SAPIENZA
UNIVERSITÀ DI ROMA

Artificial Intelligence

2023/2024 Prof: Sara Bernardini

Lab 5:
AC-3 and AcyclicCG

Francesco Argenziano
email: argenziano@diag.uniroma1.it

*The slides have been prepared using the textbook material available on the web, and the slides of the previous editions of the course by Prof. Luigia Carlucci Aiello, Prof. Daniele Nardi and Dott. Fabio Previtali.

Example: Scheduling Tasks

Consider the following constraint network $\gamma = (V, D, C)$. Lisa is moving out of Egham 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 council,

(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 $v \in V$: $D_v = \{\mathrm{M}, \mathrm{Tu}, \mathrm{W}, \mathrm{Th}, \mathrm{F}\}$.

Example: Scheduling Tasks

- 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. 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.

Example: Scheduling Tasks

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.

Example: Scheduling Tasks

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.

Example: Scheduling Tasks

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.

Example: Scheduling Tasks

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.

Example: Scheduling Tasks

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 :
    $D_a = \{W\}$
    $D_b = \{Tu\}$
    $D_c = \{Th\}$
    $D_d = \{W\}$
    $D_e = \{F\}$
    $D_f = \{M\}$
    $D_g = \{W\}$

Example: Constraint Network

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

- Variables: $V = \{a, b, c, d, e, f, g\}$
- Domains: for all $u \in V$, $D_u = \{1, 2, 3, 4, 5, 6\}$
- Constraints: $a = 2d,\ g = 3d,\ d = b,\ e = a - 3,\ c = b + 3,\ f = e + 2$

Run the AcyclicCG algorithm from the lecture slides Draw the constraint graph of γ. Pick d as the root and draw the directed tree obtained by performing step 1. Give the resulting variable ordering obtained by performing step 2. If the ordering of the variables is not unique, break ties using alphabetical order. List the calls to $Revise(\gamma, v_{parent(i)}, v_i)$ in the order executed by running step 3, and, for each of them, give the resulting domain of $v_{parent(i)}$. For each recursive call to BacktrackingWith-Inference in step 4, give the domain D'_{v_i} of the selected variable v_i after Forward Checking and the value $d \in D'_{v_i}$ assigned to v_i.

Example: Constraint Network

(Solution) The variable ordering is: d, a, b, c, e, f, g. The constraint graph and the directed tree are given in the Figures 1 and 2 below:

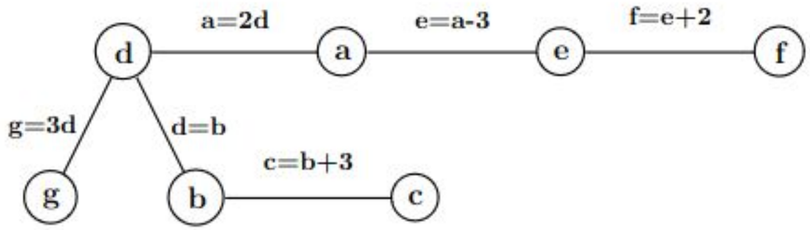
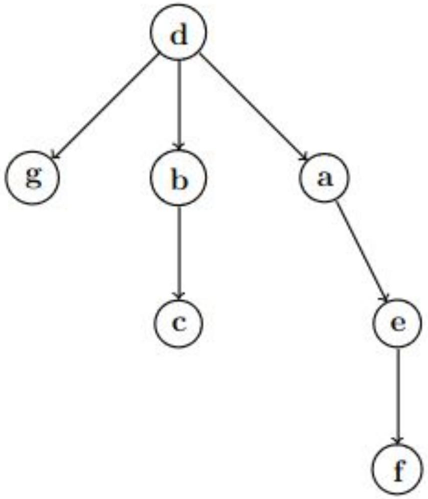


Figure 1: The constraint graph



Example: Constraint Network

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

- $i = 7 : Revise(\gamma, d, g), g = 3d : D_d = \{1, 2\}$
- $i = 6 : Revise(\gamma, e, f), f = e + 2 : D_e = \{1, 2, 3, 4\}$
- $i = 5 : Revise(\gamma, a, e), e = a - 3 : D_a = \{4, 5, 6\}$
- $i = 4 : Revise(\gamma, b, c), c = b + 3 : D_b = \{1, 2, 3\}$
- $i = 3 : Revise(\gamma, d, b), d = b : D_d = \{1, 2\}$
- $i = 2 : Revise(\gamma, d, a), a = 2d : D_d = \{2\}$

BackTrackingWithInference; possible D'_{v_i} and $d \in D'_{v_i}$ are:

- $D'_d = \{2\}; d = 2$
- $D'_a = \{4\}; d = 4$
- $D'_b = \{2\}; d = 2$
- $D'_c = \{5\}; d = 5$
- $D'_e = \{1\}; d = 1$
- $D'_f = \{3\}; d = 3$
- $D'_g = \{6\}; d = 6$