Principles of Knowledge Representation and Reasoning

B. Nebel, S. Wölfl, J. Hué
Winter Semester 2013/2014

University of Freiburg
Department of Computer Science

Exercise Sheet 7

Due: December 11, 2013

Exercise 7.1 (Find answer sets, 2)

Find an answer set for the program Π_n consisting of the following n rules

$$p_i \leftarrow not\ p_{i+1}. \qquad (1 \leq i < n)$$

where n is a natural number.

Exercise 7.2 (Well-founded semantics, 2)

Let Π be a logic program and $Atoms(\Pi)$ be the atoms occurring in Π. Show that the consequences of the reduct $\Pi^{Atoms(\Pi)}$ of Π by $Atoms(\Pi)$ are contained in any answer set of Π.

Exercise 7.3 (Travelling salesman problem, 6)

The *travelling salesman problem (TSP)* is specified as follows: Given a set of cities and the (Euclidean) distances between each pair of them, what is the shortest possible route that visits each city exactly once and returns to the first city in the route?

In this exercise, the task is to create a program that will translate an instance of the Travelling salesman problem into ASP, run an ASP solver to find a shortest path and write the result to standard output.

Source code **must be submitted** to: `hue@informatik.uni-freiburg.de`.

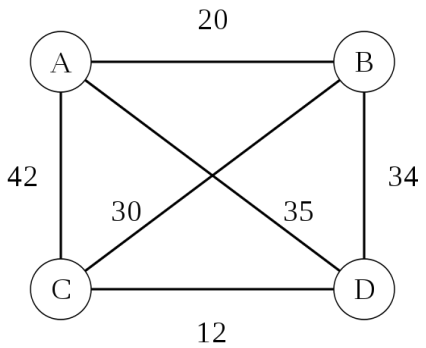

Figure 1: Example of the Travelling salesman problem

Details: The input format is as follows: the first line provides the exhaustive list of cities. Each of the following lines specifies the distance of a direct route between two cities (one that does not pass through any of the other cities) as a triple: `(source destination distance)`. The last line is a 0. Note that

two cities need not be connected via any direct route. The graph is assumed to be symmetric, i.e., the distance between a and b is identical to the distance between b and a. Thus, the graph given by Figure 1 can be encoded in the following way:

```
a b c d
a b 20
a c 42
a d 35
b c 30
b d 34
c d 12
0
```

Remember that a solution is a subset of the given roads (in both directions) and that each city needs to be visited exactly once (except the origin city which is also the end city).

Hint: The command `#minimize[ travelled(X,Y,Z) = Z ].` will attach to each predicate `travelled(X,Y,Z)` the weight `Z` and keep as answer set only the ones with a minimal sum over the weights of true atoms.