50.021 – Artificial Intelligence

Kwan Hui

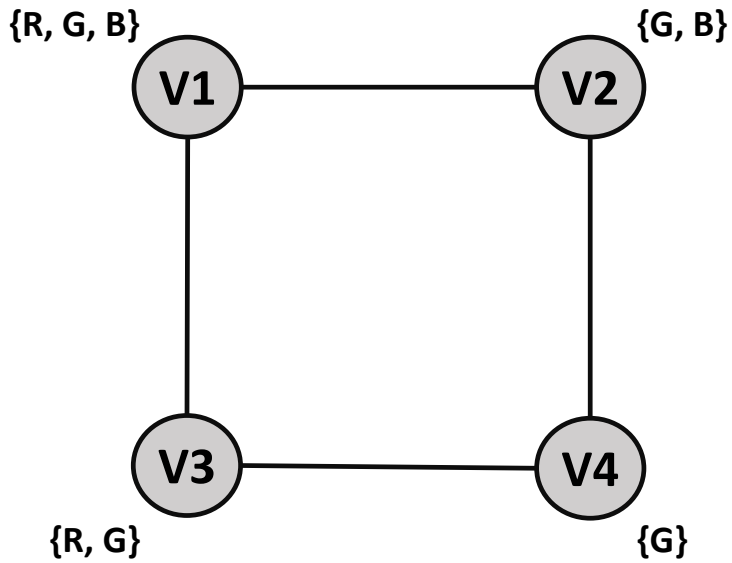
Week 3 - Constraint Satisfaction Problems

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources.]

These answers are provided only as a brief guide. There could be more than one way to answer the questions.

1 Pure backtracking

Let's look at a simple map coloring problem and explore the behavior of constraint propagation, backtracking and forward checking in this context. Here is the relevant problem specification:



Assume we use pure backtracking to search for a solution to this problem. We use a fixed variable ordering in the search (V1, V2, V3 then V4) and the values are considered in the order shown in the picture above. Then, show the order in which individual variable assignments are considered by backtracking (this is analogous to the order in which nodes are expanded by depth-first search).

Show all assignments even if it is immediately found to be inconsistent upon testing. Write each variable assignment per line as a number from 1 to 4 (indicating the

variable) followed by a letter, drawn from R, G, B. The first four assignments are given below, complete the rest.

```
V1 R
V2 G
V3 R
V3 G
?
```

Answer: Pure backtracking
V1 R
V2 G
V3 R (V3 R is inconsistent with V1 R, so backtrack)
V3 G
V4 G
V2 B
V3 R
V3 G
V4 G
V1 G
V2 G
V2 B
V3 R
V4 G

Note: There are more inconsistent values than the one stated above. You should try to understand and work it out.

2 Backtracking with forward checking

Repeat the previous question but using backtracking with forward checking (BT-FC) to search for a solution to this problem. We use the same variable ordering and value ordering as before. Show the order in which assignments are considered by BT-FC. Whenever propagating after an assignment causes a domain to become empty, that causes the search to backtrack. Write all assignments as before.

Answer: Backtracking with forward checking
V1 R (R removed from V3 as part of forward checking)
V2 G
V2 B
V3 G
V1 G
V2 B
V3 R
V4 G

Note: There are more values removed than the one stated above. You should try to understand and work it out.

3 Arc consistency

Apply the AC-3 algorithm on the same constraint graph. Write the values in each of the indicated variable domains after any changes required to achieve arc consistency

for just that arc. Then, assume that the following arcs are done sequentially, with the effects on the domains propagating. Write domains as a sequence of letters, for example, R B. If there are no values left in a domain, write None.

```
V1 - V2: D1=RGB, D2=GB
V4 - V2: D4=   , D2=
V1 - V3: D1=   , D3=
?
```

```
V1 - V2: D1=RGB, D2=GB
V4 - V2: D4=   , D2=
V1 - V3: D1=   , D3=
?
```

Answer: AC-3 Algorithm
V1 - V2: D1=RGB, D2=GB
V4 - V2: D4=G, D2=GB
V1 - V3: D1=RGB, D3=RG
V4 - V3: D4=G, D3=RG
V2 - V4: D2=B, D4=G (D2=G deleted, add all neighbours)
V3 - V4: D3=R, D4=G (D3=G deleted, add all neighbours)
V1 - V2: D1=RG, D2=B (D1=B deleted, add all neighbours)
V4 - V2: D4=G, D2=B
V1 - V3: D1=G, D3=R (D1=R deleted, add all neighbours)
V4 - V3: D4=G, D3=R
... (Potentially still more edges in queue)
Solution: D1=G, D2=B, D3=R, D4=G

Note: As mentioned during the lecture and cohort sessions, the ordering of arcs do not matter in general. The important thing is that you consider all edges/arcs initially and add the appropriate edges/arcs when you remove values.