

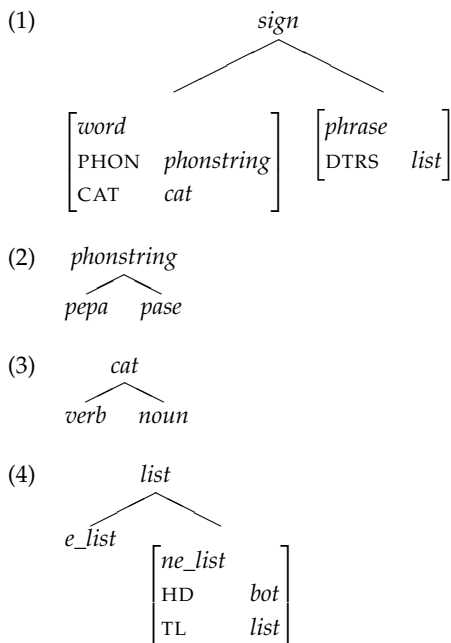
## Pokus o nápovědu

Některé úkoly byly napoprvé poněkud náročné. Tak to zkusím trochu ocukrovat.

Následující gramatika je trochu jednodušší než ta minulá:

## 1 Gramatika 0

### 1.1 Signatura

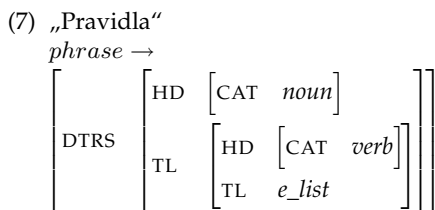
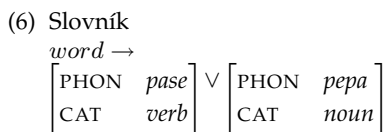


(5) „Computer-friendly“ zápis signatury (1–4):

```

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

### 1.2 Teorie



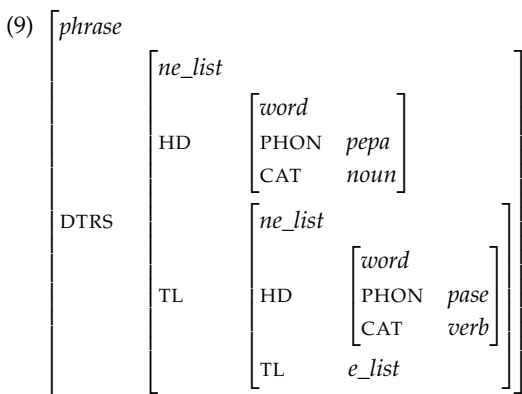
(8) „Computer-friendly“ zápis teorie (6–7):

```

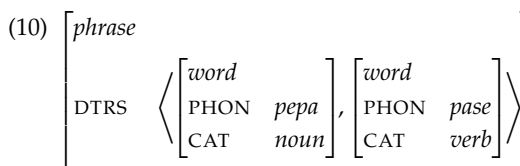
word * > phon:pepa,cat:noun;
        phon:pase,cat:verb.
phrase * > dtrs:[(cat:noun),(cat:verb)].
    
```

### 1.3 Interpretace

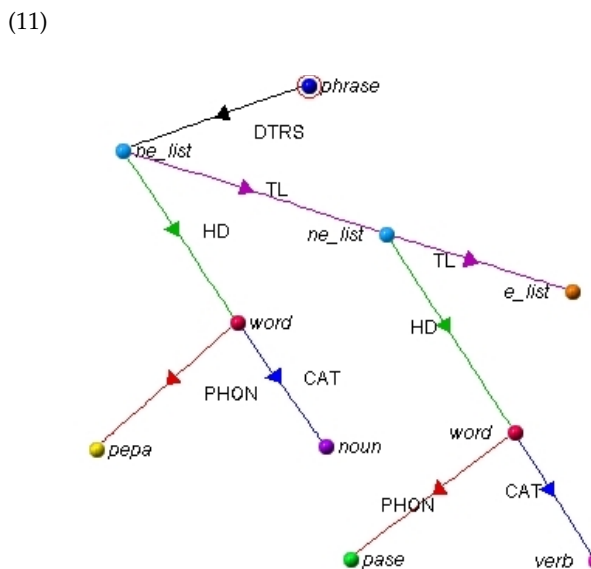
Tato gramatika připouští (kromě slov *pepa* a *pase*) jedinou větu, která odpovídá následující struktuře:



Totéž, ale trochu přehledněji, s prvky seznamu uzavřenými v loměných závorkách:



A ještě jednou totéž, tentokrát v podobě orientovaného acyklického grafu (DAG). Uzly představují typy, hrany představují atributy.



### 1.4 Softwarové nástroje

Tato část je určena jen pro vážné zájemce. Ostatní ji mohou přeskóčit.

### 1.4.1 MoMo

- <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://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/gram0a.mmp>.
- Interpretace je možné si prohlédnout i na obrázcích, bez instalace MoMo:
  - slovo *pase*:  
<http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase.jpg>
  - věta *pepa pase*:  
<http://jakobson/~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://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase\\_bad\\_signature1.jpg](http://jakobson/~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://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase\\_bad\\_signature2.jpg](http://jakobson/~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://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase\\_bad\\_signature3.jpg](http://jakobson/~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://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase\\_bad\\_description.jpg](http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pase_bad_description.jpg)
  - věta *Pepa pase*, MoMo hlásí: Satisfaction failed (místo *pase* je zde *e\_list*)  
[http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pepapase\\_satisfaction\\_failed.jpg](http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pepapase_satisfaction_failed.jpg)
  - věta *Pepa pase*, MoMo hlásí: Modelling failed (*pase* je tedy jako *noun*)  
[http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pepapase\\_modelling\\_failed.jpg](http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/MOMO/pepapase_modelling_failed.jpg)

### 1.4.2 ConTroll

- <http://www.sfs.uni-tuebingen.de/controll/>, k instalaci je třeba Sicstus Prolog nebo SWI Prolog

- *constraint solver*, po zadání dotazu hledá odpověď podle gramatiky, teoreticky lze použít pro analýzu i generování, gramatiku HPSG lze implementovat téměř úprav, vhodné pro testování menších gramatik, u složitějších gramatik však nastávají potíže s postupem výpočtu, částečně řešitelné pomocí *delay statements*.
- Gramatika 0 (soubory *signature* a *theory*) a ukázky výstupů viz <http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/ConTroll/>

### 1.4.3 Trale

- <http://www.ale.cs.toronto.edu/docs/>, k instalaci je třeba Sicstus Prolog
- parser, umožňuje implementovat gramatiku HPSG bez větších úprav
- Gramatika 0 (soubory *signature* a *theory*) a ukázky výstupů viz <http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0a/TRALE/>
- další verze téže gramatiky viz <http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0b/> a <http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM0c/>

## 2 Gramatika 1

viz <http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM1/>

## 3 Úkoly – nápověda

1. Uveďte příklady „vět“:
  - (a) které gramatika připouští a které jsou správně český
  - (b) které gramatika připouští a které nejsou správně český
  - (c) které gramatika nepřipouští

– tedy snad nápověda není třeba
2. Upravte gramatiku tak, aby připouštěla pouze slovosled podmět-sloveso-předmět.
3. Upravte gramatiku tak, aby brala v úvahu pády.
  - jednu z možností, jak vyřešit obojí, ukazuje Gramatika 1a, viz <http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM1a/>, příslušné soubory *signature* a *theory* jsou ve formátu ConTroll: <http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM1a/ConTroll/signature> a <http://jakobson/~rosen/VYUKA/DEKL/GRAMATIKY/GRAM1a/ConTroll/theory>, uvedené níže.
4. Zkuste gramatiku vylepšit podle vlastní úvahy.

## 4 Gramatika 1a

(12) signatura

```
type_hierarchy
bot
  sign phon:list synsem:synsem
  word
  phrase dtrs:dtrs
  synsem
```

```

    verb
    noun case:case
    vp
    s
case
    nom
    acc
dtrs hd_dtr:sign nhd_dtr:sign
string
.

(13) teorie

% LEXICON

word ==> (phon:[pase],synsem:verb);
         (phon:[pepa],synsem:(noun,case:nom));
         (phon:[kozu],synsem:(noun,case:acc)).

% RULE

phrase ==> (phon:append(First,Second),
            ((synsem:s,
              dtrs:(hd_dtr:(phon:Second,
                          synsem:vp),
                    nhd_dtr:(phon:First,
                              synsem:(noun,
                                      case:nom))))
            ;
            (synsem:vp,
              dtrs:(hd_dtr:(phon:First,
                          synsem:verb),
                    nhd_dtr:(phon:Second,
                              synsem:(noun,
                                      case:acc)))))).

% RELATION

%   declaration of types in the relation

append(list,list) **> list.

%   definition of the relation

append([],L) := L.
append([H|T1],L2) := [H|append(T1,L2)].

```