universität innsbruck

Constraint Solving	WS 2021	LVA 703305
Week 2		October 28, 2021

Homework

1. Find all the stable models of the following logic programs using the procedure presented in the lecture. In each case demonstrate the procedure for at least one candidate model which is not a stable model.
   Hint: Know how to do the procedure on paper, but use Clingo to get all the complete models.

   (a)
   ```
   a. b.
   d:- a,not i.
   c:- not f,not d, h.
   i:- f,h,not d.
   f:- b,d.
   e:- not f.
   h:- e, not i.
   ```

   (b)
   ```
   q:- not p.
   p:- not w,k.
   w:- not r.
   r:- not q.
   k:- not v.
   v:- not k.
   ```

2. Using the procedure presented in the lecture, ground the following logic programs. Demonstrate each step.

   (a)
   ```
   r(1..3).
   w(2..5).
   t(X,Y):- not r(X), q(X,Y).
   q(X,Y):- w(X), w(Y), X!=Y.
   ```

   (b)
   ```
   q(0,I,0) :- I=0..4.
   r(I,J)   :- I=0..3,J=1..4,J=I+1.
   t(0,I,J) :- I=0..2 , J=0..2 , I=J.
   t(X,Y,Z) :- r(W,X),r(K,Z),t(W,Y,K).
   q(X,Y,Z) :- t(W,Y,Z),r(K,X),q(K,Y,W).
   ```

What do operations t and q represent?

3. Write an answer set program which decides if a graph has a path between two nodes. This program should be ran as follows:

   ```
   clingo path_input.lp path_gen_test.lp -c n=a -c m=b 0
   ```

   where a and b are the nodes we are checking the existance of a path between. Your program must work on the following input.

   ```
   node(1..6).
   edge(4,1). edge(1,2). edge(4,2).
   edge(2,3). edge(4,3). edge(5,4).
   ```

   Note that edges should be considered bi-directional. It is not necessary for the input to represent this property.

4. Write an answer set program which decides if a string s is a palindrome. We will assume that $1 \leq |s|$. This program should be ran as follows:

   ```
   clingo Pal_input.lp Pal_gen_test.lp -c n=a 0
   ```

   Where a can either be the length of the string or the largest index (i.e. $|s| - 1$ for 0-indexed strings). Make it clear which case is used.

   Hint: Section 3 of github.com/potassco/guide/releases/tag/v2.2.0 may help. Some of the code provided in the lecture may help. A solution that is not recursive is also possible.