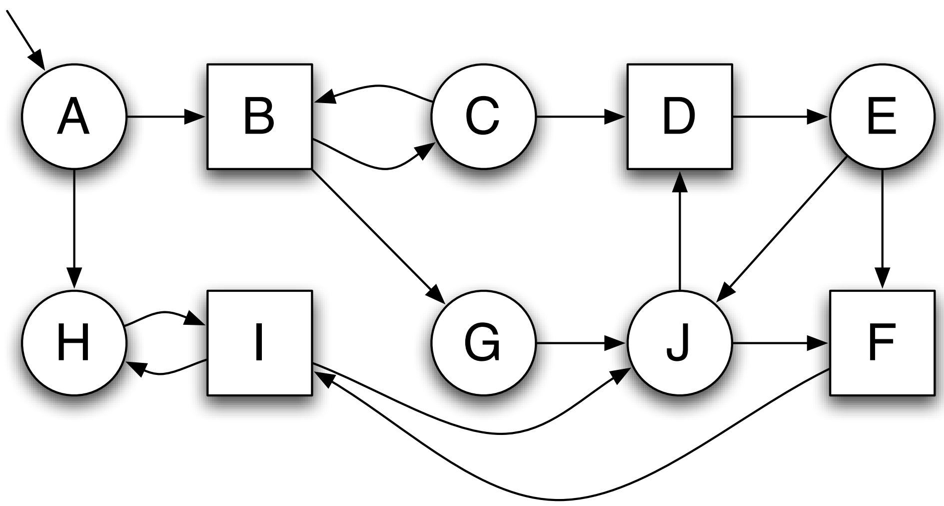
INFO-F-410 Embedded Systems Design
Game Theory

Gilles Geeraerts

Academic year 2010–2011

Exercise 1

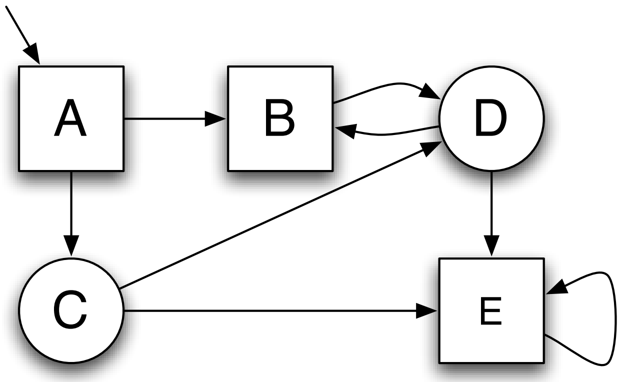
Let us consider the following arena where player B plays with round nodes:



- Formalise, using Muller conditions (weak or strong), the following objectives for B:
  1. The play always reaches I.
  2. The play always reaches I and J.
  3. The play never leaves either nodes A, B, D, E, G and J, or nodes A, B, C, D, E, and I.
  4. The play visits at least infinitely often J.
  5. The play visits infinitely often exactly A, I and H.
- For all these conditions try to devise a winning strategy for B, if you believe it exists.
- Apply the Attractor based algorithm on condition 1.

Exercise 2

Let us consider the following arena where player B plays with round nodes:

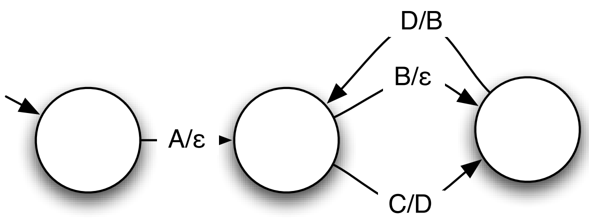


The objective for B is as follows: if the play visits C, it can't visit B but must visit D. If the play visits B, it can't visit C nor E

- Formalise that objective as a weak Muller condition.
- To formalise strategies that need memory, one can use a Mealy machine, which is a finite automaton that produces a word on its output while recognising a word on its input. Edges in a Mealy machines are thus labelled by pairs (i, o), where i is the read letter and o the letter produced on the output. To encode a strategy, the Meal machine should read on input the sequence of visited locations (i.e. the play), and produce on the output:

  1. Either ε when the position does not belong to B (the strategy we are building is for B only).
  2. Or the next state to be played by B, according to the strategy, if the position belongs to B.

  Is the strategy represented hereunder winning for the objective above ? Why ?



- Apply the algorithm studied during lectures to compute a winning strategy for B.
- Formalise that strategy as a Mealy machine.

Exercise 3

- In the algorithm to solve weak parity games, we have initialised the sequence of sets A_i as follows:

$$\begin{aligned} A_k &= \mathrm{Attr}_A(C_k) \\ A_{k-1} &= \mathrm{Attr}_B(C_{k-1} \setminus A_k) \end{aligned}$$

  where k is the maximal color and C_i is the set of nodes colored by i. Can we replace the definition of A_{k-1} by:

$$A_{k-1} = \mathrm{Attr}_B(C_{k-1}) \setminus A_k$$

  If yes, explain why. If no, give a counter-example.

- Is it correct to say that, for any set A and B: $\mathrm{Attr}_X(A \cup B) = \mathrm{Attr}_X(A) \cup \mathrm{Atrr}_X(B)$ (for some player X) ? Use your answer to explain why the definition of A_{k-2} given in the course is:

$$A_{k-2} = \mathrm{Attr}_A(C_{k-2} \setminus A_{k-1} \cup A_k)$$

  and not:

$$A_{k-2} = \mathrm{Attr}_A(C_{k-2} \setminus A_{k-1}) \cup A_k$$

Answers

Exercise 1

Objectives:

1. Weak objective: $\{S \mid I \in S\}$
2. Weak objective: $\{S \mid \{I, J\} \subseteq S\}$
3. Weak objective: $\{S \mid S \subseteq \{A, B, D, E, G, J\}$ or $S \subseteq \{A, B, C, D, EI\}\}$
4. Strong objective: $\{S \mid J \in S\}$
5. Strong objective: $\{\{A, I, H\}\}$

Winning strategies for B:

1. Any strategy s.t. $A \to H$, $H \to I$
2. Any strategy s.t. $A \to B$, $C \to D$, $E \to J$, $J \to F$, $G \to J$.
3. No winning strategy. From A, we must go to B to avoid H. Hence, player A can bring the play to C. From C, no choice allows player B to win: if player B ever chooses $C \to B$, player A can choose G as next state, and player B looses. If player B ever chooses $C \to D$, player A can force the game to visit E and J or F.
4. Any strategy s.t. $J \to D$, $A \to B$, $C \to D$, $E \to J$ and $G \to J$.
5. No winning strategy. Since A has no input edge, it is not possible to visit A infinitely often.

Attractor for B of $\{I\}$:

i	$\text{Attr}_B^i(\{J\})$
0	$\{I\}$
1	$\{F, H, I\}$
2	$\{A, E, F, H, I, J\}$
3	$\{A, D, E, F, G, H, I, J\}$
4	$\{A, C, D, E, F, G, H, I, J\}$
5	$\{A, B, C, D, E, F, G, H, I, J\}$

Player B can win from any initial position. To find a winning strategy, always go to node in a smaller attractor, i.e.:

- $A \to H$ ($\text{Attr}_B^2 \to \text{Attr}_B^1$)
- $H \to I$ ($\text{Attr}_B^1 \to \text{Attr}_B^0$)
- $C \to D$ ($\text{Attr}_B^4 \to \text{Attr}_B^3$)
- $E \to F$ ($\text{Attr}_B^2 \to \text{Attr}_B^1$)
- $G \to J$ ($\text{Attr}_B^3 \to \text{Attr}_B^2$)
- $J \to F$ ($\text{Attr}_B^2 \to \text{Attr}_B^1$)

Exercise 2

Idea of the winning strategy:

- If the play visits C, goto B, then, always choose to go to D from B.
- If the play has never visited C and we are in D, go to E

Thus, choosing the right successor for D requires memory.

Objective as a weak Muller:

$$\left\{ \begin{array}{c} \{C,D\},\{C,D,A\},\{C,D,E\},\{C,D,E,A\}, \\ \emptyset,\{A\},\{B\},\{D\},\{E\},\{A,B\},\{A,D\},\{A,E\},\{B,D\},\{D,E\},\{A,B,D\},\{A,D,E\} \end{array} \right\}$$

Idea of the construction: first consider all the subsets containing C. These must contain D and can't contain B. Thus we are left with four possibilities (for A and E).

Then, consider the remaining $2^4 = 16$ possibilities, and rule out the sets that contain B and E (since the objective says « if we visit B, we can't visit E »).

The Mealy machine is not a winning strategy, since it always plays the same move from D, i.e., go to B. This is losing if we have visited C before.

Reduction to a parity game, see Figure 1.

Computation of the winning states:

- $A_{11} = \mathrm{Attr}_A(\{17\}) = \{17, 18\}$
- $A_{10} = \mathrm{Attr}_B(\emptyset) = \emptyset$
- $A_9 = \mathrm{Attr}_A(\{11, 15, 16, 18\} \setminus \emptyset \cup A_{11}) = \{7, 11, 14, 15, 16, 17, 18\}$
- $A_8 = \mathrm{Attr}_B(\{13\} \setminus A_9 \cup \emptyset) = \{5, 6, 10, 13\}$
- $A_7 = \mathrm{Attr}_A(\{12\} \setminus A_8 \cup A_9) = \{7, 9, 11, 12, 14, \ldots, 18\}$
- $A_6 = \mathrm{Attr}_B(\{4, 7, 8, 10, 14\} \setminus A_7 \cup A_8) = \mathrm{Attr}_B(\{4, 8, 10\} \cup A_8) = \{2, \ldots 6, 8, 10, 13\}$
- $A_5 = A_7$
- $A_4 = \mathrm{Attr}_B(\{3\} \setminus A_5 \cup A_6) = \{1, 2, \ldots 6, 8, 10, 13\}$. Remark: at this point all the nodes belong either to A_4 or to A_5.
- $A_3 = A_5$
- $A_2 = A_4$
- $A_1 = A_3$
- $A_0 = A_2$.

Thus, $W_B = \{1, 2, 3, 4, 5, 6, 8, 10, 13\}$, $W_A = \{7, 8, 9, 11, 12, 14, 15, 16, 17, 18\}$. Hence B has a winning strategy:

- When in D and having seen A and B before, goto B ($3 \to 4$ in the parity game).
- When in D and having seen A, B and D before, goto B ($8 \to 4$ in the parity game).

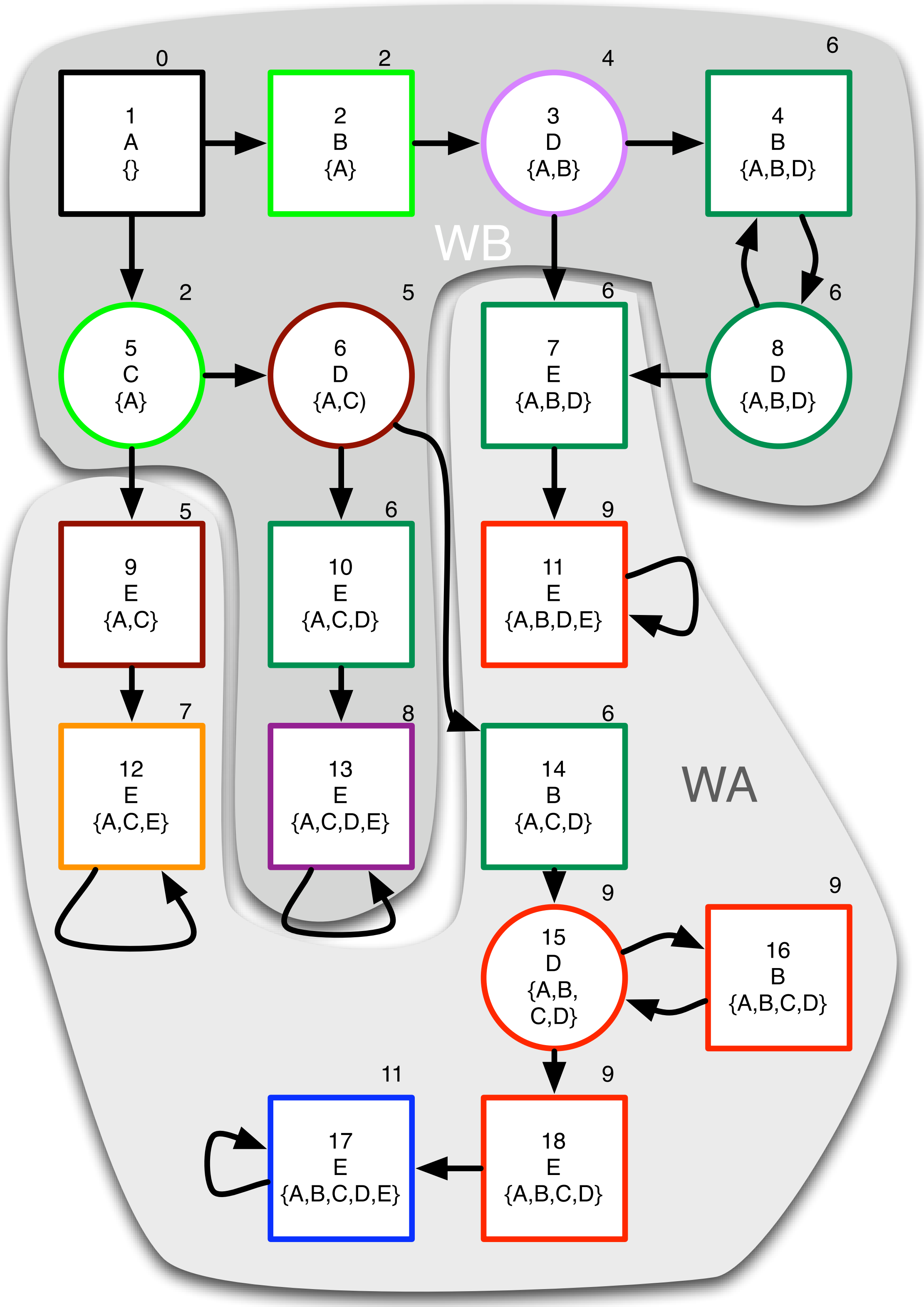
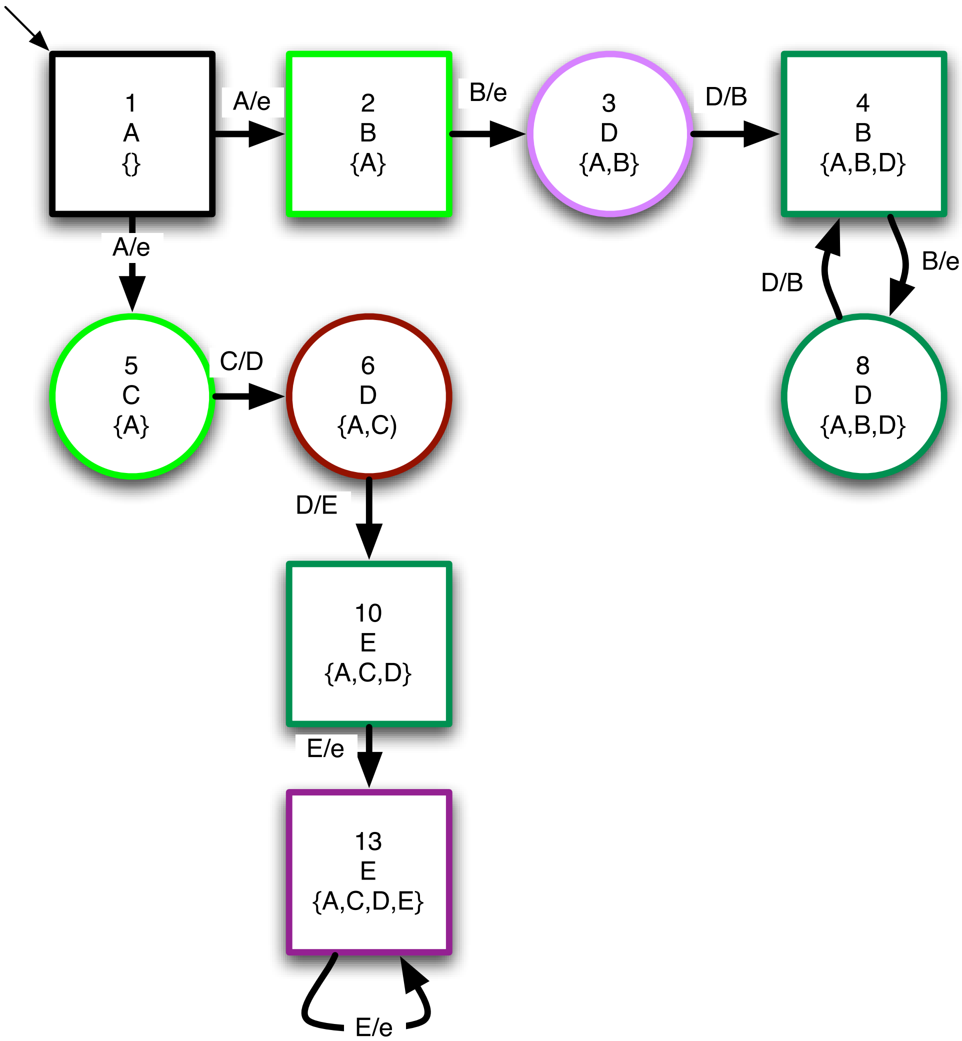


Figure 1: Reduction to a parity game, and winning regions

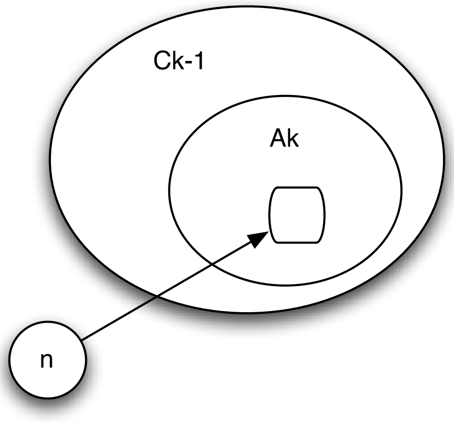
- When in C, an having seen A before, goto D ($5 \to 6$ in the parity game).
- When in D and having seen A and C before, goto E ($6 \to 10$ in the parity game).

This can be formalised as the following Mealy machine (which has to be made deterministic to be implemented. This is achieved by merging states 2 and 5):

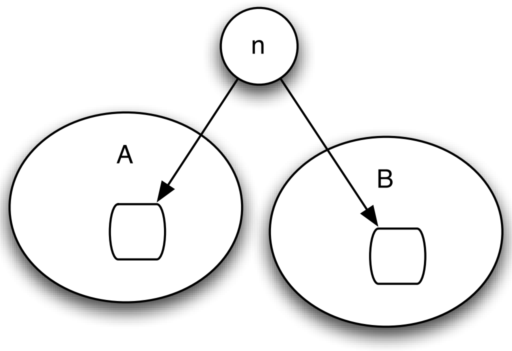


Exercise 3

1. No, we can't, as shown on this counter-example. Clearly, $n \in \mathrm{Attr}_B(C_{k-1})$, but $n \notin A_k$, hence, $n \in \mathrm{Attr}_B(C_{k-1}) \setminus A_k$. However, $n \notin \mathrm{Attr}_B(C_{k-1} \setminus A_k)$.



2. No, this is not correct. In the counter-example hereunder, if X plays with square nodes, we have $n \in \mathrm{Attr}_X(A \cup B)$, but neither $n \in \mathrm{Attr}_X(A)$ nor $n \in \mathrm{Attr}_X(B)$.



Remark that $A_k = \mathrm{Attr}_A(A_k)$. Thus, if the above property were correcty, we could have:

$$\begin{aligned} A_{k-2} &= \mathrm{Attr}_A(C_{k-2} \setminus A_{k-1} \cup A_k) \\ &= \mathrm{Attr}_A(C_{k-2} \setminus A_{k-1}) \cup \mathrm{Attr}_A(A_k) \\ &= \mathrm{Attr}_A(C_{k-2} \setminus A_{k-1}) \cup A_k \end{aligned}$$

However, the second equality does not hold, as the above counter example can be applied here too.