CS_205 Declarative Programming 20 November 2020
Monika Seisenberger

Lab Week 8: Unification, Proof Search, Recursion

1. Which of the following pairs of terms can be unified? If so, give a unifying variable instantiation.

(a) `footmassage(vincent, X) = footmassage(Y, mother(mia)).`

(b) `footmassage(Y, mother(Y)) = footmassage(mother(vincent), mother(Z)).`

(c) `kills(vincent, X) = kills(Y, mia, Z).`

(d) `kills(X, father (mia)) = kills (mother(yolanda), X).`

(e) `relationships(father(Y),father(X),X)=relationships(Y,father(Y),father(X)).`

2. Assume we have the following knowledge base:

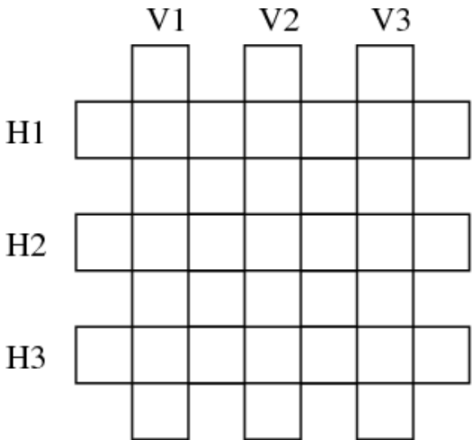
```
train(swansea, cardiff).
train(cardiff, manchester).
train(cardiff, bristol).
train(cardiff, london).
train(london, paris).
train(paris, munich).
train(munich, vienna).
```

Define a predicate connection with two cities as arguments, that yieds `true/yes` if there is a train connection between the two cities. For instance, `connection(swansea, vienna)` should lead to a positive outcome. Further, extend your database by one ore more appropriate rules, so that there is also a connection between `vienna` and `swansea`. Which problem do you now encounter for other queries? [Note, there is no need to make this solution perfect, but if you can the better of course.]

3. Here is a first puzzle. Take the following six Italian words:

```
astante , astoria , baratto , cobalto , pistola , statale .
```

They are to be arranged, crossword puzzle fashion, in the following grid, where there are three words listed vertically, and three words listed horizontally. Here the second, forth, and sixth letter of the first vertical word, coincides with the second letter of the first, second, and third word horizontally.



The following knowledge base represents a lexicon containing these words:

```
word(astante,  a,s,t,a,n,t,e).
word(astoria,  a,s,t,o,r,i,a).
word(baratto,  b,a,r,a,t,t,o).
word(cobalto,  c,o,b,a,l,t,o).
word(pistola,  p,i,s,t,o,l,a).
word(statale,  s,t,a,t,a,l,e).
```

Write a predicate crossword/6 that tells us how to fill in the grid. The first three arguments should be the vertical words from left to right, and the last three arguments the horizontal words from top to bottom.

Hint: Produce a first solution of the following form:

```
crossword(V1,V2,V3,H1,H2,H3) :-

word(V1, _,_,_,_,_,_,_), word(V2,_,_,_,_,_,_,_), ....
```

It should produce all possible ways to arrange the 6 words in the grid (not fulfilling any constraints). (How many would this be? You do not have to compute these manually, but you should be able to demonstrate the first solutions given by this approach.)

Create a second solution by adding constraints (Which of the letters do you want to coincide?) excluding the solutions that do not make sense.