Knowledge Representation and Reasoning Systems

Test 2 (2015/16)

2 hours / closed book

Group 1 (Answer set computation)

1) Present all the stable models of the following program:

```
a :- not b.
b :- not a.
c :- not b, not d.
d :- not c.

{e} :- a.

:- a, not e.
```

2) Translate the following program into a normal logic program:

```
2 {a;b;c} :- e, f.        f :- 2 {d;e} .        e.        :- not f.
```

3) Check if {b,c} is an answer set of the next disjunctive logic program:

```
a ; b :- not a.
a ; c :- not b.
c     :- not a.
```

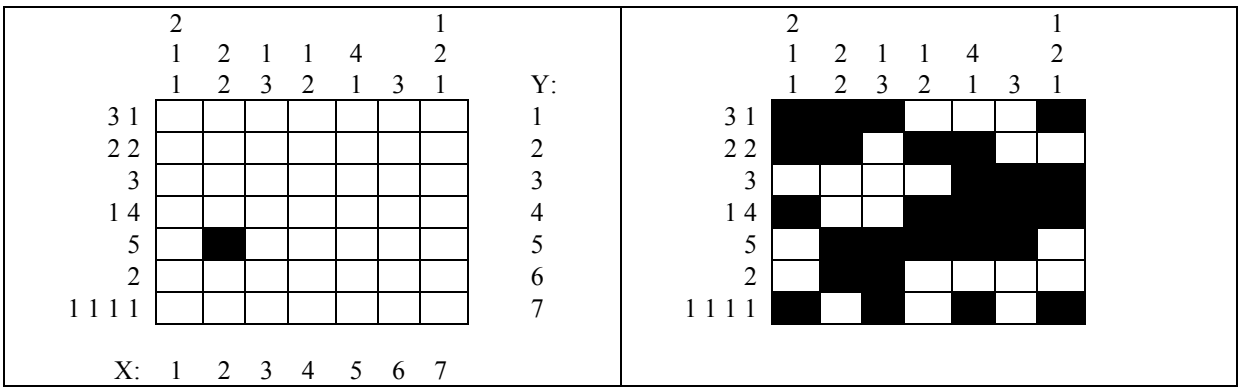
4) Use the approximation $\mathbf{solve}_P$ algorithm to determine one stable model of the program:

```
a :- not b.
b :- not a.
c :- a.
c :- b.
d :- not c.
```

Group 2 (ASP modelling)

The GCHQ Christmas card puzzle is a grid-shading puzzle, where each square is either black or white. Some of the black squares have already been filled in for you. Each row or column is labelled with a string of numbers. The numbers indicate the length of all consecutive runs of black squares, and are displayed in the order that the runs appear in that line. For example, a label "2 1 6" indicates sets of two, one and six black squares, each of which will have at least one white square separating them.

You can find an instance of the puzzle below, and the corresponding solution (the coordinates of each square are represented on the bottom and on the right hand side).



Using answer set programming, produce a program for the CLINGO system that will allow you to solve this kind of puzzles. Consider the following instance, capturing some of the information set forth:

```
h(1..7).      runh(1,1,3).          runv(1,1,2).
v(1..7).      runh(1,2,1).          runv(1,2,1).
              runh(2,1,2).          runv(1,3,1).
              runh(2,2,2).          …
              …                     black(2,5).
```

The instances of predicate **runh(Y,R,SIZE)** describes that in line **Y**, run **R** has **SIZE** black squares (and similarly for vertical runs represented by **runv(X,R,SIZE)**). The predicate **black(X,Y)** indicates that the square at position (X,Y) is black. The dimensions of the board are given by predicates **h/1** and **v/1**.

It is suggested that you use in your implementation the following auxiliary predicates, besides **black/2**:

1. **starth(X,Y,R)** / **startv(X,Y,R)** to indicate that horizontal/vertical run R starts at square **(X,Y)**.
2. **painted(X,Y)** to indicate that square (X,Y) has been painted by some run.

In your implementation you may present only the rules taking care of the horizontal runs (the vertical ones are symmetric). Don't forget that:

1. All runs must start at some square;
2. You cannot have a run in a line (or column) with lower index starting at the same or after the position of a subsequent run;
3. All runs must have non-black squares separating them (there are very simple ways of enforcing this restriction!)
4. All black squares must have been painted by some run.

THE END