Artificial Intelligence: 3rd Assignment

Deadline: April 13th, 2018

March 19th, 2018

1. You have been asked to implement a program that can produce a trip plan. The program must be able to answer questions such as:

   - In which days of the week there is a direct flight from Place1 to Place2?
   - What are the available flights from Place1 to Place2 on Thursday?
   - I need to visit cities C1, C2 and C3, starting my flight in city S on Tuesday and returning to S on Friday. In which sequence I should visit cities C1, C2 and C3 so that I don't need to have more than one flight a day?

   Your program must be implemented in Prolog. The database of flights should use the syntax shown in Figure 1, where `timetable(Place1,PLace2,List_of_flights)` represents the information about each flight (you may add more information as you wish, always following that syntax), with `Place1`, the origin, `Place2`, the destination, and `List_of_flights`, a list containing info about each flight, structured as: `departureTime/ArrivalTime/FlightNumber/ListOfDays`.

   The first problem is to find routes between two cities in a given day of the week. Route can be implemented using the predicate: `route(Place1,Place2,Day,Route)`, that satisfies the following criteria:

   - the start point (origin) of Route is Place1;
   - the destination is Place2;
   - all flights must be completed in the same day (`Day`);
   - all flights in `Route` must be in the relation `timetable`;
   - there must be sufficient time for transfers.

   The route must be structured in the following way: `From-To:FlightNumber:DepartureTime`.

   You may also need the auxiliary predicates:

   - `flight(Place1,PLace2,Day,Flight_num,Dep_time,Arr_time)`: says that there exists a flight between `Place1` and `Place2` in day `Day` with departure time and arrival time.
   - `deptime(Route,Time)`: departure time of `Route` is `Time`.
   - `transfer(Time1,Time2)`, there is a distance of at least 40 minutes between `Time1` and `Time2` so that a transfer can be made to another flight.

   Write a Prolog program using this info that can answer questions like the three posed above. The third question can be written as a combination of calls to the predicate route/4.

   Write the same program using your favorite procedural language.

   Compare both programs discussing about: development time, code size, execution time.

```
timetable(edinburgh,london,
[ 9:40/10:50/ba4733/alldays,
 13:40/14:50/ba4773/alldays,
   19:40/20:50/ba4833/[mo,tu,we,th,fr,su]]).

timetable(london,edinburgh,
[ 9:40/10:50/ba4732/alldays,
 11:40/12:50/ba4752/alldays,
   18:40/19:50/ba4822/[mo,tu,we,th,fr]]).

timetable(london,ljubljana,
[13:20/16:20/ju201/[fr],
   13:20/16:20/ju213/[su]]).

timetable(london,zurich,
[ 9:10/11:45/ba614/alldays,
 14:45/17:20/sr805/alldays]).

timetable(london,milan,
[ 8:30/11:20/ba510/alldays,
 11:00/13:50/az459/alldays]).

timetable(ljubljana,zurich,
[11:30/12:40/ju322/[tu,th]]).

timetable(ljubljana,london,
[11:10/12:20/yu200/[fr],
 11:25/12:20/yu212/[su]]).

timetable(milan,london,
[ 9:10/10:00/az458/alldays,
 12:20/13:10/ba511/alldays]).

timetable(milan,zurich,
[ 9:25/10:15/sr621/alldays,
 12:45/13:35/sr623/alldays]).

timetable(zurich,ljubljana,
[13:30/14:40/yu323/[tu,th]]).

timetable(zurich,london,
[ 9:00/9:40/ba613/[mo,tu,we,th,fr,sa],
   16:10/16:55/sr806/[mo,tu,we,th,fr,su]]).

timetable(zurich,milan,
[ 7:55/8:45/sr620/alldays]).
```

Figure 1: Example of a Flight database in Prolog

2. Write a Prolog program, using DCG (Definite Clause Grammar), that can analyze sentences in Portuguese. Part of a Portuguese grammar is given in Figure 2, where the first predicate `frase_nom` describes structures of singular sentences and the second set of predicates `frase_nom_p` describe plural sentences. This syntax is accepted by Prolog! (consult "Definite Clause Grammars" - in any Prolog book). A sentence is normally composed of a nominal sentence followed by a verbal sentence.

   Define the remaining of the Portuguese grammar using this syntax (possible verbal phrases, complements etc) and a dictionary of words (verbs, names, adjectives, etc). Use the set of phrases of Figure 2 to test your program. **Don't forget that any name started with a capital letter in Prolog is interpreted as a variable, therefore if your sentence starts with a capital letter, you should surround it by quotes!**

```
sentenca --> noun_phrase, verbal_phrase.
sentenca --> noun_phrase_p, verbal_phrase_p.

frase_nom --> determiner_f, noun_f.
frase_nom --> determiner_m, noun_m.
frase_nom --> noun_f.
frase_nom --> noun_m.

frase_nom_p --> determiner_p_f, noun_p_f.
frase_nom_p --> determiner_p_m, noun_p_m.
frase_nom_p --> noun_p_f.
frase_nom_p --> noun_p_m.
```

Figure 2: Subset of structure of the Portuguese language

Modify this program so that it also returns the syntactic tree corresponding to the sentence that was analyzed. In other words, besides answering if the sentence is well constructed according to the defined syntax, it needs also to produce the syntactic grammar tree. For example, the query `sentence(X,['A',vida,corre],[])` should return in variable X the value:

`sent(noun_prase(determiner('A'),noun(vida)),verbal_phrase(verb(corre))).`

Observe that the sentence "O sino bateu na menina" is syntactically correct, but it can be semantically incorrect depending on the context. There are techniques to code associated semantics and contexts to natural language processing, but this is **out of the scope of this assignment.**

Write the same program using any other language processing tool or procedural language of your choice.

EXAMPLES OF CORRECT PHRASES

```
sentence(['A',menina,corre,para,a,floresta],[]).
sentence(['A',menina,corre,para,a,mae],[]).
sentence(['A',vida,corre],[]).
sentence(['O',tempo,corre],[]).
sentence(['O',cacador,correu,com,os,lobos],[]).
sentence(['A',noticia,correu,pela,cidade],[]).
sentence(['As',lagrimas,corriam,pelo,rosto],[]).
sentence(['O',rio,corre,para,o,mar],[]).
sentence(['A',menina,bateu,a,porta],[]).
sentence(['A',porta,bateu],[]).
sentence(['O',vento,bateu,a,porta],[]).
sentence(['A',menina,bateu,na,porta],[]).
sentence(['O',martelo,bateu,na,porta],[]).
sentence(['A',menina,bateu,no,cachorro],[]).
sentence(['A',menina,bateu,no,tambor],[]).
sentence(['Os',tambores,bateram],[]).
sentence(['O',sino,bateu],[]).
sentence(['A',menina,corre],[]).
sentence(['A',vida,correu],[]).
sentence(['A',noticia,correu,para,a,floresta],[]).
sentence(['A',vida,correu,com,os,lobos],[]).
sentence(['A',menina,bateu,a,mae],[]).
```

EXAMPLES OF INCORRECT PHRASES

```
sentence(['A',tempo,corre],[]).
sentence(['O',tempo,correram],[]).
sentence(['A',cacador,corriam,pela,rosto],[]).
sentence(['A',tambores,correu,pela,floresta],[]).
sentence(['Os',tambores,bateu,na,porta],[]).
sentence(['O',sino,bateu,na,meninas],[]).
```

Figure 3: Set of test phrases (above: correct sentences; below: incorrect sentences)

What to deliver:

1. written report about your implementations.

   Organization:

   Introduction (diferences between procedural and declarative languages, advantages and disadvantage of both representations)

   Description/Representation of your problem in Prolog

   Description/Representation of your problem in the other language.

   Discussion about both representations

   Final Comments and Conclusions

2. source code, how to compile, execute, what is the input format (for Prolog and the other language). Besides, also mention the environment and versions of system and languages used to develop your programs.

Submission, as usual, will be through Moodle.