CS 188
Fall 2022

Regular Discussion 3 Solutions

1 CSP: Air Traffic Control

We have five planes: A, B, C, D, and E and two runways: international and domestic. We would like to schedule a time slot and runway for each aircraft to **either** land or take off. We have four time slots: $\{1, 2, 3, 4\}$ for each runway, during which we can schedule a landing or take off of a plane. Constraints:

- Plane B has lost an engine and must land in time slot 1.
- Plane D can only arrive at the airport to land during or after time slot 3.
- Plane A is running low on fuel but can last until at most time slot 2.
- Plane D must land before plane C takes off, because some passengers must transfer from D to C.
- Planes A, B, and C cater to international flights and can only use the international runway.
- Planes D and E cater to domestic flights and can only use the domestic runway.
- No two aircrafts can reserve the same time slot for the same runway.

(a) Complete the formulation of this problem as a CSP in terms of variables, domains, and constraints (both unary and binary). Constraints should be expressed implicitly using mathematical or logical notation rather than with words.

Variables: A, B, C, D, E for each plane.

Domains: a tuple *(time slot, runway type)* for time slot $\in \{1, 2, 3, 4\}$ and runway type $\in \{international, domestic\}$.

Constraints:

$$B[0] = 1$$

$$D[0] \geq 3$$

$$A[0] \leq 2$$

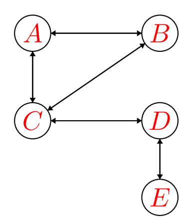
$$D[0] < C[0]$$

$$A[1] = B[1] = C[1] = international$$

$$D[1] = E[1] = domestic$$

$$A \neq B, B \neq C, A \neq C, D \neq E$$

Constraint Graph:



(b) Since unary constraints determine which runway each plane must use, for the next two parts we will look at just the timeslot.
What are the domains of the variables after enforcing unary constraints and arc-consistency?

A	~~1~~	2	~~3~~	~~4~~
B	1	~~2~~	~~3~~	~~4~~
C	~~1~~	~~2~~	~~3~~	4
D	~~1~~	~~2~~	3	~~4~~
E	1	2	~~3~~	4

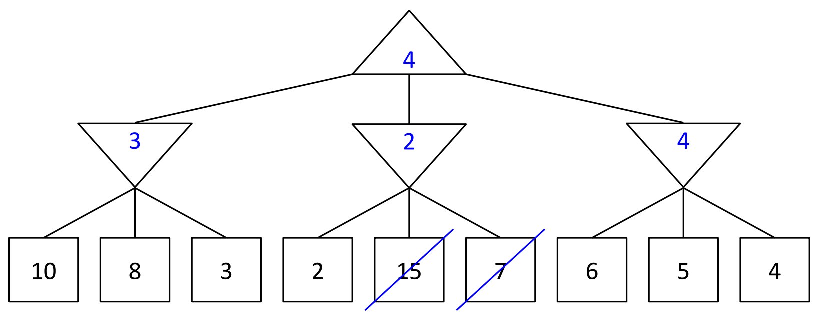
(c) Arc-consistency can be expensive to enforce, and we may get faster solutions using only **forward-checking**. Using MRV, perform backtracking search, breaking ties by picking lower values first. List the variable assignments in the order they occur (including assignments that are reverted).

A	1	2	3	4
B	1	2	3	4
C	1	2	3	4
D	1	2	3	4
E	1	2	3	4

Ans: (B, 1), (A, 2), (C, 3), (C, 4), (D, 3), (E, 1)

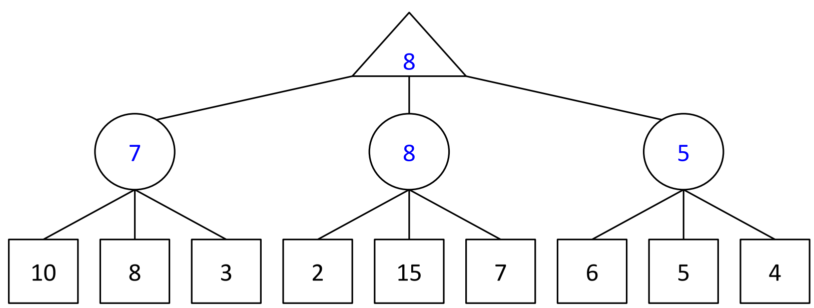
2 Game Trees

(a) Consider the zero-sum game tree shown below. Triangles that point up, such as at the top node (root), represent choices for the maximizing player; triangles that point down represent choices for the minimizing player. Assuming both players act optimally, fill in the minimax value of each node.



(b) The search goes from left to right. Can any nodes be pruned? Explain. Notice that the fifth and sixth leaf nodes are pruned. When we explore the middle branch, we know that our maximizer value will be at least 3. Upon seeing a 2 as the first leaf node for the middle branch, we can conclude that the value of the middle branch would be 2 or less, so we don't need to look at the other leaves.

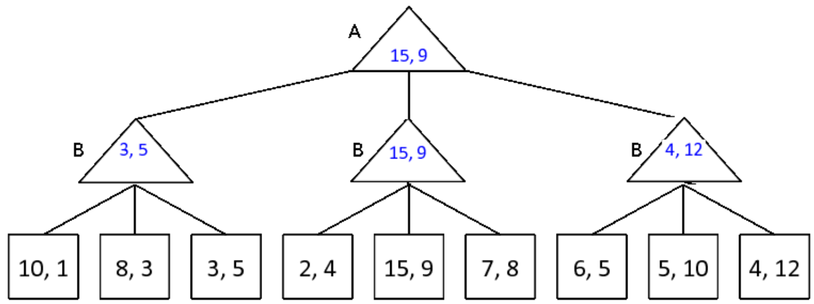
(c) Consider the same zero-sum game tree, except that now, instead of a minimizing player, we have a chance node that will select a value uniformly at random. Fill in the expectimax value of each node.



(d) Can any nodes be pruned? Explain. No nodes can be pruned. There will always be the possibility that an as-yet-unvisited leaf of the current parent chance node will have a very high value, which increases the overall average value for that chance node. For example, when we see that leaf 4 has a value of 2, which is much less than the value of the left chance node, 7, at this point we cannot make any assumptions about how the value of the middle chance node will ultimately be more or less in value than the left chance node. As it turns out, the leaf 5 has a value of 15, which brings the expected value of the middle chance node to 8, which is greater than the value of the left chance node.

A side note: In the case where there is an upper bound to the value of a leaf node, there is a possibility of pruning: suppose that an upper bound of +10 applies only to the children of the rightmost chance node. In this case, after seeing that leaf 7 has a value of 6 and leaf 8 has a value of 5, the best possible value that the rightmost chance node can take on is $\frac{6+5+10}{3} = 7$, which is less than 8, the value of the middle chance node. Therefore, it is possible to prune leaf 9 in this case.

(e) Let's look at a non-zero-sum version of a game. In this formulation, player A's utility will be represented as the first of the two leaf numbers, and player B's utility will be represented as the second of the two leaf numbers. Fill in this non-zero game tree assuming each player is acting optimally.



(f) Can any nodes be pruned? Explain. No nodes can be pruned. Because this game is non-zero-sum, there can exist a leaf node anywhere in the tree that is good for both player A and player B.