PATR, část 3

Gramatika 3

jako gramatika 2, navíc se “sémantikou”
pravidla zůstávají stejná, jen lexikální kategorie se rozšíří o další atribut
tento atribut musí být pod atributem head

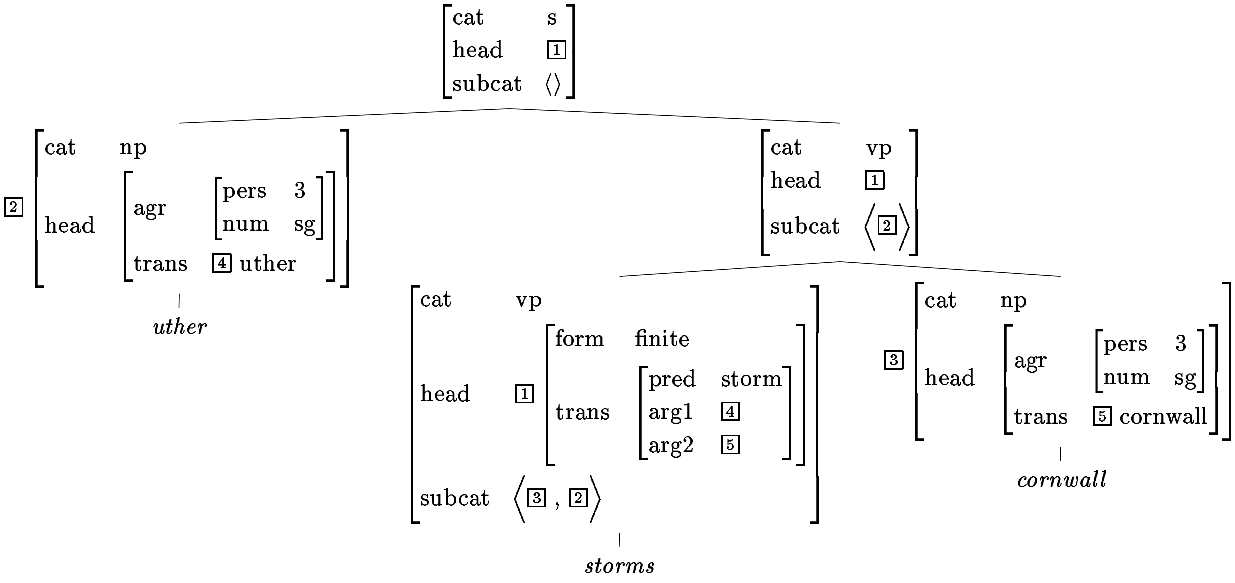
(1) Jednoduchá “sémantická” reprezentace věty *Uther storms Cornwall*

$$\begin{bmatrix} \text{pred} & \text{storm} \\ \text{arg1} & \text{uther} \\ \text{arg2} & \text{cornwall} \end{bmatrix}$$

(2) Uther ↦ $\begin{bmatrix} \text{cat} & \text{np} \\ \text{head} & \begin{bmatrix} \text{agr} & \begin{bmatrix} \text{pers} & 3 \\ \text{num} & \text{sg} \end{bmatrix} \\ \text{trans} & \text{uther} \end{bmatrix} \end{bmatrix}$

(3) *storms* ↦ $\begin{bmatrix} \text{cat} & \text{v} \\ \text{head} & \begin{bmatrix} \text{form} & \text{finite} \\ \text{trans} & \begin{bmatrix} \text{pred} & \text{storm} \\ \text{arg1} & \boxed{1} \\ \text{arg2} & \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{subcat} & \left\langle \begin{bmatrix} \text{cat} & \text{np} \\ \text{head} & \begin{bmatrix} \text{trans} & \boxed{2} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \text{cat} & \text{np} \\ \text{head} & \begin{bmatrix} \text{agr} & \begin{bmatrix} \text{num} & \text{sg} \\ \text{per} & 3 \end{bmatrix} \\ \text{trans} & \boxed{1} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$

(4) Syntaktický strom věty *Uther storms Cornwall*



(5) Jednoduchá “sémantická” reprezentace věty *Uther persuades Arthur to storm Cornwall*

$$\begin{bmatrix} \text{pred} & \text{persuade} \\ \text{arg1} & \text{uther} \\ \text{arg2} & \boxed{1}\ \text{arthur} \\ \text{arg3} & \begin{bmatrix} \text{pred} & \text{storm} \\ \text{arg1} & \boxed{1} \\ \text{arg2} & \text{cornwall} \end{bmatrix} \end{bmatrix}$$

(6) *persuades* ↦

$$\begin{bmatrix} \text{cat} & \text{v} \\ \text{head} & \begin{bmatrix} \text{form} & \text{finite} \\ \text{trans} & \begin{bmatrix} \text{pred} & \text{persuade} \\ \text{arg1} & \boxed{1} \\ \text{arg2} & \boxed{2} \\ \text{arg3} & \boxed{3} \end{bmatrix} \end{bmatrix} \\ \text{subcat} & \left\langle \boxed{4} \begin{bmatrix} \text{cat} & \text{np} \\ \text{head} & \begin{bmatrix} \text{trans} & \boxed{2} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \text{cat} & \text{vp} \\ \text{head} & \begin{bmatrix} \text{form} & \text{inf} \\ \text{trans} & \boxed{3} \end{bmatrix} \\ \text{subcat} & \left\langle \boxed{4} \right\rangle \end{bmatrix}, \begin{bmatrix} \text{cat} & \text{np} \\ \text{head} & \begin{bmatrix} \text{agr} & \begin{bmatrix} \text{num} & \text{sg} \\ \text{per} & 3 \end{bmatrix} \\ \text{trans} & \boxed{1} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

(7) *to* ↦

$$\begin{bmatrix} \text{cat} & \text{v} \\ \text{head} & \begin{bmatrix} \text{form} & \text{inf} \\ \text{trans} & \boxed{2} \end{bmatrix} \\ \text{subcat} & \left\langle \begin{bmatrix} \text{cat} & \text{vp} \\ \text{head} & \begin{bmatrix} \text{form} & \text{nonfin} \\ \text{trans} & \boxed{2} \end{bmatrix} \\ \text{subcat} & \left\langle \boxed{1} \right\rangle \end{bmatrix}, \boxed{1} \begin{bmatrix} \text{cat} & \text{np} \end{bmatrix} \right\rangle \end{bmatrix}$$

Úkoly

1. Zkompilujte gramatiku 3 a vyzkoušejte ji na všech příkladech. Všimněte si, jaký je výsledek u vět s pomocným slovesem *have*. Např. u věty *Knights have stormed Cornwall* vypadá sémantická representace takto:[1]

   $$\begin{bmatrix} \text{pred} & \text{perfective\_} \\ \text{arg1} & \begin{bmatrix} \text{pred} & \text{storm\_} \\ \text{arg1} & \text{knights\_} \\ \text{arg2} & \text{cornwall\_} \end{bmatrix} \end{bmatrix}$$

   Zkuste upravit gramatiku tak, aby sémantická representace stejné věty vypadala takto:[2]

   $$\begin{bmatrix} \text{pred} & \text{storm\_} \\ \text{aspect} & \text{perfective} \\ \text{tense} & \text{present} \\ \text{arg1} & \text{knights\_} \\ \text{arg2} & \text{cornwall\_} \end{bmatrix}$$

[1] Na výstupu z QPATRu jsou atributy uvedeny v abecedním pořadí.

[2] Není třeba upravovat pravidla gramatiky, stačí upravit slovníkové heslo pro *have* tak, aby zacházelo se sémantickou reprezentací komplementu podobně jako ‘sloveso’ *to*.

Dále zkuste pozměnit sémantickou reprezentaci substantiv:

$$\begin{bmatrix} \text{pred} & \text{storm\_} \\ \text{aspect} & \text{perfective} \\ \text{tense} & \text{present} \\ \text{arg1} & \begin{bmatrix} \text{pred} & \text{knight\_} \\ \text{number} & \text{plural} \end{bmatrix} \\ \text{arg2} & \begin{bmatrix} \text{pred} & \text{cornwall\_} \\ \text{number} & \text{singular} \end{bmatrix} \end{bmatrix}$$

Nakonec se postarejte o slovesa v prostém přítomném času:

$$\begin{bmatrix} \text{pred} & \text{sleep\_} \\ \text{aspect} & \text{imperfective} \\ \text{tense} & \text{present} \\ \text{arg1} & \begin{bmatrix} \text{pred} & \text{uther\_} \\ \text{number} & \text{singular} \end{bmatrix} \end{bmatrix}$$

2. Rozšiřte gramatiku tak, aby analyzovala věty *Uther sleeps often* a Uther often sleeps a dávala takovouto sémantickou reprezentaci:

   $$\begin{bmatrix} \text{pred} & \text{sleep\_} \\ \text{aspect} & \text{imperfective} \\ \text{tense} & \text{present} \\ \text{modifier} & \begin{bmatrix} \text{pred} & \text{often\_} \end{bmatrix} \\ \text{arg1} & \begin{bmatrix} \text{pred} & \text{uther\_} \\ \text{number} & \text{singular} \end{bmatrix} \end{bmatrix}$$

   Vyzkoušejte, zda gramatika zvládne i další věty s adverbiem na různých pozicích. Nastavte QPATR tak, aby zobrazoval všechny výsledky. Vyzkoušejte např. věty:
   Uther storms Cornwall often.
   Uther has often stormed Cornwall.
   Knights have stormed Cornwall often.
   Uther persuades knights to sleep often.
   Uther persuades knights to storm cornwall often.
   Uther often persuades knights to sleep.
   Uther often persuades knights to sleep often.

   Nejste-li spokojeni s výsledkem, zkuste najít chybu a navrhnout řešení. Alespoň u poslední věty však řešení vyžaduje podstatný zásah do gramatiky.

3. V předchozím úkolu bylo možné rozvinout sloveso jen jedním adverbiem. Toto omezení lze překonat třeba tak, že zavedeme sémantickou reprezentaci, ve které je klauze argumentem adverbia:

   $$\begin{bmatrix} \text{pred} & \text{often\_} \\ \text{arg} & \begin{bmatrix} \text{pred} & \text{sleep\_} \\ \text{aspect} & \text{imperfective} \\ \text{tense} & \text{present} \\ \text{arg1} & \begin{bmatrix} \text{pred} & \text{uther\_} \\ \text{number} & \text{singular} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

   Zkuste v gramatice provést všechny potřebné změny, vyzkoušejte ji na všech větách z předchozího úkolu a výsledky porovnejte.

Příloha 1: Gramatika 3 v kódu QPATR

```
/********************************************************************/
/*  SHIEBER3.GRM                                                    */
/*                                                                  */
/*  demonstration grammar three (pp. 76-82) in QPATR syntax         */
/*     subject-verb agreement                                       */
/*     complex subcategorization                                    */
/*     logical form construction                                    */
/*                                                                  */
/*  Stuart M. Shieber, An Introduction to Unification-Based         */
/*  Approaches to Grammar. Stanford, 1986.                          */
/********************************************************************/

% grammar rules *****************************************************

%   sentence formation
1 # s(S) ---> np(NP), vp(VP) :: S/head *= VP/head,
                                S/head/form *= finite,
                                VP/syncat/first *= NP,
                                VP/syncat/rest *= end.

%   trivial verb phrase
2 # vp(VP) ---> v(V) :: VP/head *= V/head,
                        VP/syncat *= V/syncat.

%   complements
3 # vp(VP_1) ---> vp(VP_2), xp(XP) ::
                     VP_1/head *= VP_2/head,
                     VP_2/syncat/first *= XP,
                     VP_2/syncat/rest  *= VP_1/syncat.

% lexicon ***********************************************************

uther     lex  np(F) ::
               F/head/agreement/gender *= masculine,
               F/head/agreement/person *= third,
               F/head/agreement/number *= singular,
               F/head/trans *= uther_.

cornwall  lex  np(F) ::
               F/head/agreement/gender *= masculine,
               F/head/agreement/person *= third,
               F/head/agreement/number *= singular,
               F/head/trans *= cornwall_.

knights   lex  np(F) ::
               F/head/agreement/gender *= masculine,
               F/head/agreement/person *= third,
               F/head/agreement/number *= plural,
               F/head/trans *= knights_.

%--------------------------------------------------------------------

sleeps    lex  v(F) ::
```

```
               F/head/form *= finite,
               F/syncat/first/cat *= np,
               F/syncat/first/head/agreement/person *= third,
               F/syncat/first/head/agreement/number *= singular,
               F/syncat/rest *= end,
               F/head/trans/pred *= sleep_,
               F/head/trans/arg1 *= F/syncat/first/head/trans.

sleep     lex  v(F) ::
               F/head/form *= finite,
               F/syncat/first/cat *= np,
               F/syncat/first/head/agreement/number *= plural,
               F/syncat/rest *= end,
               F/head/trans/pred *= sleep_,
               F/head/trans/arg1 *= F/syncat/first/head/trans.

sleep     lex  v(F) ::
               F/head/form *= nonfinite,
               F/syncat/first/cat *= np,
               F/syncat/rest *= end,
               F/head/trans/pred *= sleep_,
               F/head/trans/arg1 *= F/syncat/first/head/trans.

%--------------------------------------------------------------------

storms    lex  v(F) ::
               F/head/form *= finite,
               F/syncat/first/cat *= np,
               F/syncat/rest/first/cat *= np,
               F/syncat/rest/first/head/agreement/person *= third,
               F/syncat/rest/first/head/agreement/number *= singular,
               F/syncat/rest/rest *= end,
               F/head/trans/pred *= storm_,
               F/head/trans/arg1 *= F/syncat/rest/first/head/trans,
               F/head/trans/arg2 *= F/syncat/first/head/trans.

stormed   lex  v(F) ::
               F/head/form *= pastparticiple,
               F/syncat/first/cat *= np,
               F/syncat/rest/first/cat *= np,
               F/syncat/rest/rest *= end,
               F/head/trans/pred *= storm_,
               F/head/trans/arg1 *= F/syncat/rest/first/head/trans,
               F/head/trans/arg2 *= F/syncat/first/head/trans.

storm     lex  v(F) ::
               F/head/form *= nonfinite,
               F/syncat/first/cat *= np,
               F/syncat/rest/first/cat *= np,
               F/syncat/rest/rest *= end,
               F/head/trans/pred *= storm_,
               F/head/trans/arg1 *= F/syncat/rest/first/head/trans,
               F/head/trans/arg2 *= F/syncat/first/head/trans.

%--------------------------------------------------------------------
```

```
has       lex v(F) ::
              F/head/form *= finite,
              F/syncat/first/cat *= vp,
              F/syncat/first/head/form *= pastparticiple,
              F/syncat/first/syncat/rest *= end,
              F/syncat/first/syncat/first *= F/syncat/rest/first,
              F/syncat/rest/first/cat *= np,
              F/syncat/rest/first/head/agreement/number *= singular,
              F/syncat/rest/first/head/agreement/person *= third,
              F/syncat/rest/rest *= end,
              F/head/trans/pred *= perfective_,
              F/head/trans/arg1 *= F/syncat/first/head/trans.

have      lex v(F) ::
              F/head/form *= finite,
              F/syncat/first/cat *= vp,
              F/syncat/first/head/form *= pastparticiple,
              F/syncat/first/syncat/rest *= end,
              F/syncat/first/syncat/first *= F/syncat/rest/first,
              F/syncat/rest/first/cat *= np,
              F/syncat/rest/first/head/agreement/number *= plural,
              F/syncat/rest/rest *= end,
              F/head/trans/pred *= perfective_,
              F/head/trans/arg1 *= F/syncat/first/head/trans.

%--------------------------------------------------------------------

persuades lex v(F) ::
              F/head/form *= finite,
              F/syncat/first/cat *= np,
              F/syncat/rest/first/cat *= vp,
              F/syncat/rest/first/head/form *= infinitival,
              F/syncat/rest/first/syncat/rest *= end,
              F/syncat/rest/first/syncat/first *= F/syncat/first,
              F/syncat/rest/rest/first/cat *= np,
              F/syncat/rest/rest/first/head/agreement/number *= singular,
              F/syncat/rest/rest/first/head/agreement/person *= third,
              F/syncat/rest/rest/rest *= end,
              F/head/trans/pred *= persuade_,
              F/head/trans/arg1 *= F/syncat/rest/rest/first/head/trans,
              F/head/trans/arg2 *= F/syncat/first/head/trans,
              F/head/trans/arg3 *= F/syncat/rest/first/head/trans.

%--------------------------------------------------------------------

to        lex v(F) ::
              F/head/form *= infinitival,
              F/syncat/first/cat *= vp,
              F/syncat/first/head/form *= nonfinite,
              F/syncat/first/syncat/rest *= end,
              F/syncat/first/syncat/first *= F/syncat/rest/first,
              F/syncat/rest/first/cat *= np,
              F/syncat/rest/rest *= end,
              F/head/trans *= F/syncat/first/head/trans.
```

```
% category labels ***************************************************

category_label(F, C) :- F/cat *= C.

s(F)  :- category_label(F, s ).
np(F) :- category_label(F, np).
vp(F) :- category_label(F, vp).
v(F)  :- category_label(F, v ).
xp(F). % dummy label


% logical form ******************************************************

semantic_representation(F, LF) :- F/head/trans *= LF.


% example sentences *************************************************

ex1(1, s, [uther, sleeps]).
ex1(2, s, [knights, sleep]).
ex1(3, s, [uther, storms, cornwall]).
ex1(4, s, [uther, has, stormed, cornwall]).
ex1(5, s, [knights, have, stormed, cornwall]).
ex1(6, s, [uther, persuades, knights, to, sleep]).
ex1(7, s, [uther, persuades, knights, to, storm, cornwall]).
```