Exam 2019
(solution – parts I and III)

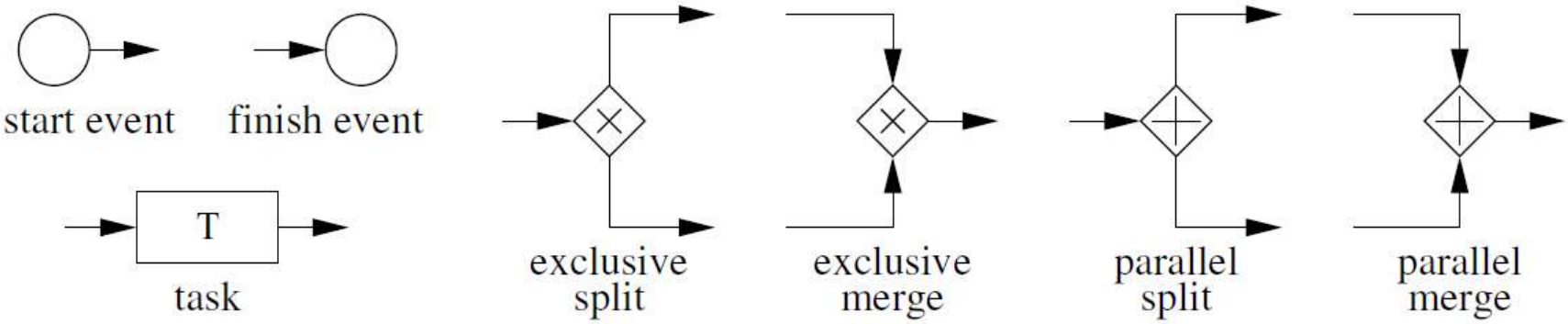
Radu Mateescu

Inria and LIG / Convecs

http://convecs.inria.fr

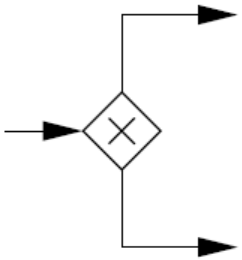
Part I: Modeling in LNT

- Model BPMN (Business Process Modeling Notation) workflows

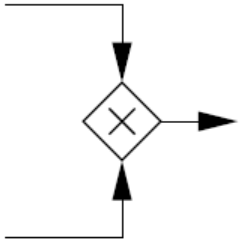


- BPMN semantics: token-based execution of gateways
- LNT modeling:
  - Nodes: cyclic LNT processes
  - Flows: LNT channels

Examples: exclusive gateways

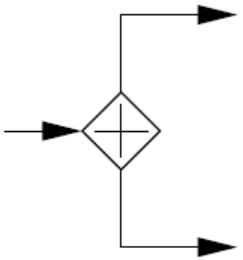

exclusive split

```
process SPLIT_EXCLUSIVE_2 [INPUT, OUTPUT1,
                           OUTPUT2:none] is
    loop
       INPUT; select OUTPUT1 [] OUTPUT2 end select
    end loop
end process
```

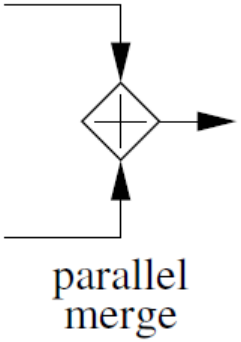

exclusive merge

```
process MERGE_EXCLUSIVE_2 [INPUT1, INPUT2,
                           OUTPUT:none] is
    loop
       select INPUT1 [] INPUT2 end select; OUTPUT
    end loop
end process
```

Question I.1: Parallel gateways

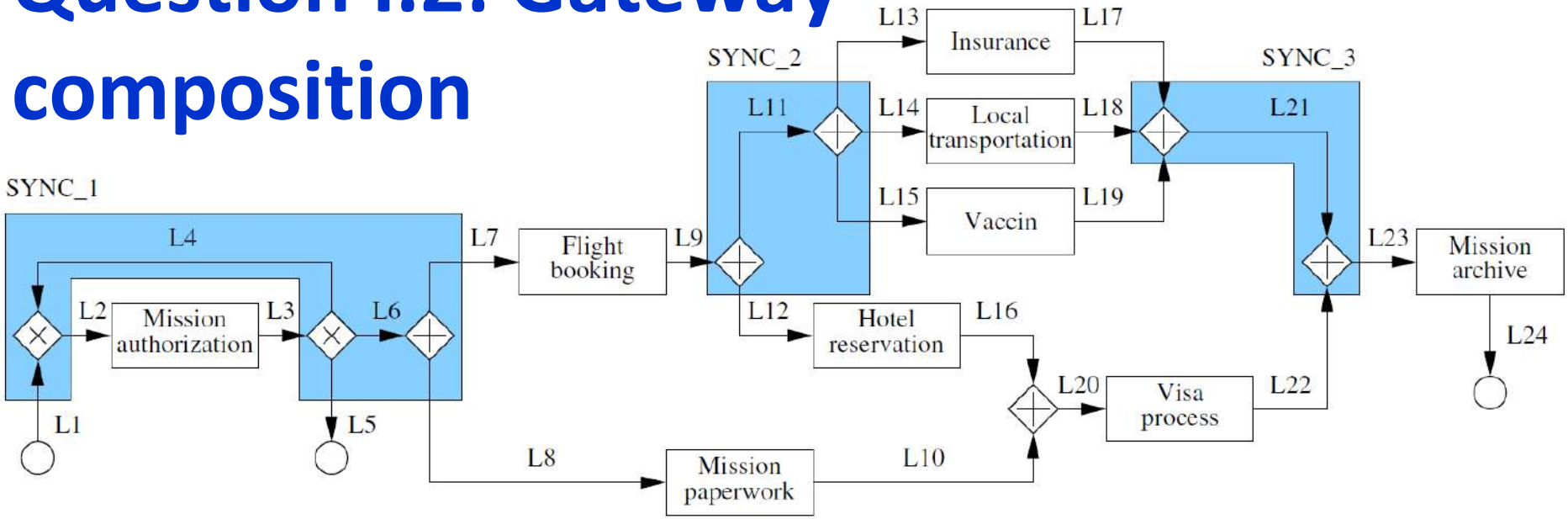



```
process SPLIT_PARALLEL_2 [INPUT, OUTPUT1,
                          OUTPUT2:none] is
   loop
      INPUT; par OUTPUT1 || OUTPUT2 end par
   end loop
end process
```




```
process MERGE_PARALLEL_2 [INPUT1, INPUT2,
                          OUTPUT:none] is
   loop
      par INPUT1 || INPUT2 end par; OUTPUT
   end loop
end process
```

Question I.2: Gateway composition



process SYNC_1 [L1, L2, L3, L5, L7, L8:none] **is**

hide L4, L6:none **in**

par

L4 -> MERGE_EXCLUSIVE_2 [L1, L4, L2]

|| L4, L6 -> SPLIT_EXCLUSIVE_3 [L3, L4, L5, L6]

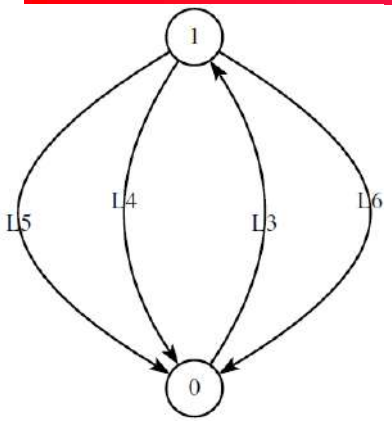
|| L6 -> SPLIT_PARALLEL_2 [L6, L7, L8]

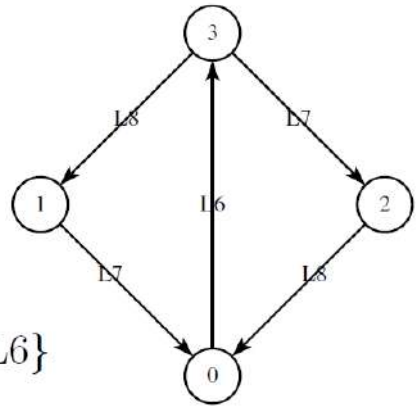
end par

end hide

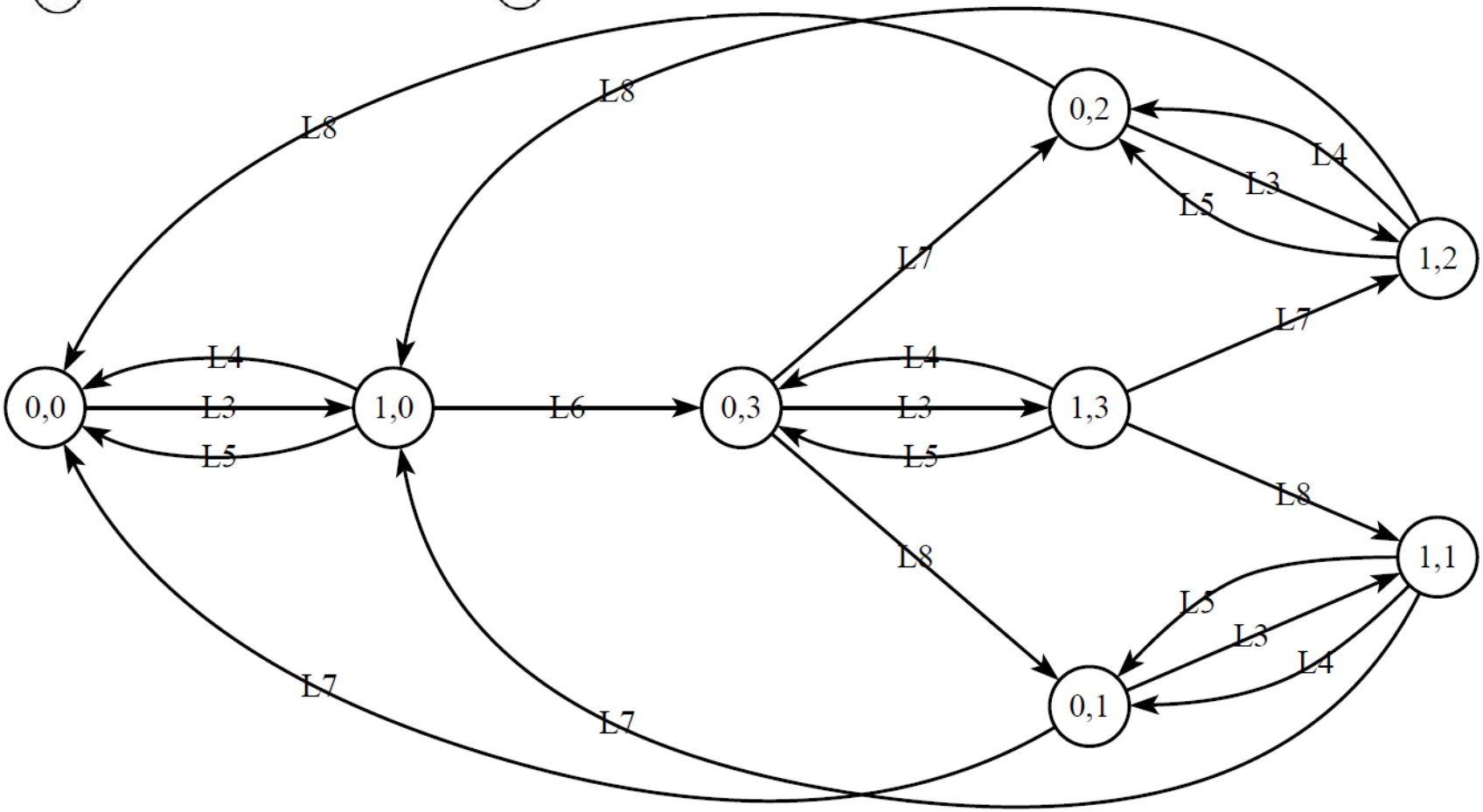
end process

Question I.3: Synchronous product





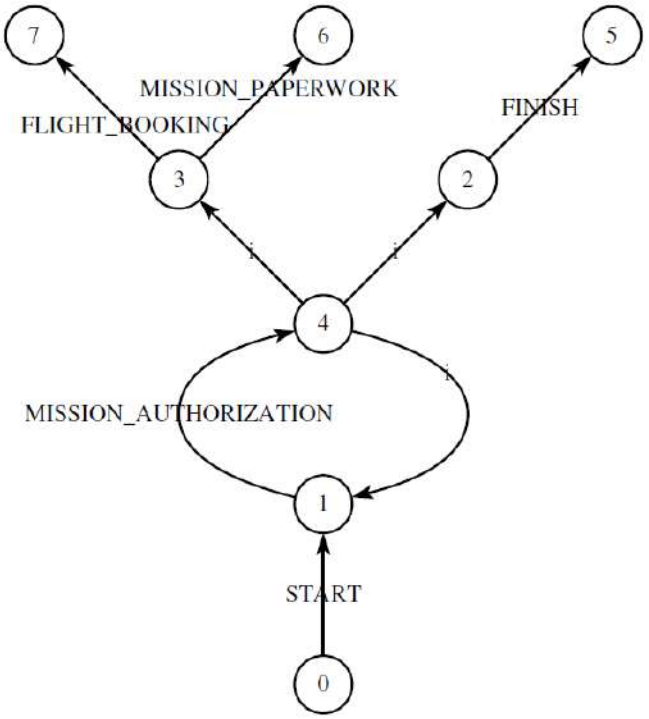
$\otimes \{L6\}$



Question I.4: Global workflow

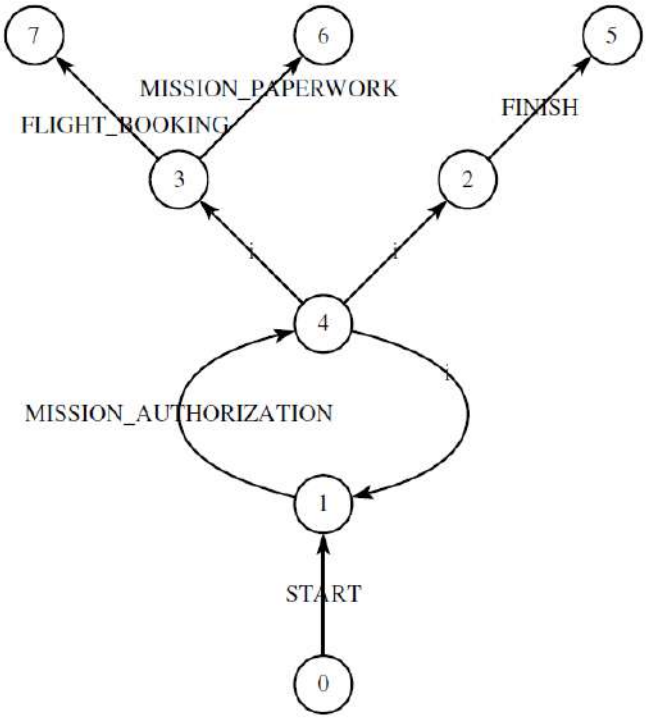
```
process MAIN [START, MISSION_AUTHORIZATION, …, FINISH:none] is
   hide L1-L3, L5, L7-L10, L12-L20, L22-L23:none in
      par L1-L3, L5, L7-L10, L12-L20, L22-L23 in
         par
                  SYNC_1 [L1, L2, L3, L5, L7, L8]
         ||       SYNC_2 [L9, L12, L13, L14, L15]
         ||       SYNC_3 [L17, L18, L19, L22, L23]
         ||       MERGE_PARALLEL_2 [L10, L16, L20]
         end par
      ||          TASKS_EVENTS […]
      end par
   end hide
end process
```

Question III: Temporal logic



1. **[true] false** = { 5, 6, 7 } // no successor (deadlock)
2. ⟨ **START** ∨ **FINISH** ⟩ **true** = { 0, 2 }
   // some START or FINISH successor
3. **[MISSION_AUTHORIZATION] true** = { 0, …, 7 }
   // tautology!
4. ⟨ **true*. FINISH** ⟩ **true** = { 0, 1, 2, 4 }
   // some sequence containing FINISH
5. **[true*. FLIGHT_BOOKING] false** = { 2, 5, 6, 7 }
   // cannot reach FLIGHT_BOOKING
6. **[(¬FINISH)*]** ⟨ **true*. FINISH** ⟩ **true** = { 0, 1, 2, 4 }
   // fair execution of FINISH
7. **μ*X* . [true] *X*** = { 2, 3, 5, 6, 7 } // all outgoing sequences are finite
8. **μ*X* .** ⟨ **true** ⟩ **true** ∧ **[¬FINISH] *X*** = { 2 } // inevitable execution of FINISH
9. **ν*X* . [FINISH] true** ∧ **[true] *X*** = { 0, …, 7 } // tautology!
10. **ν*X* .** ⟨ **MISSION_AUTHORIZATION . true** ⟩ ***X*** = { 1 } // infinite outgoing sequence

Question III: Temporal logic



- Iterative computation (minimal fixed point):

$\varphi_7 = \mu X . [\ \mathbf{true}\ ]\ X$

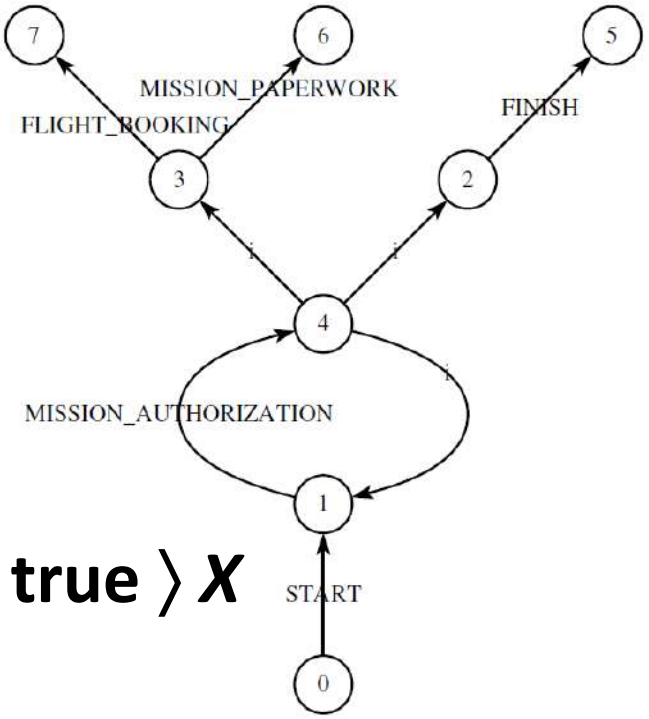
$X_0 = \{\ \}$ // empty set

$X_1 = [[\ [\ \text{true}\ ]\ X\ ]]\ [\ \{\ \}\ /\ X\ ] = \{\ 5, 6, 7\ \}$
// deadlock states

$X_2 = [[\ [\ \text{true}\ ]\ X\ ]]\ [\ \{\ 5, 6, 7\ \}\ /\ X\ ] = \{\ 2, 3, 5, 6, 7\ \}$
// all successors must lead to states 5, 6, or 7

$X_2 = [[\ [\ \text{true}\ ]\ X\ ]]\ [\ \{\ 2, 3, 5, 6, 7\ \}\ /\ X\ ] = \{\ 2, 3, 5, 6, 7\ \}$
// all successors must lead to states 2, 3, 5, 6, or 7
// (stabilization)

Question III: Temporal logic



- Iterative computation (maximal fixed point):

$\varphi_{10} = \nu X \,.\, \langle$ **MISSION_AUTHORIZATION . true** $\rangle X$

$X_0 = \{ 0, \ldots, 7 \}$ // full set

$X_1 = [[\langle \text{MISSION\_AUTHORIZATION} \,.\, \text{true} \rangle X]] [\{ 0, \ldots, 7 \} / X] = \{ 1 \}$

// states with an outgoing 2-step sequence starting with
// MISSION_AUTHORIZATION

$X_2 = [[[\text{true}] X]] [\{ 1 \} / X] = \{ 1 \}$

// states with an outgoing 2-step sequence starting with
// MISSION_AUTHORIZATION and leading to state 1
// (stabilization)