– Übung 9 –

1 Generalized Quantifiers

In the following exercises we will assume a uniform representation for all syntactic category symbols, as follows.

1. Every logical form will be represented by two Prolog arguments: a term, representing the intended logical form, plus a list of slot variables. This list is permitted to be empty, but it must always be present.
2. For categories that denote variable binding operators (so far that's just Det and (Wh)NP), we add an extra argument position in the category symbol to hold that variable.

You may assume a lexicon something like the following.

```
%%%___/ Lexicon \_________________________________________

det( few(X,P,Q), X, [P,Q] ) --> [few].
det( some(X,P,Q), X, [P,Q] ) --> [some].
det( every(X,P,Q), X, [P,Q] ) --> [every].
/* etc. */

noun( hero(A), [A] ) --> [hero].
noun( villain(A), [A] ) --> [villain].
/* etc. */

verb( exist(A), [A] ) --> ( [exists] ; [exist] ).
verb( admire(A,B), [B,A] ) --> ( [admires] ; [admire] ).
/* etc. */

%%%______________________________________________________
```

1 Suppose we have the following DCG rule:

```
s( LF, SArgs ) -->
        np( Quant, Var, QArgs ), % few(X,hero(X),Q),X,[Q]
        vp( Reln, RArgs ),       % exist(A),[A]
        {/* missing unifications */}.
```

Fill in the missing unifications.

2 Same again.

```
np( LF, Var, Args ) -->
        det( Quant, QVar, QArgs ),
        nbar( Property, PArgs ),
        {/* missing unifications */}.
```

3 One more time.

```
vp( LF, Args ) -->
        verb( Relation, RArgs ),
        np( Quant, Var, QArgs ),
        {/* missing unifications */}.
```

2 Modification

For all the examples that we will consider, the semantics of modification is just conjunction. That is, the things that are *blue cubes* are just the things that are *blue* and that are also *cubes*. Conjunction in Prolog is, of course, represented by the comma, so we would assign to the string *blue cubes* a logical form like `(blue(A),cube(A))`.

We will consider here just two kinds of modifiers: Adjectives and Prepositional Phrases.

4 Given the following rule for N′, fill in the missing unifications.

```
nbar( LF, Args ) -->
        adj( AdjProperty, AdjArgs ),
        nbar( N1Property, N1Args ),
        { /* ... */ }.
```

5 Now write a DCG rule corresponding to the CFG rule $N' \rightarrow N'\ PP$.

6 One more missing unifications problem.

```
pp( LF, Args ) -->
        prep( Relation, RArgs ),
        np( Quant, QVar, QArgs ),
        { /* ... */ }.
```

7 What logical forms do we derive for the following strings? (Assume that *a* is synomymous with *some*, that is, they get the same logical forms.)

1. *a (/some) blue cube*
2. *on a blue cube*
3. *red pyramid*
4. *red pyramid on a blue cube*
5. *every red pyramid on a blue cube*