CSCI 235, Lab Exercise 8, Prolog

Deadline: 11.11.2022 at 11.59PM

Solutions must be submitted into Moodle as a single text file, called **lab08.pl**. Do not use any archiver.
Goal of this exercise is to make you acquainted with the basics of the Prolog language.

Task 1 (Rosetta)

Consider the following two tables

English	German	
table	tisch	m
chair	stuhl	m
bed	bett	n
child	kind	n
brother	bruder	m
sister	schwester	f
house	haus	n
sun	sonne	f
cloud	wolke	f
wind	wind	m
rain	regen	m

English	Italian	
table	tavolo	m
chair	sedia	f
bed	letto	m
child	bambino	m
child	bambina	f
brother	fratello	m
sister	sorella	f
house	casa	f
sun	sole	f
cloud	nube	f
wind	vento	m
rain	pioggia	f

The first table consists of English nouns, their German translations, and the grammatical gender in German. Like in Russian, nouns in German can be (f)eminine, (n)eutral, or (m)asculine.

The second table contains English nouns with Italian translations and the grammatical gender in Italian. Italian has only two genders (f)eminine and (m)asculine.

1. Create a predicate `enggerm` with three arguments, and enter the table above as facts.

   Try out that your predicate works, for example by typing goals

   ```
   enggerm( rain, G, A ).
   ```

```
enggerm( E, wolke, A ).
enggerm( E, G, f ).   % Enumerates all feminine nouns.
```

2. Do the same for the Italian table. Create a predicate `engit` with three arguments.

3. German has three definite articles, which depend on grammatical gender of the word. Here they are:

   ```
   article( f, die ).
   article( m, der ).
   article( n, das ).
   ```

   Create a Horn clause `gerwitharticle( E, A, G )` that expresses that `E` is an English word, `G` is its German translation, and `A` is the correct article. For example `gerwitharticle( sun, A, G )` should result in `A = die, G = sonne`.

4. Define, as a Horn clause, a predicate `gerit( G, I )` that connects German words to their Italian translations. Make sure that it works in both directions, i.e. when `G` is fixed, or `I` is fixed.

5. Define, as a Horn clause, a predicate `samegender(E)` that succeeds if `E` is an English word, whose German and Italian translations have the same grammatical gender. For example `samegender(E)` should enumerate table, brother, sister, sun, cloud, and wind.

Task 2 (Swedish Royal Family)

In the second part of this exercise, we are going to study the Swedisch royal family. Study the file **sweden.docx**.

1. Create predicates `male`, `female` and create facts about who is male or female. Names normally start with a capital, which Prolog doesn't like, so in your code all names must start with a lowercase letter. You may ignore all diacritics, so ä and å can be written as a. If a name consists of more than one part, use an underscore to connect the parts, so `Christopher O'Neill` becomes `christopher_o_neill`.

2. Add all instances of the `parent` relation as fact.

3. Add all instances of `rulesover`, where the king and queen rule over sweden, and everyone who is duke/duchess of something, rules over this something. For example `carl_philip` rules over `varmland` and `victoria` over `vastergotland`.

4. Create Horn clauses for the predicates `sibling(X,Y)` , `brother` and `sister`. Make sure that nobody is his/her own sibling. (Use `\=` for inequality testing.)

5. Define the `father` and `mother` predicate as parents who are male/female.

6. Define `son` and `daughter` predicates.

7. Define the `king` predicate as male persons who rule over Sweden. Similarly, define the `queen` predicate.

8. Define the `duke` predicate as a male person who rules over something that is not Sweden. Similarly, define duchess. (Use `\=` for nonequality.)