Prof. O. Nierstrasz
Mohammadreza Hazhirpasand, Joel Niklaus

Solution Applications of Logic Programming

- Exercises are given every week on the PL page of the SCG website (http://scg.unibe.ch/teaching/pl)
- Solutions to each assignment must be sent to **mohammadreza.hazhirpasand@inf.unibe.ch**
- The solutions of the assignments are to be delivered before every Thursday at 11 PM. Solutions handed in later than the specified time will not be accepted. In case of serious reasons send an e-mail to **mohammadreza.hazhirpasand@inf.unibe.ch**

Exercise (6 points)

1. Using the *partition* predicate, write a program to ask for a number and then break a given list into smaller and greater lists (2 pts)

   Note: you need to define a predicate asking for a number whereby the list will be broken into smaller and greater lists. The predicate should be passed as the first argument to the partition predicate (a similar example exists on slide 9).

   Answer:

```
qr(X,Y) :- X>Y.
partition(qr(8),[1,2,3,4,5,6,7,8,9],L,R).
```

2. Create a finite collection of definite clause grammar rules to check whether a sentence is grammatically correct. A sentence can be composed of the following words: (4 pts)

   article a, the

   noun girl, boy

   pronoun that, this

   auxiliary is

   verb sleeps, likes.

   A sentence must be in one of the following forms `subject-predicate` and `subject-predicate-object`.

   - `subject` is formed out of either an `article` and a `noun`, or a `pronoun`. For example, `a girl` or `that`.
   - `predicate` is either an `auxiliary` or a `verb`
   - `object` is formed out of an `article` and a `noun`

   In the `subject-predicate` form of a sentence, the `predicate` can be only `sleeps`. In the `subject-predicate-object` form of a sentence the verb can be either `likes` or `is`. If the `predicate` is `likes`, the `subject` is composed of an `article` and a `noun`. If the

`predicate` is `is`, the `subject` is a `pronoun`.

Write a Prolog question to produce all correct sentences in the grammar. You can check your grammar by phrase(Your-predicate-sentence(),X).

You can test your program with the following examples:

```
this is a sleeps // False
this is a likes // False
the boy likes // False
that boy likes // False
a boy the the girl // False
this is a boy // True
that is the girl // True
the girl likes a girl // True
```

Answer:

```
sentence --> subjectpredicate.
sentence --> subjectpredicateobject.

subjectpredicate --> subject, verb1.
subjectpredicateobject --> article, noun, verb2, object.
subjectpredicateobject --> pronoun, auxiliary, object.

subject --> article, noun.
subject --> pronoun.
predicate --> auxiliary.
predicate --> verb.
object --> article, noun.

article --> [a].
article --> [the].
noun --> [girl].
noun --> [boy].
pronoun --> [that].
pronoun --> [this].
auxiliary --> [is].
verb1 --> [sleeps].
verb2 --> [likes].
```