okně nazvaném podle řetězce děti spí se objeví výsledná struktura

- v okně GRALE je možné si dvojitým kliknutím na příslušný řádek znovu vyvolat předtím zavřená okna se strukturami
- v okně se strukturou je možné přepínat mezi zobrazením stromu a sestavy rysů, kliknutím na proměnnou lze získat její hodnotu a dalším kliknutím ji zase skrýt, dvojitým kliknutím na atribut lze skrýt i atribut
- na dotaz ANOTHER? lze reagovat stisknutím klávesy Enter nebo N, v tom případě se vrátíme na příkazovou řádku bez možnosti prozkoumat případné další výsledky, ty získáme po stisknutí klávesy Y

1.3.7 Další možnosti, jak testovat gramatiku:

- Výpis pravidla (podle jména) a slovníkového hesla (podle slova):
`| ?- rule Rulename.`
`| ?- lex Word.`
- Výpis části hierarchie typů:
`| ?- show_approp (Type) .`
`| ?- show_subtypes (Type) .`
`| ?- show_all_subtypes (Type) .`
`| ?- show_supertypes (Type) .`
`| ?- show_all_supertypes (Type) .`
- Syntaktická analýza příkladů podle čísel:
`| ?- test (Nr) .`
`| ?- test ([From, To]) .`
`| ?- test (all) .`

1.3.8 Jak Trale ukončit

```
| ?- halt.
```