MIT EECS
6.034: Artificial Intelligence
(Fall 2006)
Instructor: Luis E. Ortiz

Solutions to Practice Problems for Constraint Satisfaction Problems

Original date: October 28, 2005; Last revised: October 23, 2006

1. (a) **None of the variable domains change:**

$$1 = \{R, B\} \quad 2 = \{R, B\} \quad 3 = \{R, B\}$$
$$4 = \{R, B\} \quad 5 = \{R, B\}$$

(b)

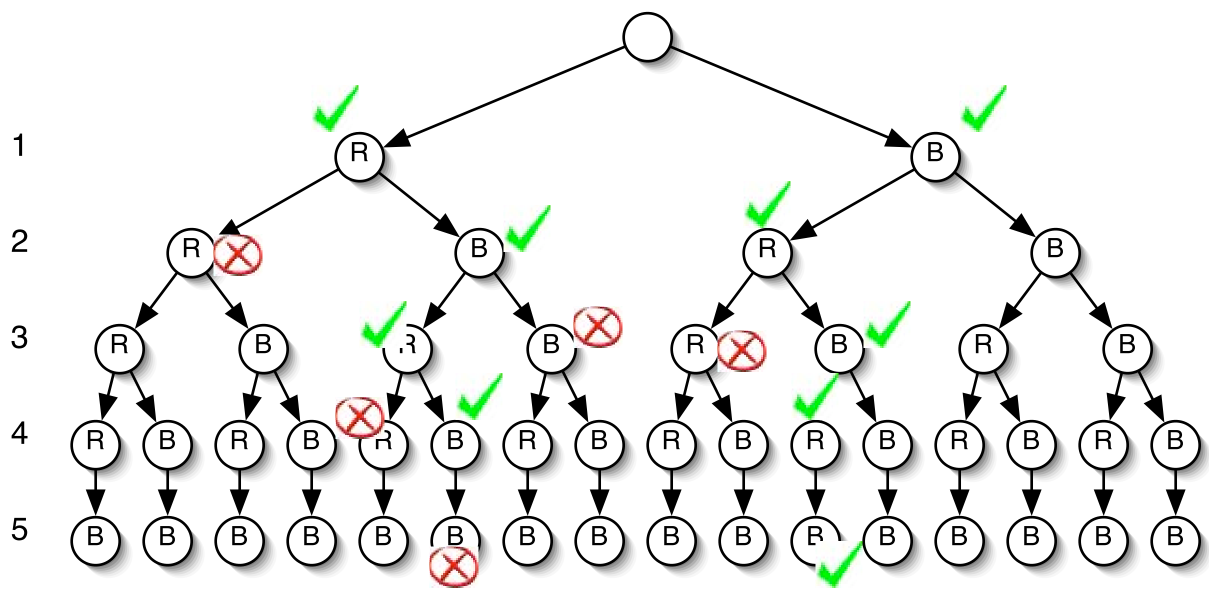
$$1 = \{B\} \quad 2 = \{R\} \quad 3 = \{B\}$$
$$4 = \{R\} \quad 5 = \{B\}$$

(c) **Forward checking is defined as a single iteration of constraint propagation only on those edges that terminate at the variable whose value was just set, and that do not originate from variables which have already been set. Therefore, after we set $1 = R$, forward checking affects the domains of variables 2 and 4 since they are adjacent to variable 1 (and have not yet been assigned).**

$$1 = \{R\} \quad 2 = \{B\} \quad 3 = \{R, B\}$$
$$4 = \{B\} \quad 5 = \{R, B\}$$

Forward checking only does one step of propagation, only to the immediate neighbors of the assigned variable.

(d)



(e) **We must keep track of the variable domains as we search since forward checking modifies these domains based on the current assignment, and we will need to restore the domain of earlier search nodes if we have to backtrack to them. We fail at a node if (1) the current assignments violate some**

constraint, or (2) if forward checking after the present assignment causes the domain of some variable to become empty. The following lists (in order from left to right) each attempted assignments and the resulting variable domains *after* forward checking.

Assignment:	None	$1 = R$	$2 = B$	$1 = B$	$2 = R$	$3 = B$	$4 = R$	$5 = B$
Domain of 1:	$\{R, B\}$	**R**	**R**	**B**	**B**	**B**	**B**	**B**
Domain of 2:	$\{R, B\}$	$\{B\}$	**B**	$\{R\}$	**R**	**R**	**R**	**R**
Domain of 3:	$\{R, B\}$	$\{R, B\}$	$\{R\}$	$\{R, B\}$	$\{B\}$	**B**	**B**	**B**
Domain of 4:	$\{R, B\}$	$\{B\}$	$\{B\}$	$\{R\}$	$\{R\}$	$\{R\}$	**R**	**R**
Domain of 5:	$\{B\}$	$\{B\}$	$\{\}$	$\{B\}$	$\{B\}$	$\{B\}$	$\{B\}$	**B**
			$\Downarrow$ **FAIL**					

Note that when we fail at $2 = B$, since there are no further values to try in the domain of variable 2, we backtrack to the assignment of variable 1. When this happens, we restore the domains from *before* variable 1 was assigned, i.e. the ones listed above under "None".

(f) *LEO: Need modification!*

Use of the most-constrained-variable strategy entails assigning the variable first whose domain is smallest. This ordering is not only performed at the start of the search. Rather, it is updated after each variable is assigned and forward checking modifies the domains of unassigned variables. For this problem, variable 5 has the smallest domain initially. After assigning variable 5, the domains of variable 2 and 4 become smaller than those of variables 3 and 5. Since variable 2 has the lowest index, it is assigned next. And so on, as shown below:

Assignment:	None	$5 = B$	$2 = R$	$1 = B$	$3 = B$	$4 = R$
Domain of 1:	$\{R, B\}$	$\{R, B\}$	$\{B\}$	**B**	**B**	**B**
Domain of 2:	$\{R, B\}$	$\{R\}$	**R**	**R**	**R**	**R**
Domain of 3:	$\{R, B\}$	$\{R, B\}$	$\{B\}$	$\{B\}$	**B**	**B**
Domain of 4:	$\{R, B\}$	$\{R\}$	$\{R\}$	$\{R\}$	$\{R\}$	**R**
Domain of 5:	$\{B\}$	**B**	**B**	**B**	**B**	**B**

2. **Formulation A:** The variables are the $3k$ instrument/time slots.
   1. Domain: **for each instrument/time slot, the set of observations requesting that instrument and time slot and the value "empty"**
   2. Size of domain: **at most m*n+1 per variable**
   3. Satisfied constraints: **C2, since each variable (instrument/time) gets at most one value, an observation.**
   4. Binary constraints?:

   - **C1 is not a binary constraint in this formulation. It requires checking all the variable assignments at once to make sure that exactly two observations from each scientist's list are made.**
   - **C3 is a binary constraint in this formulation. Place a constraint between the 3 variables with the same time slot and require that the targets of the assigned observation be equal if they are both non-empty.**

   Formulation B: The variables are the m scientists.
   1. Domain: **for each scientist, the set of all pairs of observations that scientist requested.**
   2. Size of domain: $\binom{n}{2}$, **approximately** $n^2/2$.
   3. Satisfied constraints: **C1, since we will guarantee that exactly two of the scientist's observations are scheduled.**
   4. Binary constraints?:

   - **C2 is a binary constraint in this formulation. Place a constraint between every pair of variables and require that the instrument/time slot requests don't conflict.**
   - **C3 is a binary constraint in this formulation. Place a constraint between every pair of variables and require that the targets for observations with the same time slot don't conflict.**

   Formulation C: The variables are the mn scientists' requests.
   1. Domain: {Granted, Rejected}
   2. Size of domain: 2
   3. Satisfied constraints: None
   4. Binary constraints?:

   - **C1 is not a binary constraint in this formulation. It requires checking all the variable assignments of Granted observations at once to make sure that exactly two observations from each scientist's list are granted.**
   - **C2 is a binary constraint in this formulation. Place a constraint between every pair of variables and require that the instrument/time slot requests don't conflict between any two Granted requests.**
   - **C3 is a binary constraint in this formulation. Place a constraint between every pair of variables and require that the targets of the Granted observations with the same time slot don't conflict.**