Exam 2020
(solution – parts I and III)

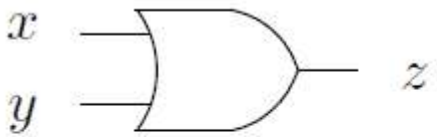
Radu Mateescu

Inria and LIG / Convecs

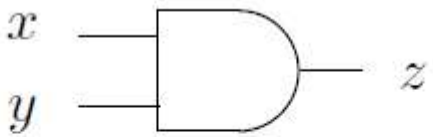
http://convecs.inria.fr

Part I: Modeling in LNT

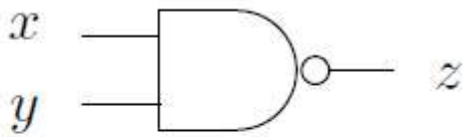
- Modeling of asynchronous circuits



OR
$(z = x \vee y)$



AND
$(z = x \wedge y)$



NAND
$(z = \neg(x \wedge y))$

- Each logical gate: a cyclic, event-driven LNT process
- Communication on wires:

channel LINK **is**

(Bool)

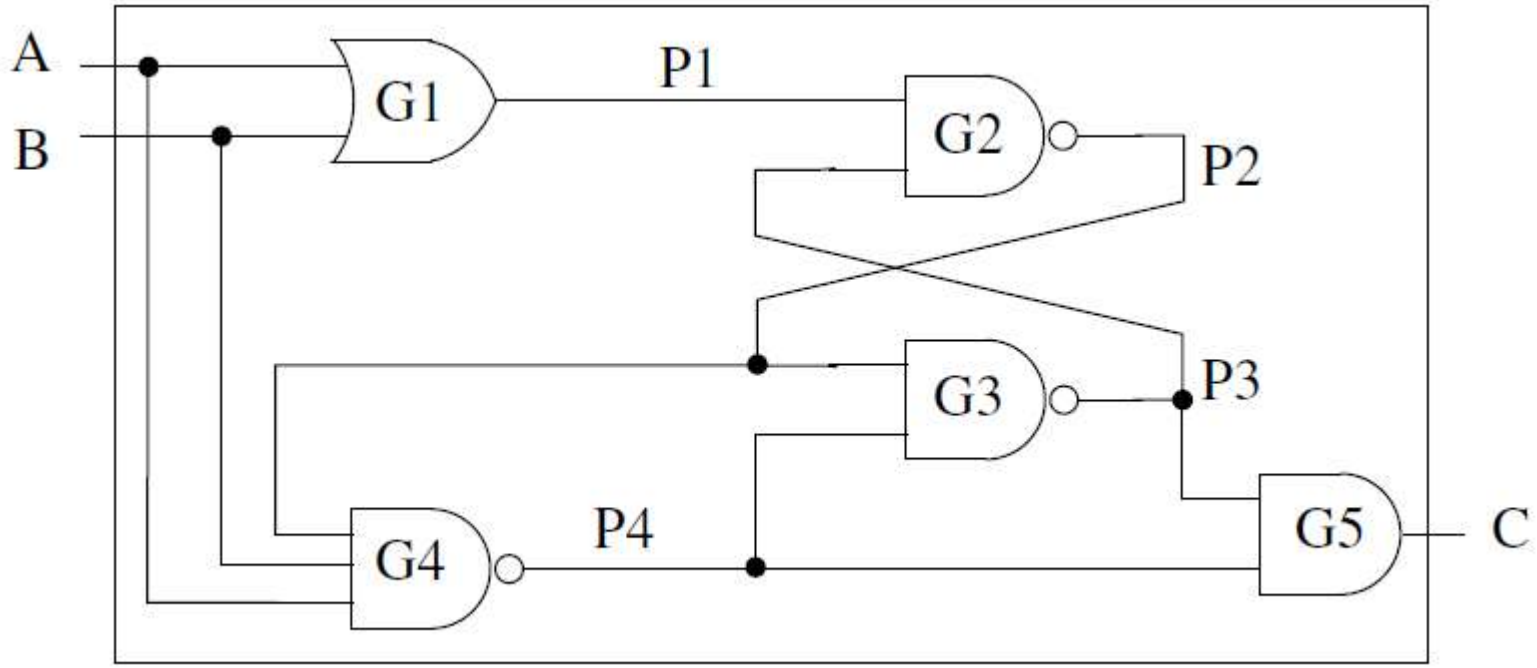
end channel

Question I.1: Basic logical elements

```
process NAND [INPUT1, INPUT2, OUTPUT:LINK] (in var X1, X2:Bool) is
   var RESULT, NEW_RESULT:Bool in
    RESULT := not (X1 and X2);
    loop
      select   INPUT1 (?X1)   []   INPUT2 (?X2)   end select;
      NEW_RESULT := not (X1 and X2);
      if NEW_RESULT != RESULT then
         RESULT := NEW_RESULT;
         OUTPUT (RESULT)
      end if
    end loop
   end var
end process
```

Question I.2: Muller's C element

- Mayevski's implementation of the C element



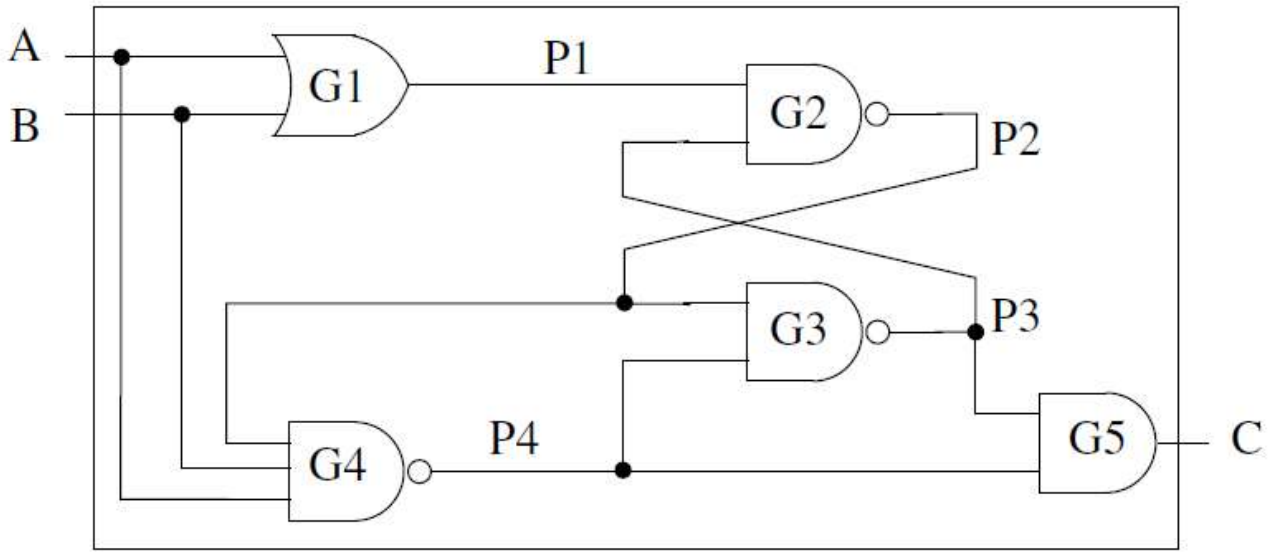
Question I.2: Muller's C element

```
process MULLER [A, B, C:LINK] (XA, XB, XP1, XP2, XP3, XP4:Bool) is
   hide P1, P2, P3, P4:LINK in
      par
           A, B, P1          ->  G1 [A, B, P1] (XA, XB)
      ||   P1, P2, P3        ->  G2 [P1, P3, P2] (XP1, XP3)
      ||   P2, P3, P4        ->  G3 [P2, P4, P3] (XP2, XP4)
      ||   A, B, P2, P4      ->  G4 [A, B, P2, P4] (XA, XB, XP2)
      ||   P3, P4            ->  G5 [P3, P4, C] (XP3, XP4)
      end par
   end hide
end process
```

Question I.2: Muller's C element

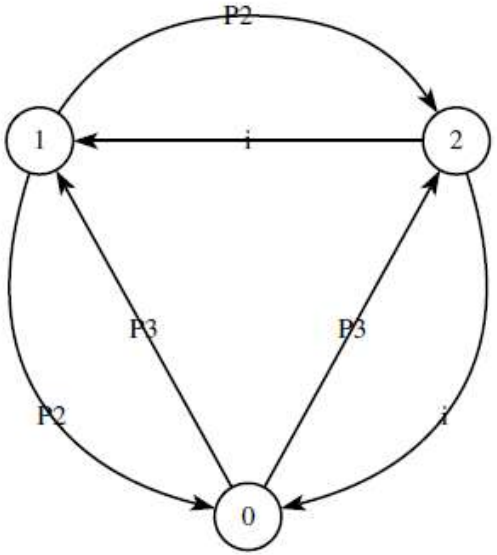
- *Subsidiary question*: knowing that inputs A, B are initially false (XA, XB must be set to false when invoking process MULLER), infer the values of parameters XP1, …, XP4.

  - XP1 = false
  - XP2 = true
  - XP3 = false
  - XP4 = true

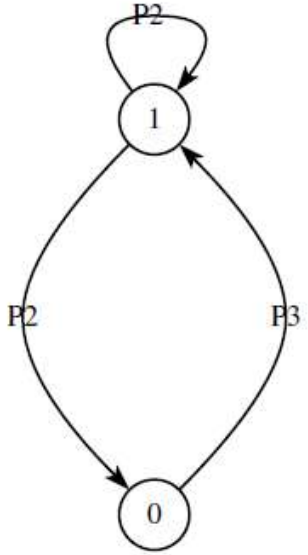


  - So the initial output C = false (same value as XA, XC, according to the C element behaviour)

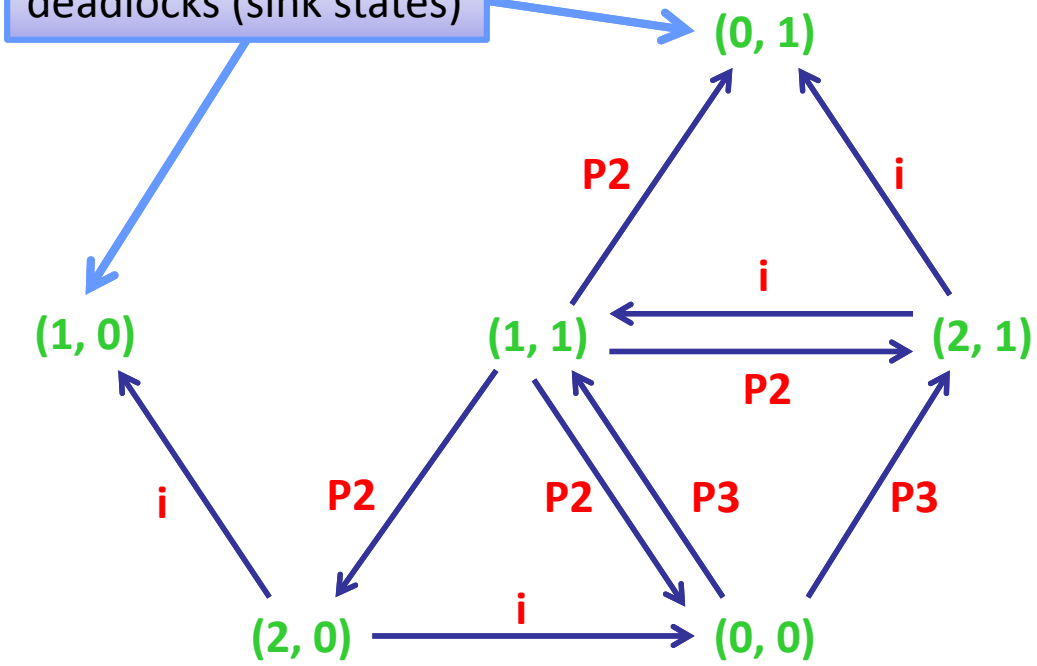
Question I.3: Synchronous product



$\otimes_{\{P2,P3\}}$



deadlocks (sink states)

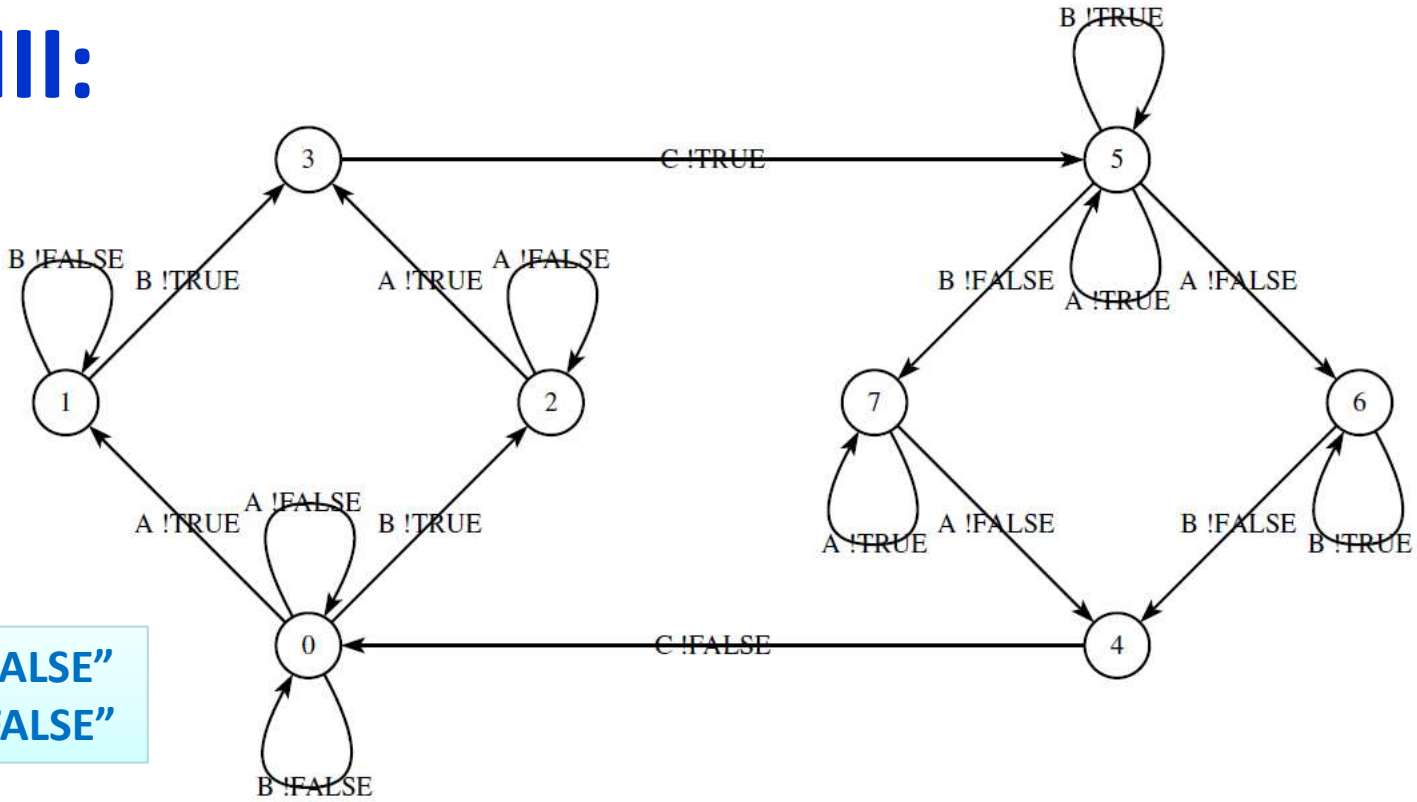


Question I.4: Environment of the circuit

```
process ENV [A, B, C:LINK]
                (in var XA, XB:Bool) is
   loop
      par
         loop LA in
            var XA_NEW:Bool in
               A (?XA_NEW);
               if XA_NEW != XA then
                  XA := XA_NEW;
                  break LA
               end if
            end var
         end loop
   ||  loop LB in
         var XB_NEW:Bool in
            B (?XB_NEW);
            if XB_NEW != XB then
               XB := XB_NEW;
               break LB
            end if
         end var
      end loop
   end par;
   C (?any Bool)
end loop
end process
```

The inputs A and B are allowed to change their value at most once between two consecutive outputs C.

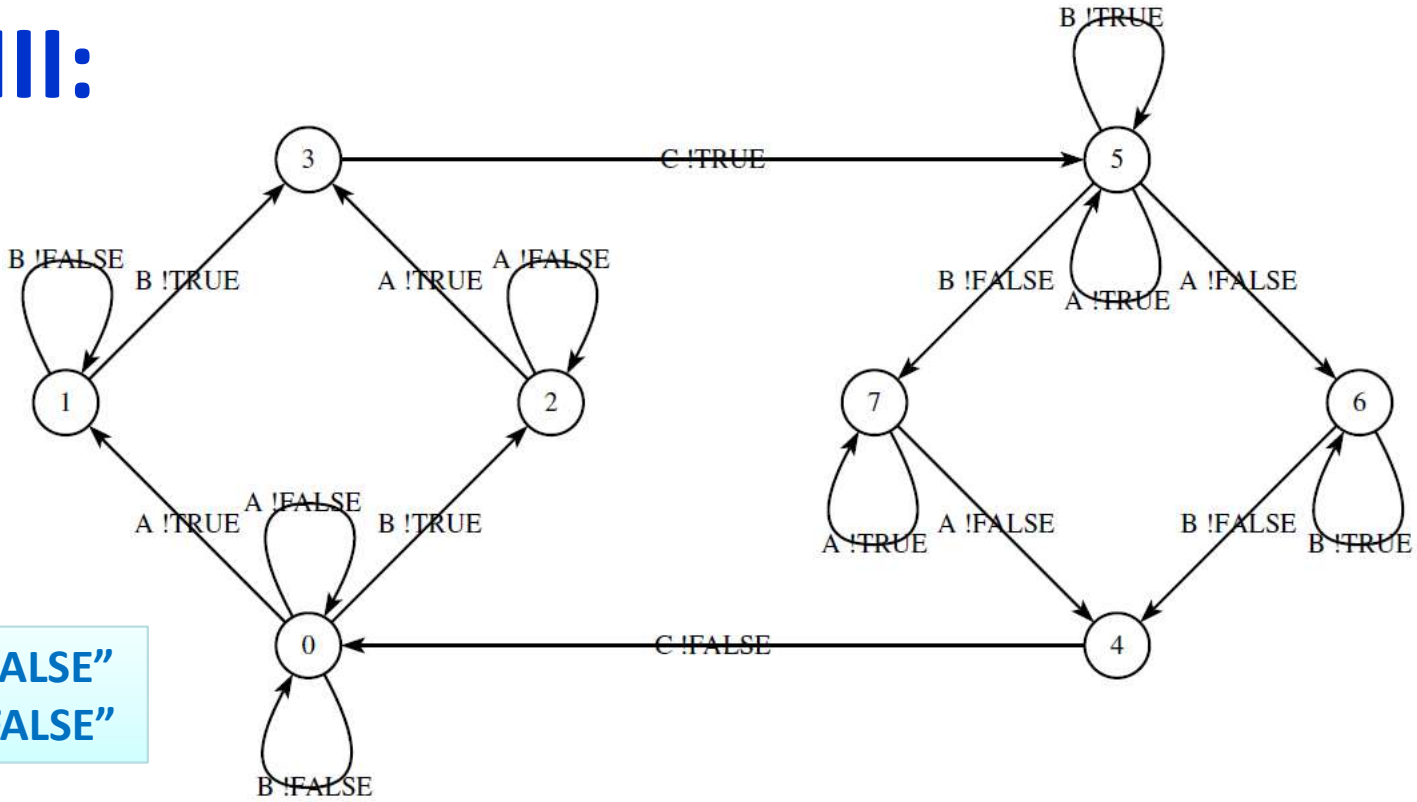
Question III: Temporal logic



A = “A !TRUE” ∨ “A !FALSE”
B = “B !TRUE” ∨ “B !FALSE”

1. ⟨ **true** ⟩ **true** = { 0, ... ,7 } // at least one successor (non deadlock)
2. **[”C !TRUE”] false** = { 0, 1, 2, 4, 5, 6, 7 } // no ”C !TRUE” successors
3. ⟨ **“A !TRUE” ∨ “B !TRUE”** ⟩ **false** = { } // contradiction!
4. **[“A !TRUE” . “B !TRUE” . “C !TRUE”] false** = { 1, 2, 3, 4, 5, 6, 7 }
   // no outgoing sequence “A !TRUE” . “B !TRUE” . “C !TRUE”
5. ⟨ **(*A* ∨ *B*)*. “C !TRUE”** ⟩ **true** = { 0, 1, 2, 3 }
   // some outgoing sequence (*A* ∨ *B*)*. “C !TRUE”

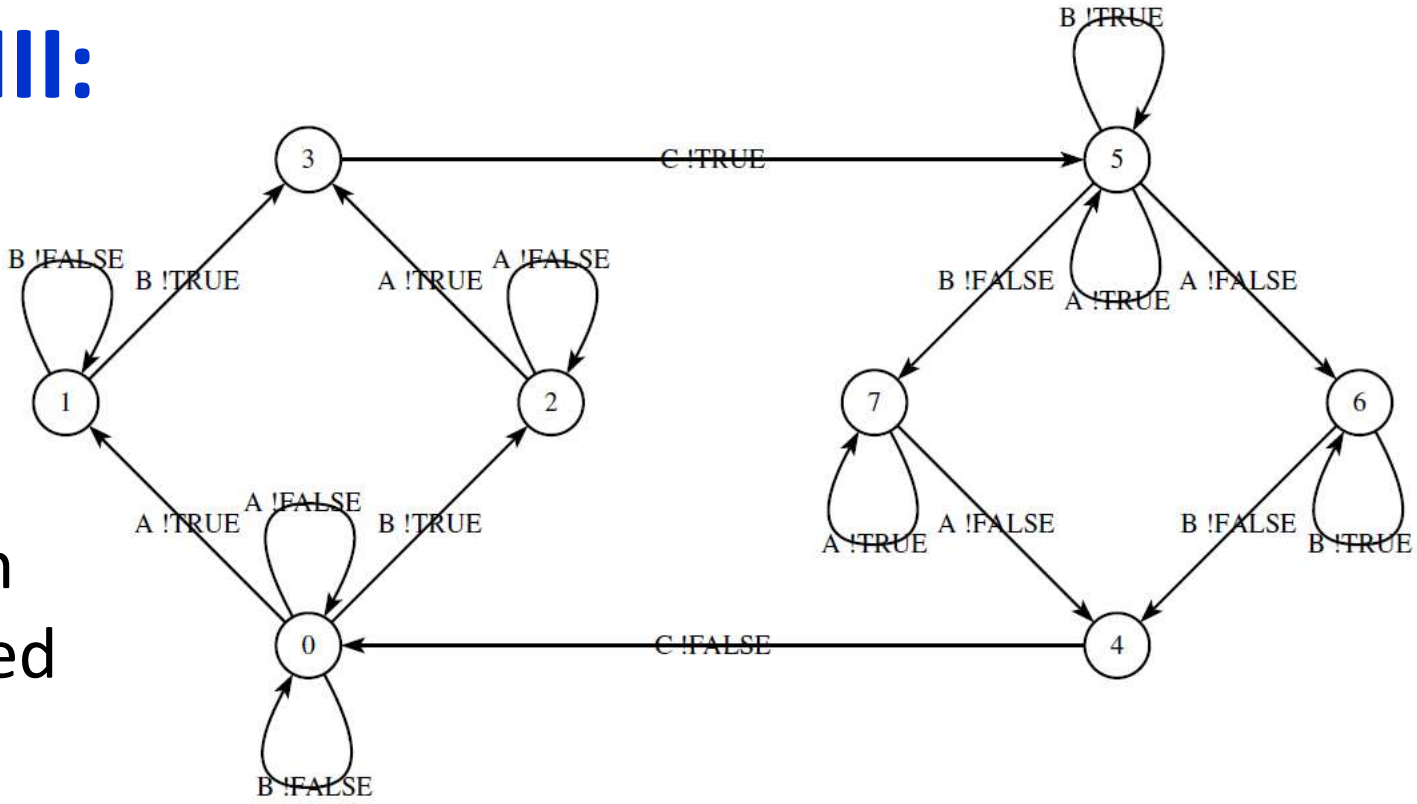
Question III: Temporal logic



A = "A !TRUE" ∨ "A !FALSE"
B = "B !TRUE" ∨ "B !FALSE"

6. $\mu X . \langle \text{true} \rangle \text{true} \wedge [\neg \text{"C !TRUE"}] X = \{ 3 \}$ // inevitable execution of "C !TRUE"
7. $[(\neg \text{"C !TRUE"})^*] \langle \text{true}^* . \text{"C !TRUE"} \rangle \text{true} = \{ 0, ..., 7 \}$ // fair exec. of "C !TRUE"
8. $\nu X . \langle A \rangle X = \{ 0, 2, 5, 7 \}$ // infinite sequence of *A*'s
9. $\mu X . \langle \text{"B !TRUE"} \rangle \text{false} \vee \langle \text{true} \rangle X = \{ \}$
   // contradiction ($\langle$ "B !TRUE" $\rangle$ false = false)!
10. $\nu X . \langle \text{"A !TRUE"} \vee \text{"B !TRUE"} \rangle X = \{ 5, 6, 7 \}$
    // infinite sequence of "A !TRUE" or "B !TRUE"

Question III: Temporal logic



- Iterative computation (minimal fixed point):

$\varphi_6 = \mu X \,.\, \langle\, \text{true} \,\rangle\, \text{true} \wedge [\, \neg \text{"C !TRUE"} \,]\, X$

$X_0 = \{\,\}$ // empty set

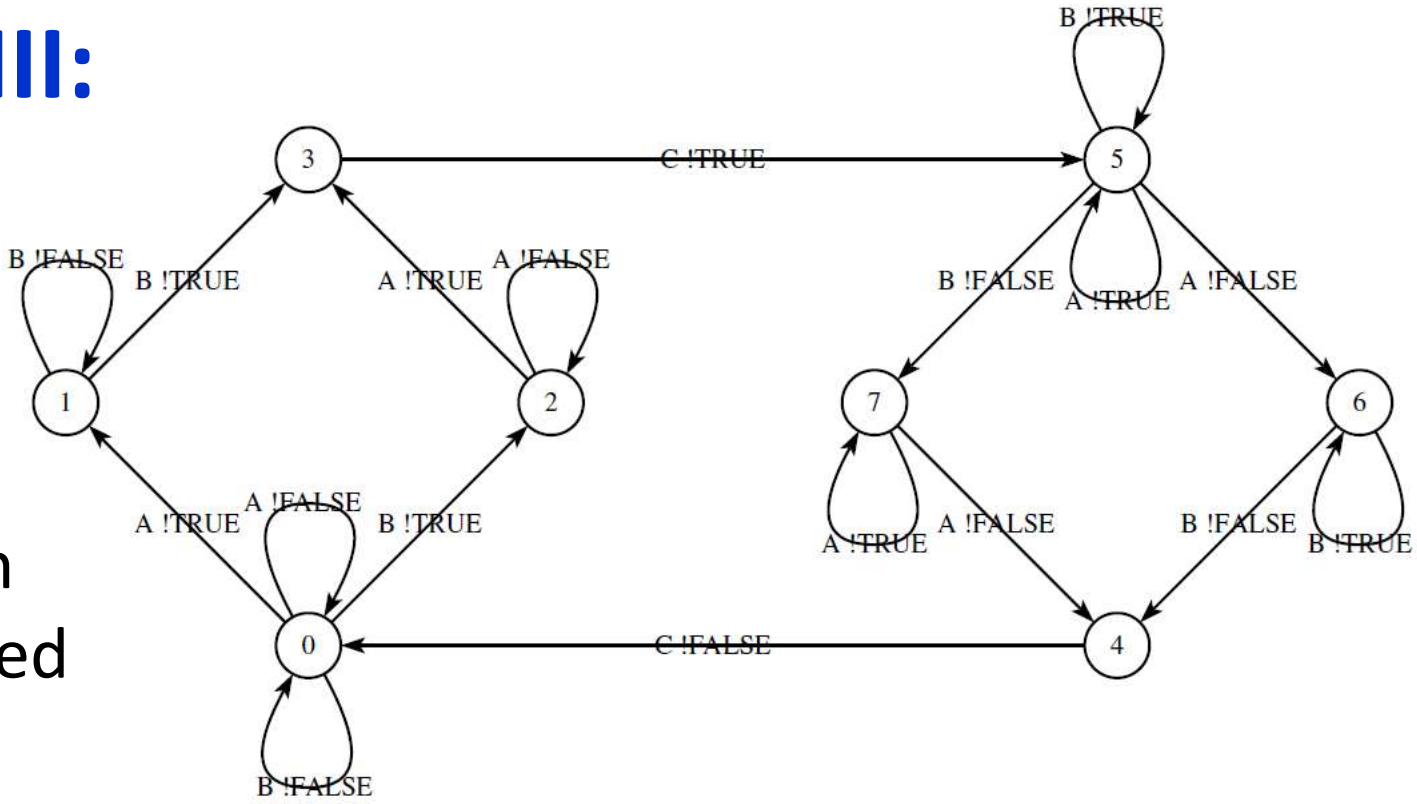
$X_1 = [[\, \langle\, \text{true} \,\rangle\, \text{true} \wedge [\, \neg \text{"C !TRUE"} \,]\, X \,]]\, [\, \{\,\} / X \,] = \{\, 3 \,\}$

// no successors other than "C !TRUE" ones

$X_2 = [[\, \langle\, \text{true} \,\rangle\, \text{true} \wedge [\, \neg \text{"C !TRUE"} \,]\, X \,]]\, [\, \{\, 3 \,\} / X \,] = \{\, 3 \,\}$

// all successors other than "C !TRUE" must lead to state 3

Question III: Temporal logic



- Iterative computation (maximal fixed point):

$\varphi_{10} = \nu X \,.\, \langle$ "A !TRUE" $\vee$ "B !TRUE" $\rangle X$

$X_0 = \{ 0,1,2,3,4,5,6,7 \}$ // full set

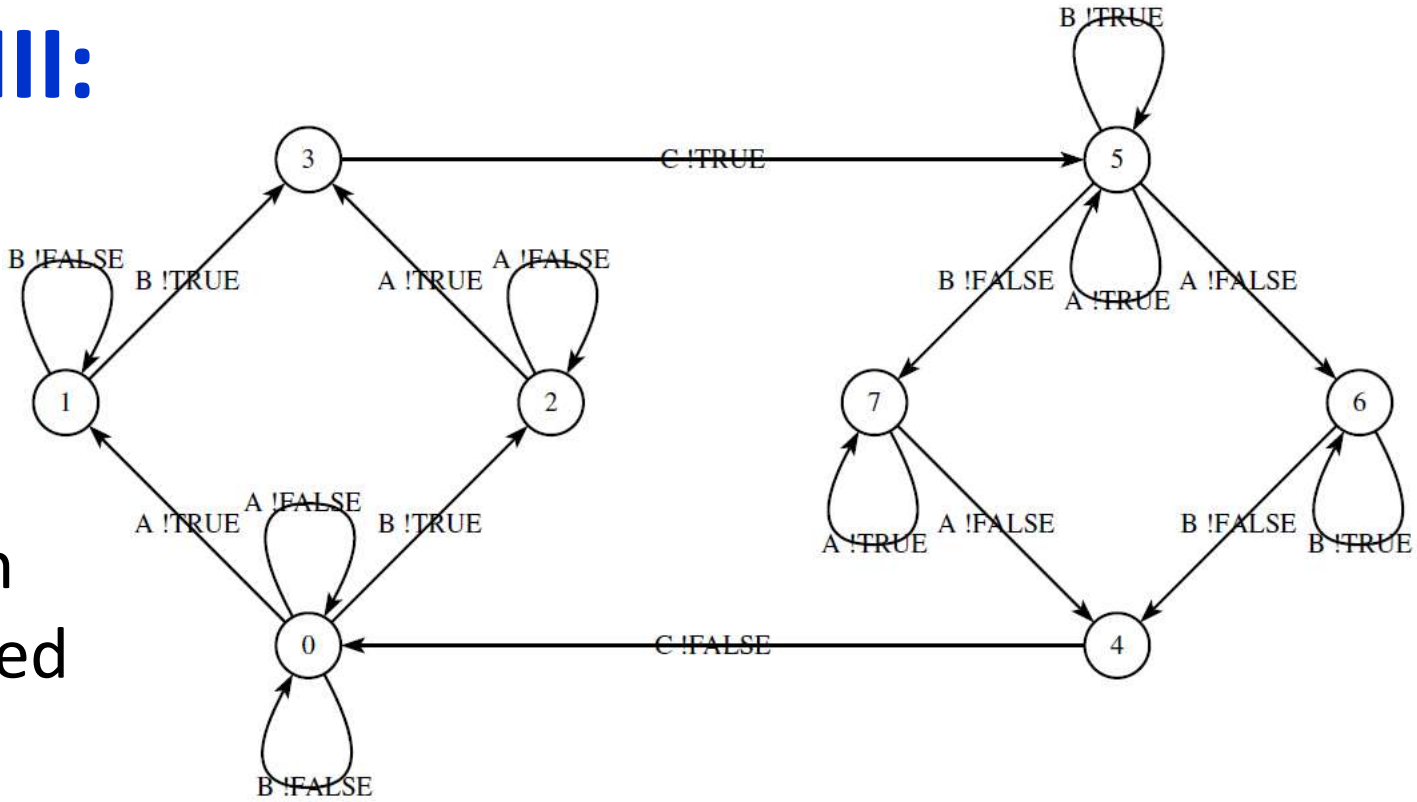
$X_1 = [[\langle$ "A !TRUE" $\vee$ "B !TRUE" $\rangle X]] [\{0,1,2,3,4,5,6,7\} / X] = \{0,1,2,5,6,7\}$

// some successor "A !TRUE" or "B !TRUE"

$X_2 = [[\langle$ "A !TRUE" $\vee$ "B !TRUE" $\rangle X]] [\{0,1,2,5,6,7\} / X] = \{0,5,6,7\}$

// some successor "A !TRUE" or "B !TRUE" leading to 0,1,2,5,6,7

Question III: Temporal logic



- Iterative computation (maximal fixed point):

$\varphi_{10} = \nu X \,.\, \langle$ "A !TRUE" $\vee$ "B !TRUE" $\rangle\, X$

$X_3 = [[\, \langle$ "A !TRUE" $\vee$ "B !TRUE" $\rangle\, X \,]]\, [\{0,5,6,7\} / X] = \{5,6,7\}$

// some successor "A !TRUE" or "B !TRUE" leading to 0,5,6,7

$X_4 = [[\, \langle$ "A !TRUE" $\vee$ "B !TRUE" $\rangle\, X \,]]\, [\{5,6,7\} / X] = \{5,6,7\}$

// some successor "A !TRUE" or "B !TRUE" leading to 5,6,7

// (stabilization)