SAARLAND UNIVERSITY ARTIFICIAL INTELLIGENCE
Prof. Jana Koehler, Dr. Álvaro Torralba

Exercise Sheet 6.
Solution

Exercise 22. (3 Points)

Consider the following constraint network $\gamma = (V, D, C)$. Lisa is moving out of Saarbrücken soon, and has many tasks to finish in only 5 days. These tasks are:

(a) arrange a farewell party for her friends,

(b) book a bus ticket,

(c) clean her apartment,

(d) de-register from the city,

(e) end all her contracts,

(f) find a subtenant for her apartment,

(g) get packing cases.

The time slots in which she can schedule these tasks are Monday (M), Tuesday (Tu), Wednesday (W), Thursday (Th) and Friday (F). To define the constraints we will use the + operation on days of the weeks. To do that, we interpret every day as a number from Monday (1) to Friday (5). For example, $F = Th + 1$, and $Th = M + 3$ but $M \neq F + 1$. Formally, the network is defined as follows:

- Variables: $V = \{\text{a, b, c, d, e, f, g}\}$.
- Domains: For all $v \in V$: $D_v = \{\text{M, Tu, W, Th, F}\}$.

- Constraints :
  - Since Lisa will need to go to the city to get party supplies and packing cases, she wants to do both on the same day. (a=g)
  - She cannot end her contracts without de-registering first, so she needs to end her contracts two days after de-registering. (e=d+2)
  - She needs to find a subtenant before the apartment gets filled with the party stuff, so she wants to do it two days before the party. (f=a-2)
  - She can only book her bus ticket after finding a subtenant. (b=f+1)
  - She cannot clean the apartment before the party, since the apartment will get dirty afterwards. So she wants to clean the apartment a day after the party. (c=a+1)
  - She will only de-register from the city once she knows when her bus is. (d=b+1)

Run the AC-3 algorithm, as specified in the lecture, on the given constraint network. For each iteration of the while-loop, give the content of M at the start of the iteration, give the pair (u, v) removed from M, give the domain D_u of u after the call to Revise(γ, u, v), and give the pairs (w, u) added into M.

Note: Initialize M as a lexicographically ordered list (i.e., (a, b) would be before (a, c), both before (b, a) etc., if any of those exist). After initialization, use M as a FIFO queue, i.e., always remove the element at the front and add new elements to the back.

(Solution)

1. $M = \{(a, c), (a, f), (a, g), (b, d), (b, f), (c, a), (d, b), (d, e), (e, d), (f, a), (f, b), (g, a)\}$;
   pair selected: (a, c);
   $a = c - 1$, hence $D_a = \{\text{M}, \text{Tu}, \text{W}, \text{Th}\}$;
   Do we have any other (*, a)? - Yes, but (f, a) and (g, a) are already in M, so nothing to do.

2. $M = \{(a, f), (a, g), (b, d), (b, f), (c, a), (d, b), (d, e), (e, d), (f, a), (f, b), (g, a)\}$;
   pair selected: (a, f);
   $a = f + 2$, hence $D_a = \{\text{W}, \text{Th}\}$;
   Do we have any other (*, a)? - Yes, but (c, a) and (g, a) are already in M, so nothing to do.

3. $M = \{(a, g), (b, d), (b, f), (c, a), (d, b), (d, e), (e, d), (f, a), (f, b), (g, a)\}$;
   pair selected: (a, g);
   $a = g$, hence no modification on the domain of a;
   no insertion into M.

4. $M = \{(b,d),(b,f),(c,a),(d,b),(d,e),(e,d),(f,a),(f,b),(g,a)\}$;
pair selected: (b,d);
$b = d-1$, hence $D_b = \{\text{M}, \text{Tu}, \text{W}, \text{Th}\}$;
Do we have any other (*, b)? - Yes, but (f,b) is already in M, so nothing to do.

5. $M = \{(b,f),(c,a),(d,b),(d,e),(e,d),(f,a),(f,b),(g,a)\}$;
pair selected: (b,f);
$b = f+1$, hence $D_b = \{Tu, W, Th\}$;
Do we have any $(*,b)$? - Yes, but (d,b) is already in M, so nothing to do.

6. $M = \{(c,a),(d,b),(d,e),(e,d),(f,a),(f,b),(g,a)\}$;
pair selected: (c,a);
$c = a+1$, hence $D_c = \{Th, F\}$;
Do we have any $(*,c)$? - No, so no modification on M.

7. $M = \{(d,b),(d,e),(e,d),(f,a),(f,b),(g,a)\}$;
pair selected: (d,b);
$d = b+1$, hence $D_d = \{W, Th, Fr\}$;
Do we have any $(*,d)$? - Yes, but (e,d) is already in M, so nothing to do.

8. $M = \{(d,e),(e,d),(f,a),(f,b),(g,a)\}$;
pair selected: (d,e);
$d = e-2$, so $D_d = \{W\}$;
Do we have any $(*,d)$? - Yes, so we add (b,d) to M.

9. $M = \{(e,d),(f,a),(f,b),(g,a),(b,d)\}$;
pair selected: (e,d);
$e = d+2$ hence $D_e = \{F\}$;
Do we have any $(*,e)$? - No, so no modification on M.

10. $M = \{(f,a),(f,b),(g,a),(b,d)\}$;
pair selected: (f,a);
$f = a-2$ hence $D_f = \{M, Tu\}$;
Do we have any $(*,f)$? - Yes, so we add (b,f) to M.

11. $M = \{(f,b),(g,a),(b,d),(b,f)\}$;
pair selected: (f,b);
$f = b-1$, hence no modification on the domain of f;
no insertion into M.

12. $M = \{(g,a),(b,d),(b,f)\}$;
pair selected: (g,a);
$g = a$, hence $D_g = \{W, Th\}$;
Do we have any $(*,g)$? - No, so no modification on M.

13. $M = \{(b,d),(b,f)\}$;
pair selected: (b,d);
$b = d - 1$ hence $D_b = \{Tu\}$;
Do we have any $(*,b)$? - Yes, so we add (f,b) to M.

14. $M = \{(b,f),(f,b)\}$;
pair selected: (b,f);
$b = f + 1$, hence no modification on the domain of f;
no insertion into M.

15. $M = \{(f,b)\}$;
pair selected: (f,b);
$f = b - 1$, hence $D_f = \{M\}$;
Do we have any $(*,f)$? Yes, so we add (a,f) to M.

16. $M = \{(a,f)\}$;
pair selected: (a,f);
$a = f + 2$, hence $D_a = \{W\}$;
Do we have any $(*,a)$? Yes, so we add (c,a) and (g,a) to M.

17. $M = \{(c,a),(g,a)\}$;
pair selected: (c,a);
$c = a + 1$, hence $D_c = \{Th\}$;
Do we have any $(*,c)$? No, so no insertion into M.

18. $M = \{(g,a)\}$;
pair selected: (g,a);
$g = a$, hence $D_g = \{W\}$;
Do we have any $(*,g)$? No, so no insertion into M.

19. M empty; return modified γ :
$D_a = \{W\}$
$D_b = \{Tu\}$
$D_c = \{Th\}$
$D_d = \{W\}$
$D_e = \{F\}$
$D_f = \{M\}$
$D_g = \{W\}$

(/Solution)

Exercise 23. (2.5 Points)

Run the AcyclicCG algorithm from the lecture slides on the problem defined in Exercise 22. More precisely, execute its 4 steps as follows:

(a) Draw the constraint graph of γ. Pick "f" as the root and draw the directed tree obtained by step 1 (see Chapter 10 slides 36 and 37 for examples). Give the resulting variable order obtained by step 2. If the ordering of variables is not unique, break ties by using the alphabetical order.

(b) List the calls to Revise$(\gamma, v_{parent(i)}, v_i)$ in the order executed by step 3, and for each of them give the resulting domain of $v_{parent(i)}$.

(c) For each recursive call to BacktrackingWithInference during step 4, give the domain D'_{v_i} of the selected variable v_i after Forward Checking, and give the value $d \in D'_{v_i}$ assigned to v_i.

Note: Step 4 runs BacktrackingWithInference with variable order $v_1, \ldots, v_n$. This means that, at the ith recursion level, "select some variable v for which a is not defined" will select v_i.

(Solution)

a) Variable ordering: f, a, b, c, d, e, g. The constraint graph and directed tree are given in Figures 1, and 2.

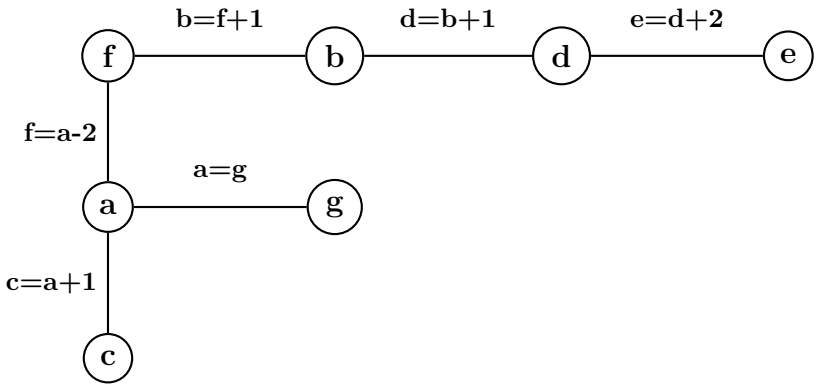


Figure 1: The constraint graph of Exercise 23 (a).

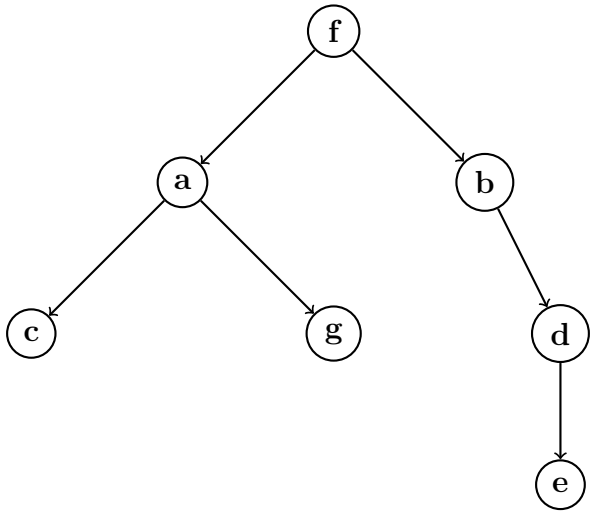


Figure 2: The directed tree of Exercise 23 (a).

b) The calls to $Revise(\gamma, v_{parent(i)}, v_i)$ and the resulting domains are:

- i=7: $Revise(\gamma, a, g)$: $D_a = D_g = \{M, Tu, W, Th, F\}$.
- i=6: $Revise(\gamma, d, e)$: $D_d = \{M, Tu, W\}$.
- i=5: $Revise(\gamma, b, d)$: $D_b = \{M, Tu\}$.
- i=4: $Revise(\gamma, a, c)$: $D_a = \{M, Tu, W, Th\}$.
- i=3: $Revise(\gamma, f, b)$: $D_f = \{M\}$.
- i=2: $Revise(\gamma, f, a)$: $D_f = \{M\}$.

c) Possible D'_{v_i} and $d \in D'_{v_i}$ are:

- $D'_f = \{M\}$; $d = M$.
- $D'_a = \{W\}$; $d = W$.
- $D'_b = \{Tu\}$; $d = Tu$.
- $D'_c = \{Th\}$; $d = Th$.
- $D'_d = \{W\}$; $d = W$.
- $D'_e = \{F\}$; $d = F$.
- $D'_g = \{W\}$; $d = W$.

(/Solution)

Exercise 24. (2.5 Points)

Consider the following constraint network $\gamma = (V, D, C)$:

- Variables: $V = \{a, b, c, d, e, f, g, h, i, j\}$.
- Domains: For all $v \in V$: $D_v = \{20, 30, 40, 50, 60\}$.
- Constraints: $a < b$; $a < c$; $a < h$; $b < c$; $a < e$; $e < i$; $e < h$; $e < g$; $e < f$; $e < d$; $c < d$; $d < f$; $c < f$; $h < i$; $i < j$;

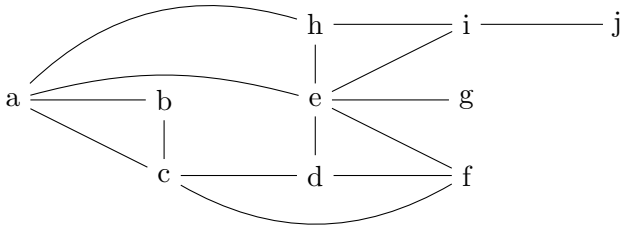
(a) Draw the constraint graph of γ.

(b) What is the optimal cutset V_0 for γ?

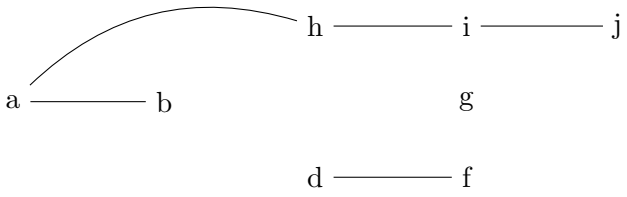
(c) If the CutsetConditioning algorithm from the lecture is called with such a minimal cutset V_0, how many calls to AcyclicCG will be performed in the worst case? Justify your answer.

(Solution)

(a) The constraint graph of γ is:



(b) $V_0 = \{c, e\}$



(c) There is only one assignment to c and e (40, 30) which results in a valid global assignment, so in the worst case, we have $D_c \times D_e = 5 \times 5 = 25$ calls to AcyclicCG as every possible combination has to be checked.

(/Solution)

Exercise 25. (2 Points)

Max and Minnie are playing a variant of the game Tic-tac-toe in which players take turns marking the cells in a 3×3 grid. Max marks Xs and Minnie marks Os. The player chooses a column and the disc is placed at **the first empty position** starting from the **bottom**.

In their variant of the game, **Minnie wins with utility -1 if *any* line (horizontal, vertical, or diagonal) fills up with three Os, whereas Max wins with utility $+1$ if *any* line (horizontal, vertical, or diagonal) fills up with three Xs**.

o		
x		
x	o	o

Figure 3: A 3×3 Tic-tac-toe grid. Max plays next.

(a) Consider the initial state depicted Figure 3. Here, it is Max's turn to play.

Draw the full Minimax tree and annotate every node with its utility.

(b) Consider the evaluation function $f(x) :=$ the number of rows, columns, and diagonals which contain at most one O. For example, in the initial state $f(x) = 6$.

Note: The evaluation function is equivalent to 8 minus the number of rows, columns, and diagonals that *do* contain at least two spaces marked as O.

Draw the **Minimax tree with this evaluation function and a depth of 2. Annotate every node with its utility.** Is this a good evaluation function? Justify your answer.

(Solution)

1. See Figure 4.
2. See Figure 5. This is a good evaluation function because it helps Max avoid the move which would make Minnie win. For the first move taken by Max, Max can now win in 1 move after Minnie makes her move.

(/Solution)

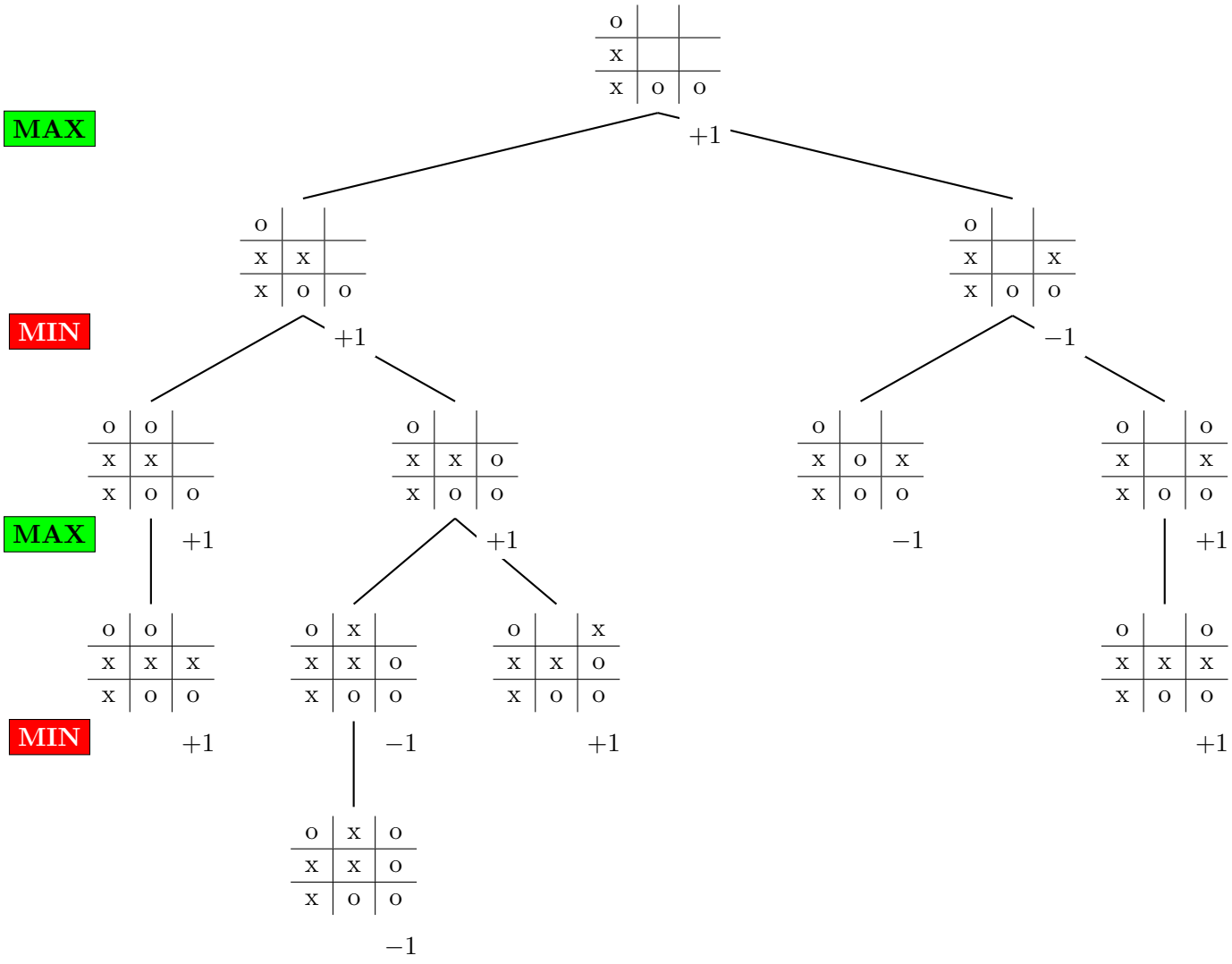


Figure 4: Solution of Exercise 25 (a).

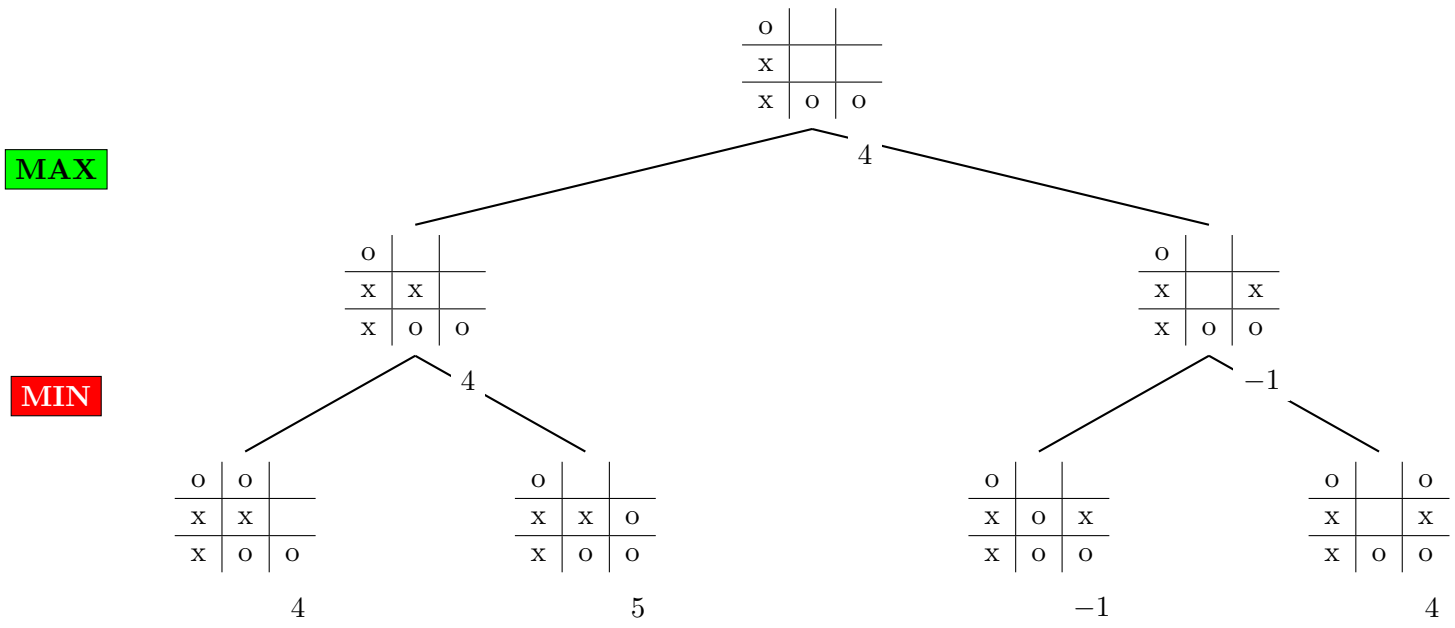


Figure 5: Solution of Exercise 25 (b).