Asynchronous games on Petri nets

Federica Adobbati

DISCo, Università degli studi di Milano-Bicocca

MOVEP June 22, 2020

Concurrent systems

- A user interacts with an environment
- The user knows the structure of the system
- The user may not observe everything
- The user has a goal (perform an action, avoid a state, ...)
- The environment is hostile or indifferent

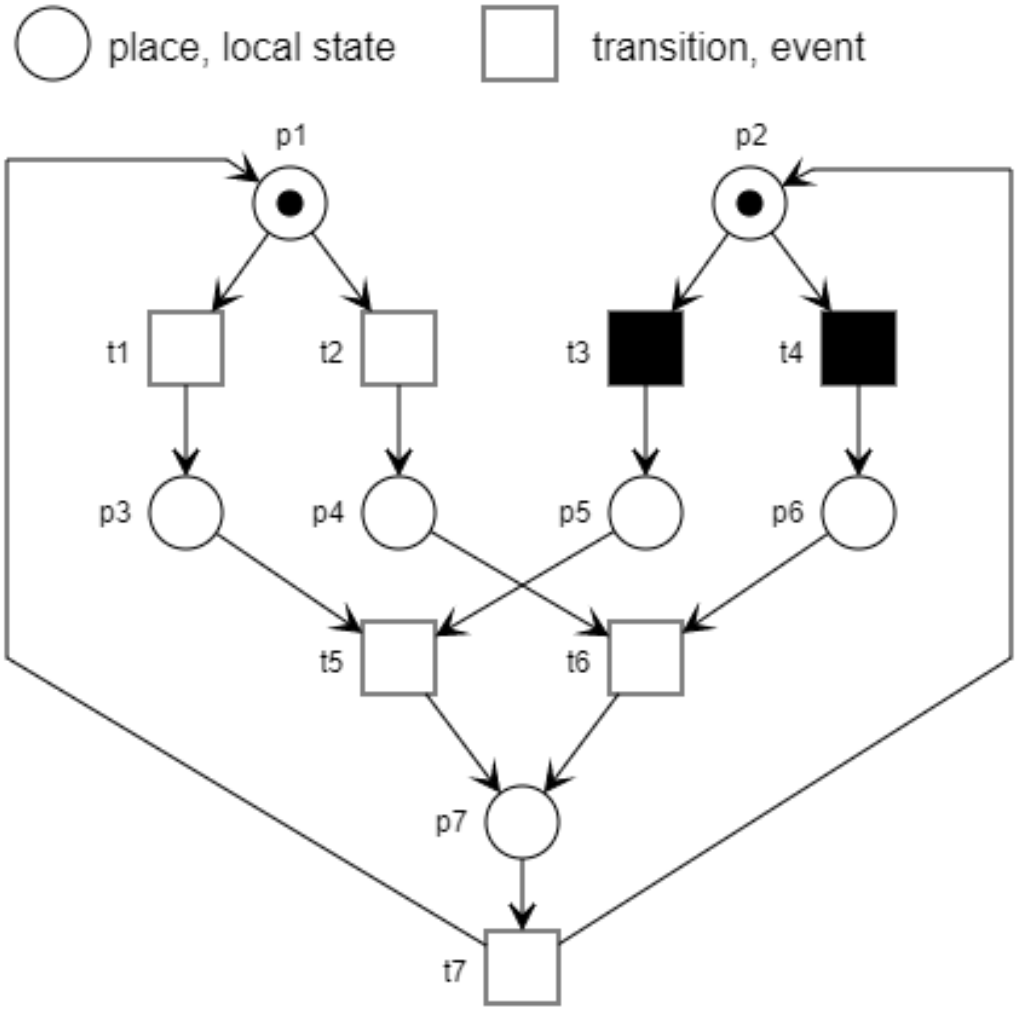
Concurrent systems

- A user interacts with an environment
- The user knows the structure of the system
- The user may not observe everything
- The user has a goal (perform an action, avoid a state, ...)
- The environment is hostile or indifferent

Applications

System control, model-checking, ...

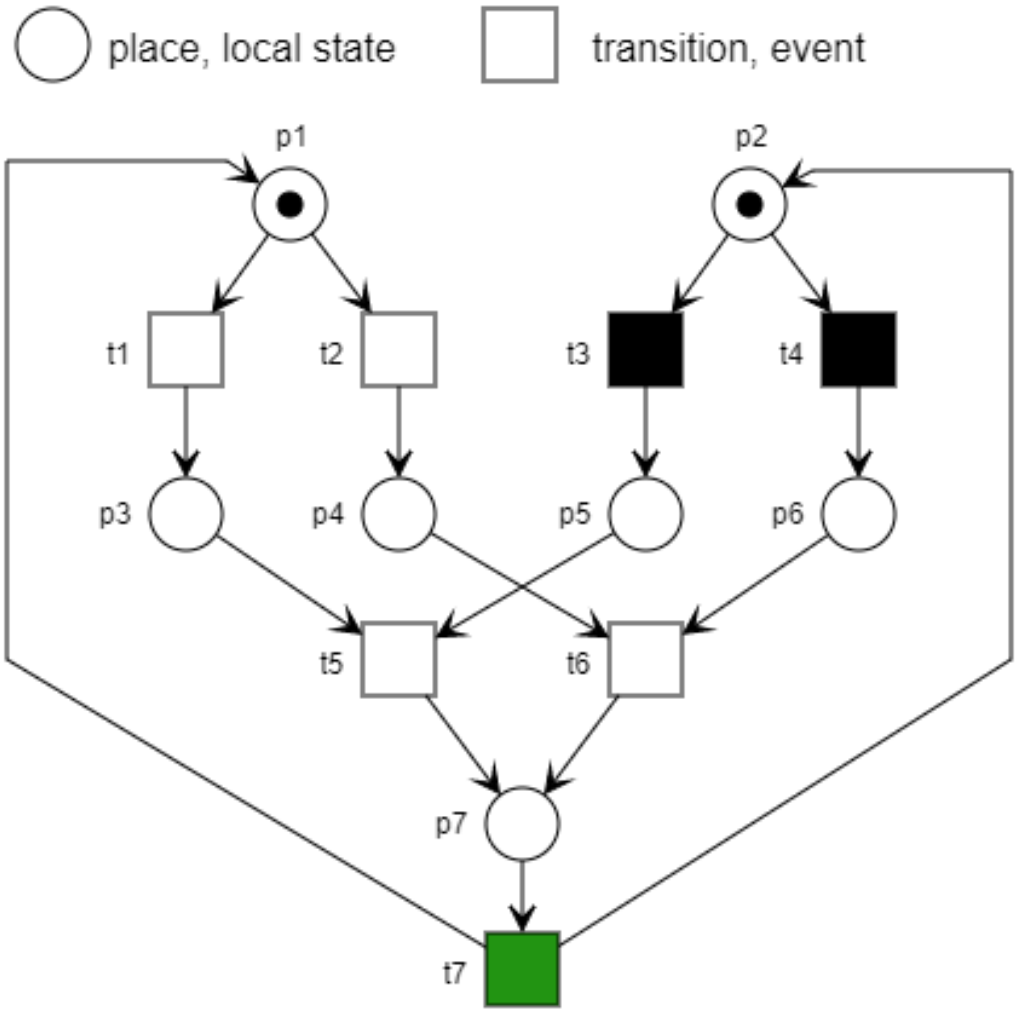
Modelling language: Petri nets



1-safe distributed nets

□ uncontrollable
■ controllable

Modelling language: Petri nets



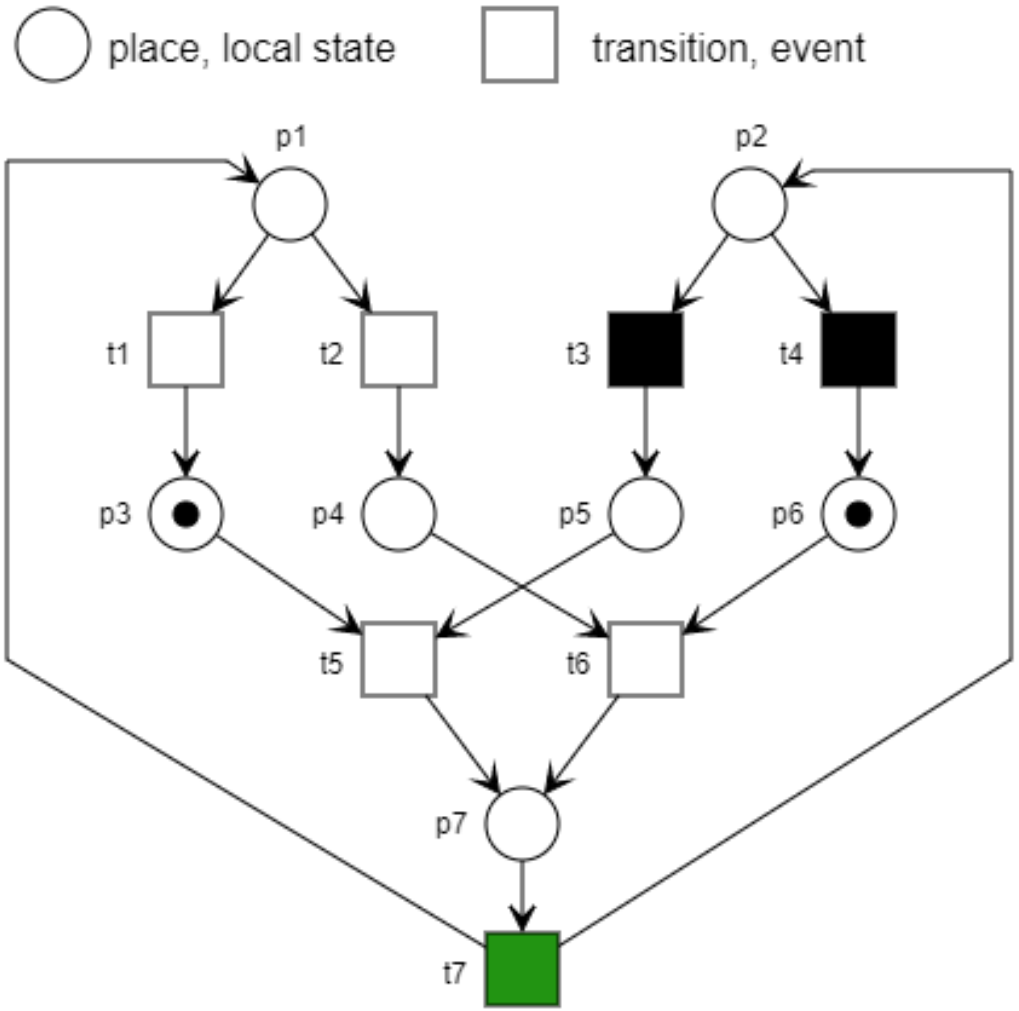
1-safe distributed nets

- ☐ uncontrollable
- ■ controllable

Example

Can the user force the occurrence of transition t_7, by controlling only black transitions?

Modelling language: Petri nets



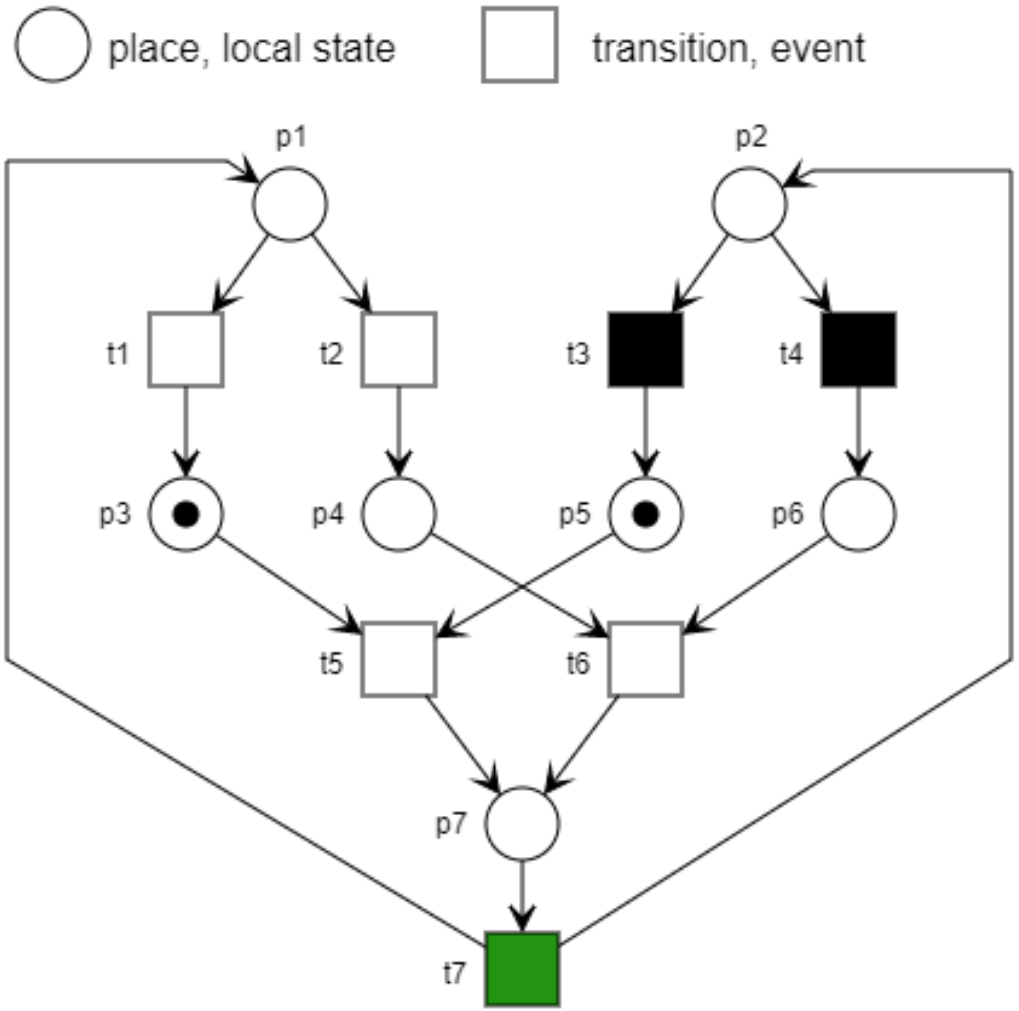
1-safe distributed nets

- ☐ uncontrollable
- ■ controllable

Example

Can the user force the occurrence of transition t_7, by controlling only black transitions?

Modelling language: Petri nets



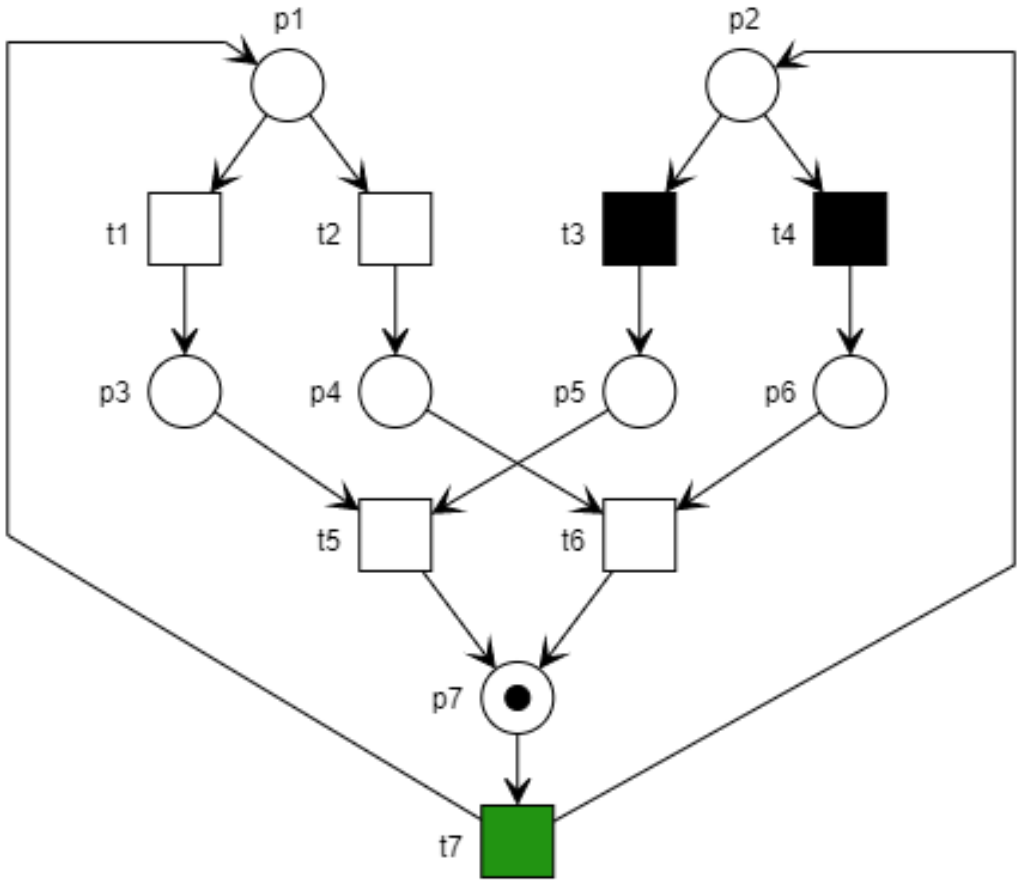
1-safe distributed nets

- ☐ uncontrollable
- ■ controllable

Example

Can the user force the occurrence of transition t_7, by controlling only black transitions?

Modelling language: Petri nets



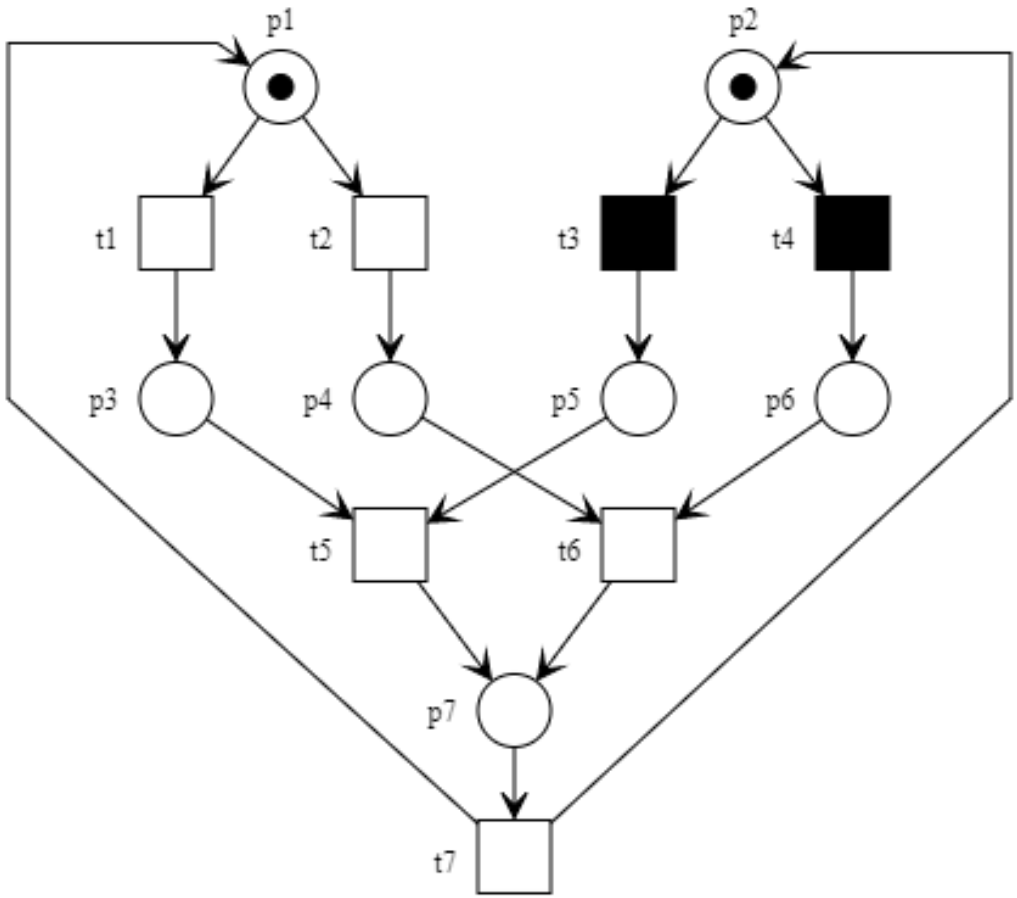
1-safe distributed nets

- ☐ uncontrollable
- ■ controllable

Example

Can the user force the occurrence of transition t_7, by controlling only black transitions?

Asynchronous game on Petri nets



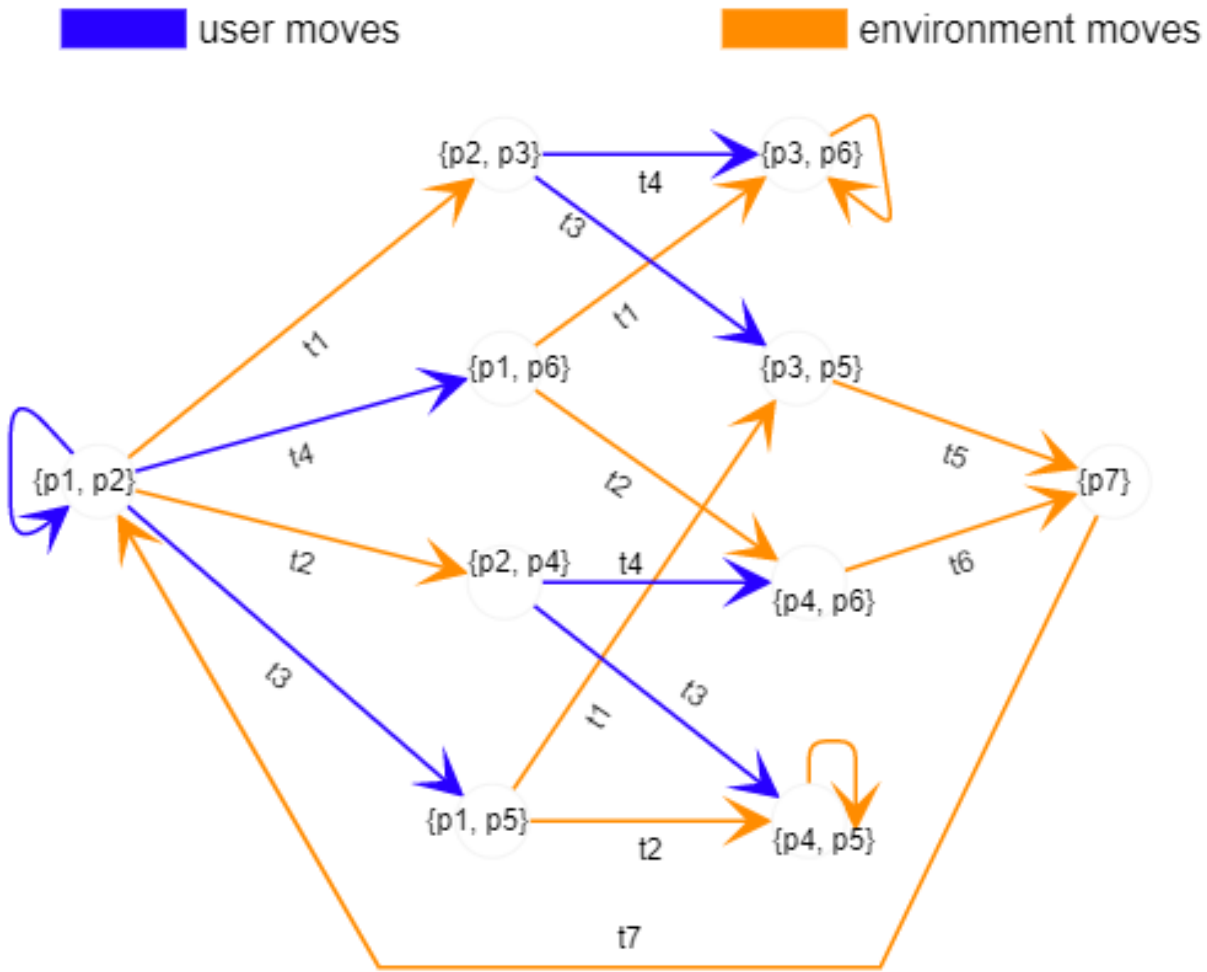
Rules

- Whenever a transition is enabled, its owner can decide to 'fire' it
- The Environment must guarantee the progress of the system

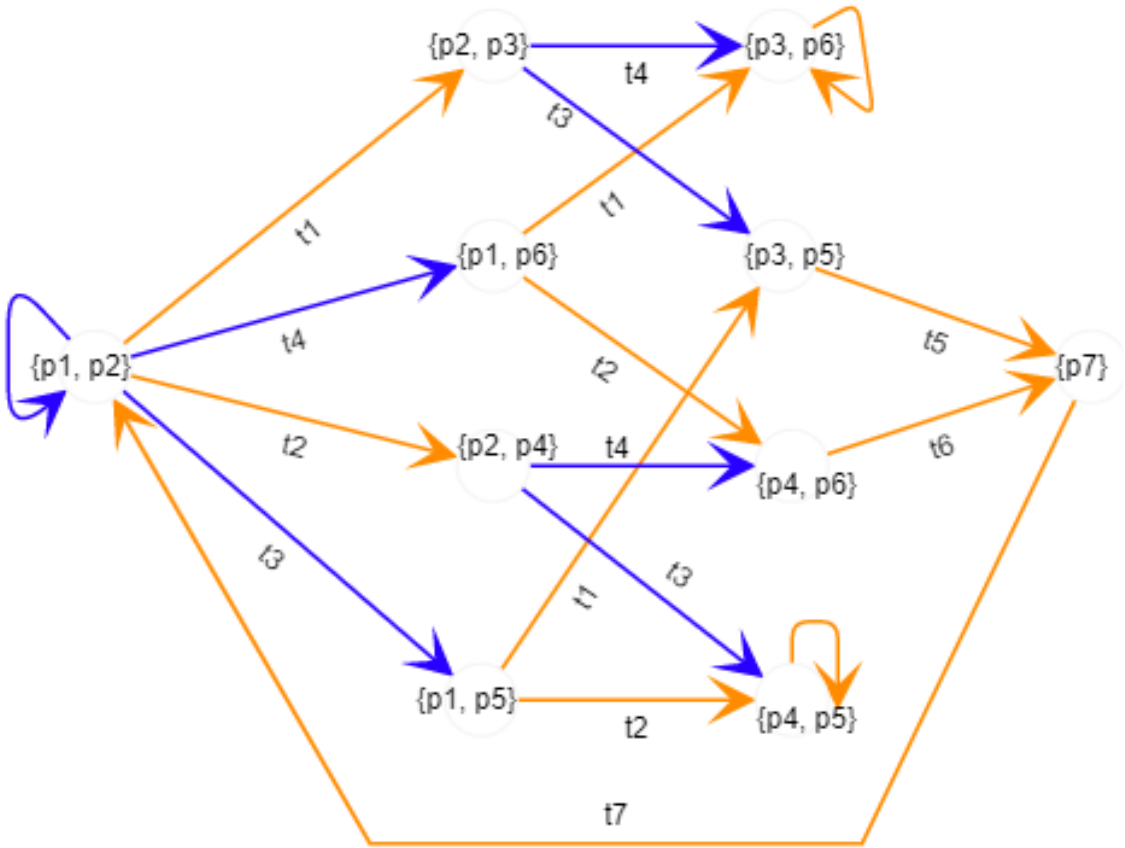
Strategy

$\alpha : Mark(N) \rightarrow 2^K$

Asynchronous game on concurrent game structure



ATL (Alur, Henzinger, Kupferman 2002)



Syntax

An ATL formula is one of the following:

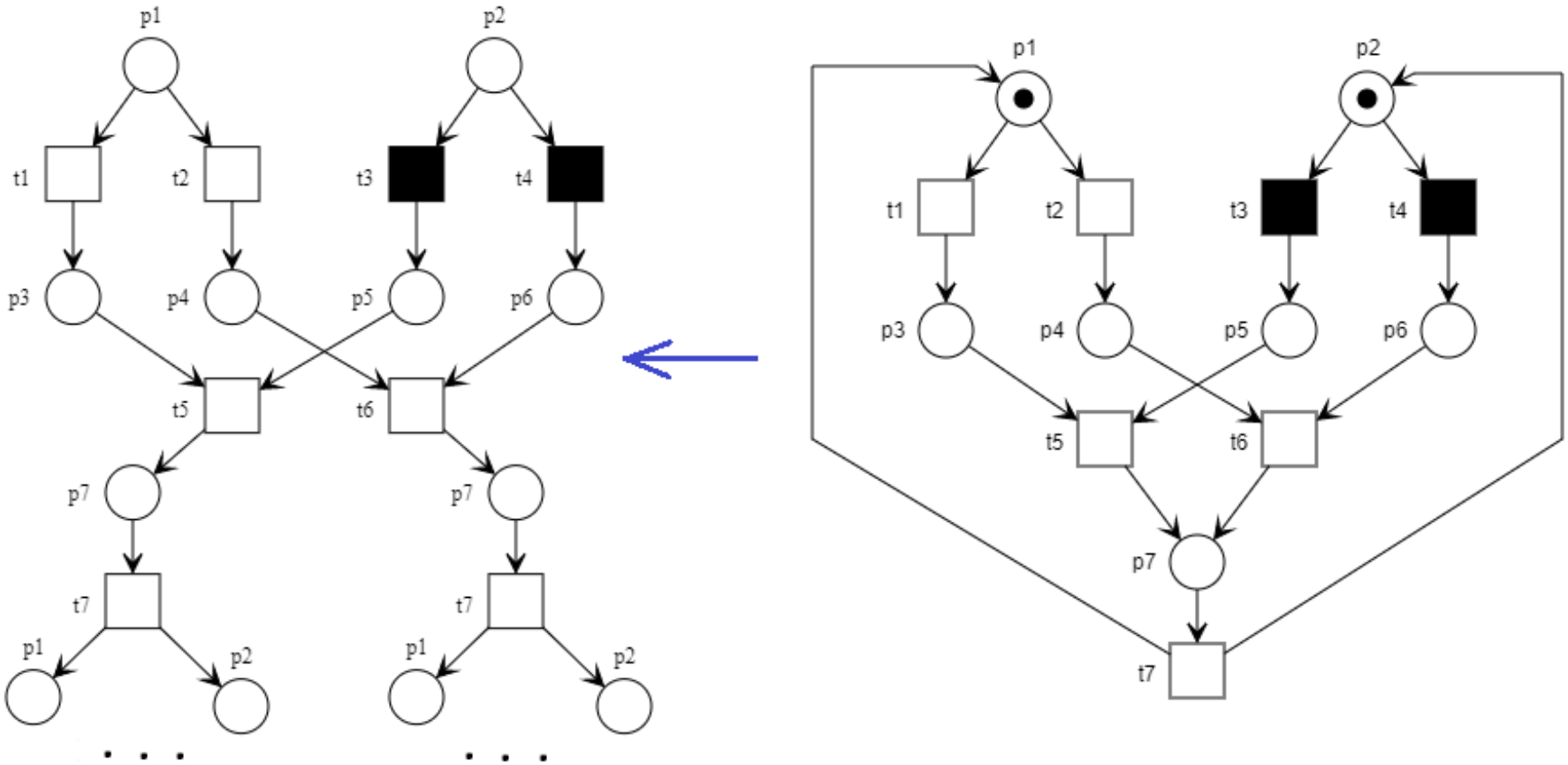
- a proposition p
- $\neg\phi$ or $\phi_1 \vee \phi_2$
- $\langle\langle A\rangle\rangle X\phi$, $\langle\langle A\rangle\rangle G\phi$, $\langle\langle A\rangle\rangle F\phi$, or $\langle\langle A\rangle\rangle \phi_1 U \phi_2$, with A set of players

ϕ, ϕ_1, ϕ_2 ATL formulas

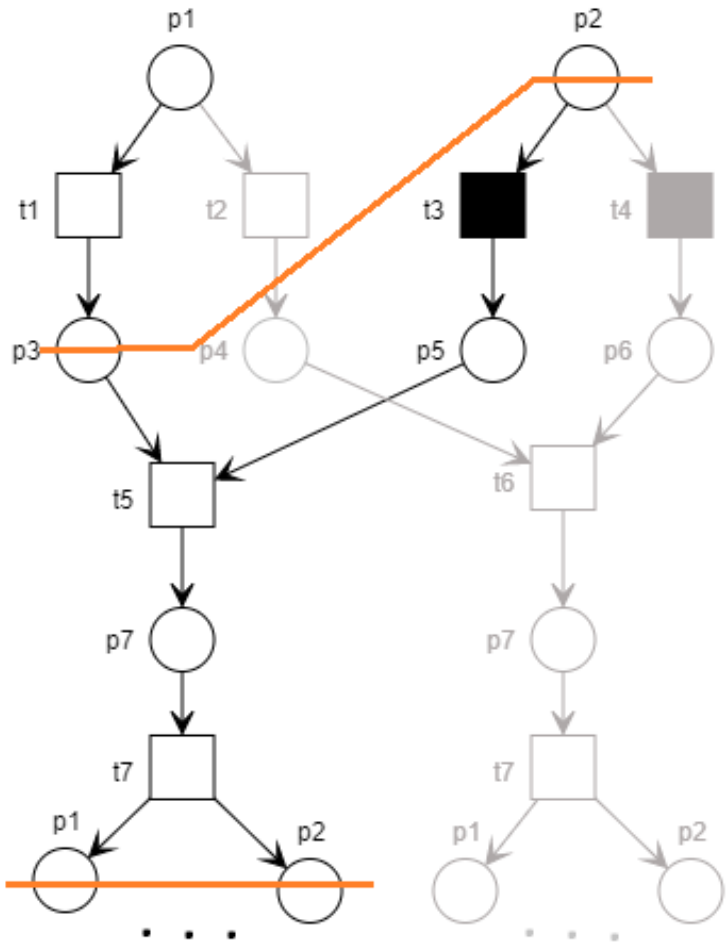
Example

Is $\langle\langle user\rangle\rangle Fp_7$ satisfied on this system?

Game on the unfolding

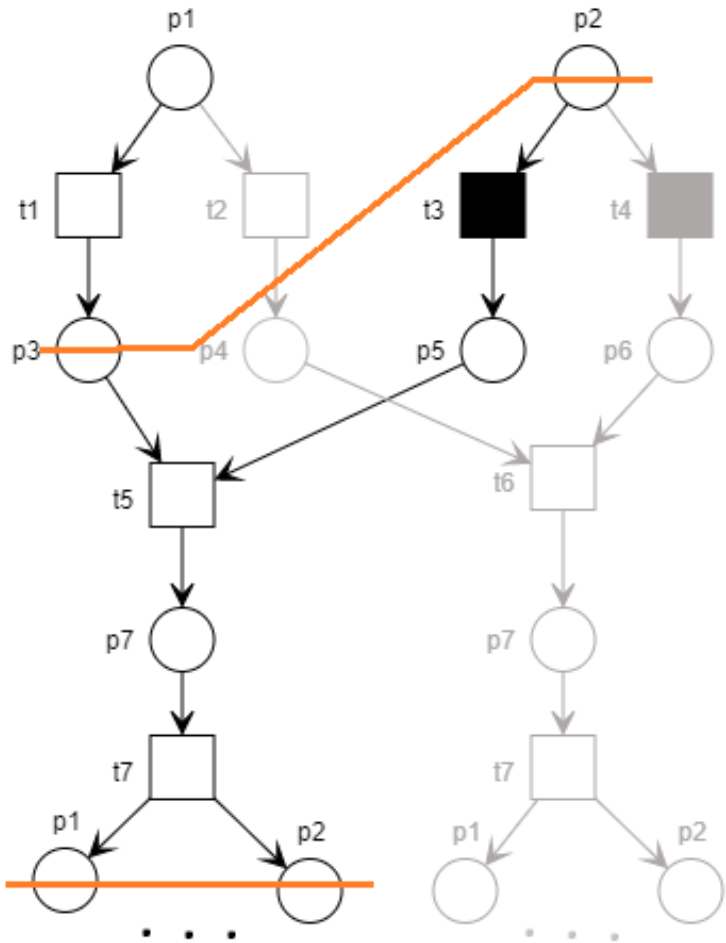


Game on the unfolding



- Run: a possible history of what happened in the net
- B-cut: a maximal set of pairwise concurrent places

Game on the unfolding



- Run: a possible history of what happened in the net
- B-cut: a maximal set of pairwise concurrent places

- Play: a run on the unfolding
- Strategy: $\alpha : Mark(N) \rightarrow 2^K$

The user has a **winning strategy** iff he wins all the plays that are consistent with the strategy

Algorithm for the reachability of a target transition

Goal

The user reaches a target transition

General idea

- Recursive algorithm on the unfolding, with backtracking
- Generation of a finite prefix of the unfolding
- In every marking, the strategy selects all the controllable transitions that are 'useful' to reach the target

Mechanism of the algorithm

Unfolding_Exploration

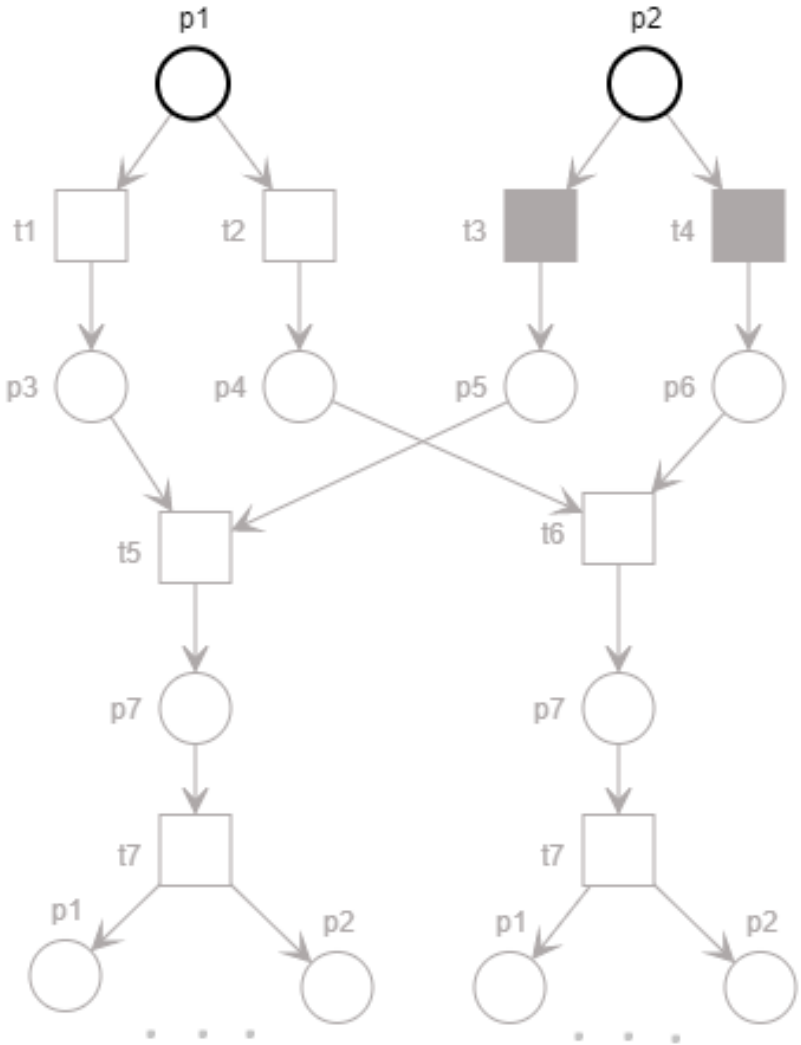
From B-cut γ add events until:

- Deadlock reached (the user tries to backtrack)
- Target fires (the environment tries to backtrack)
- Cycle detected (either no winning strategy or go to Explore_Cycle)

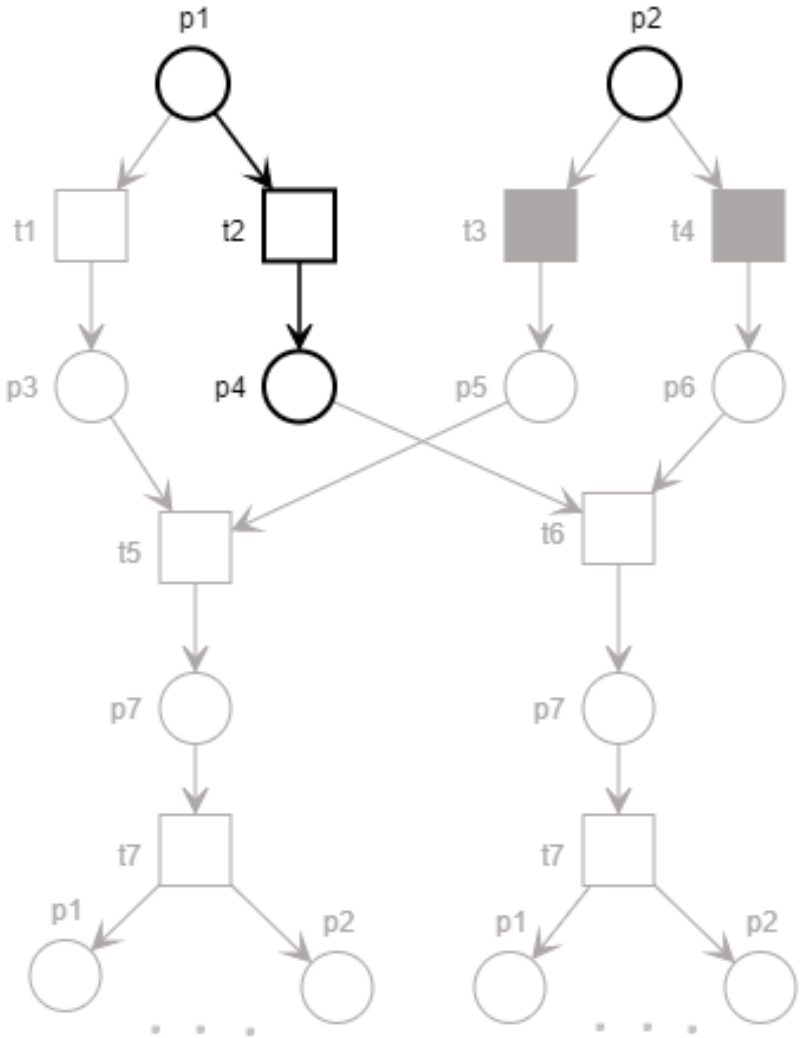
Explore_Cycle

- Add a transition concurrent with the cycle
- Restart with Unfolding_Exploration

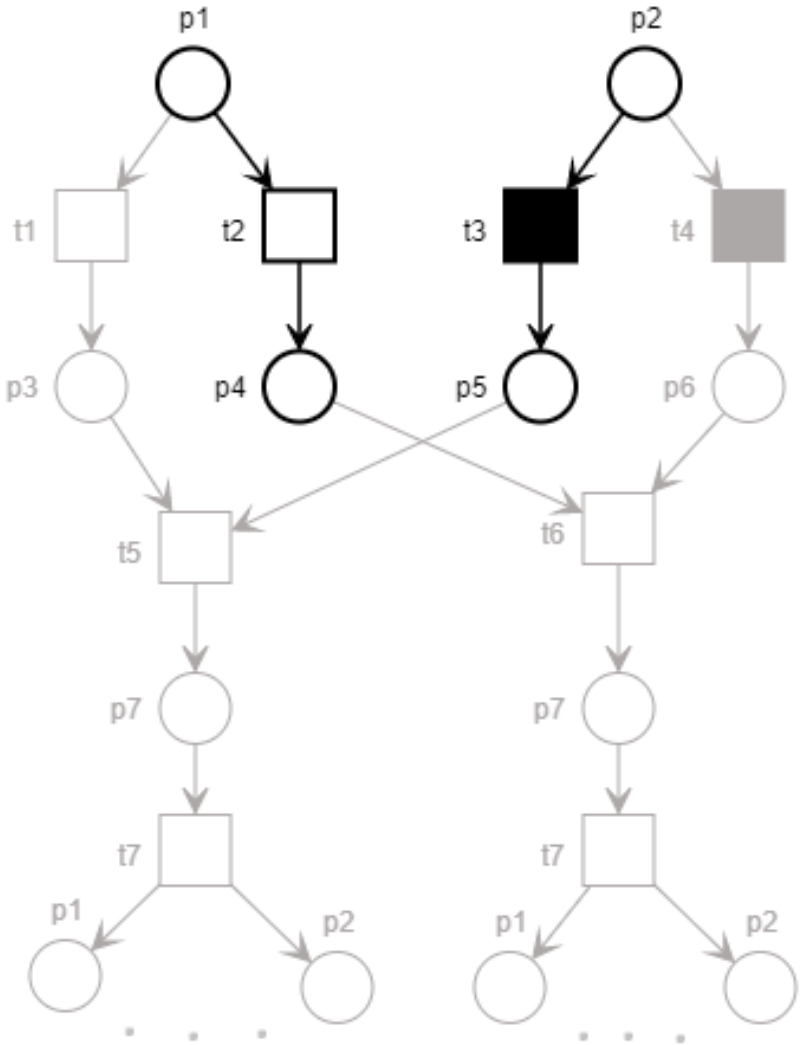
Simulation of an execution



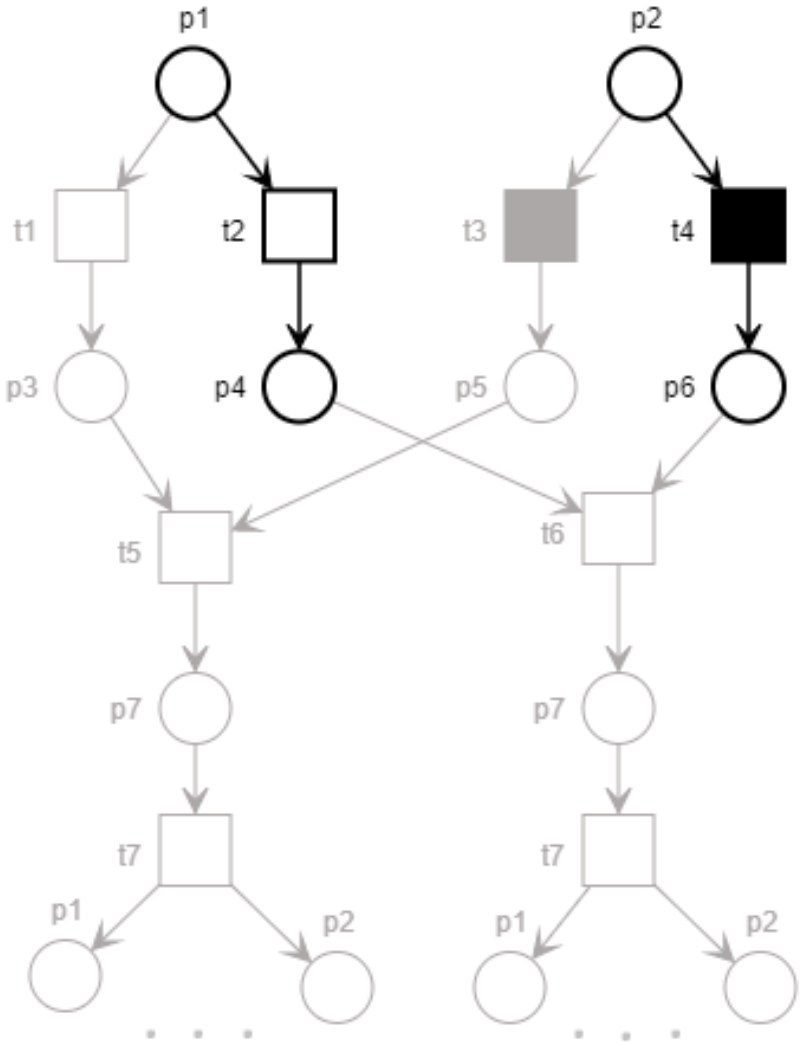
Simulation of an execution



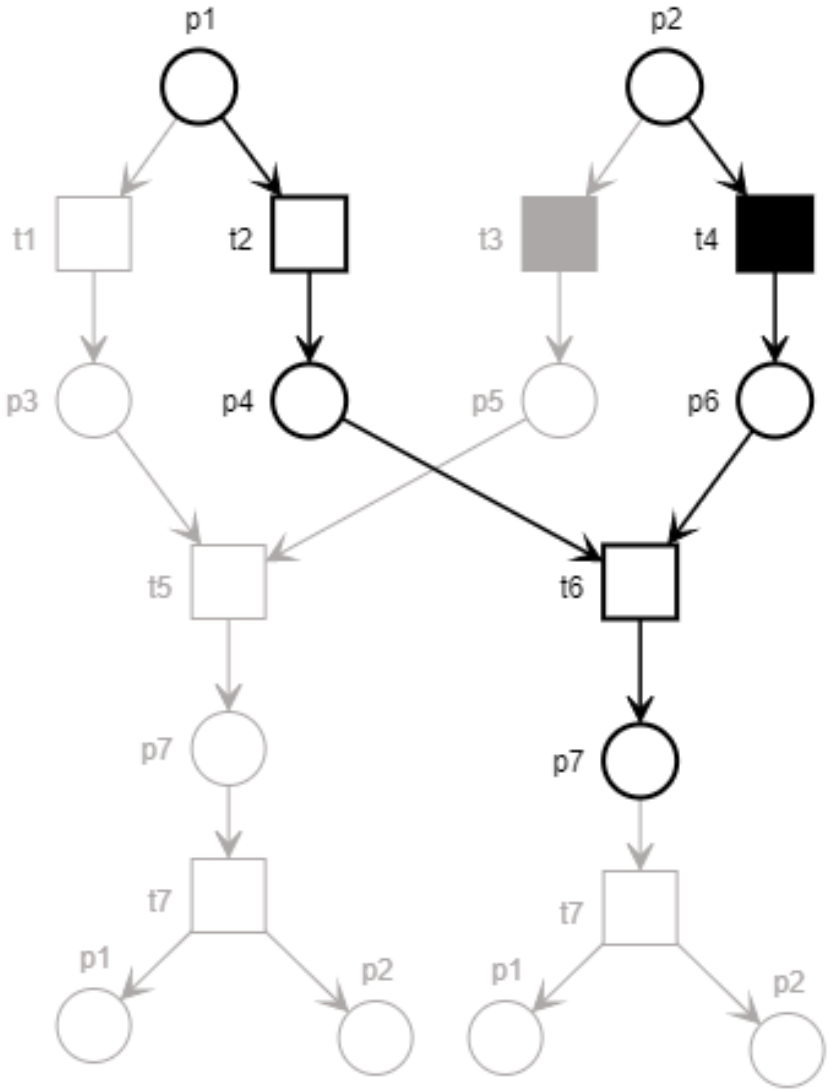
Simulation of an execution



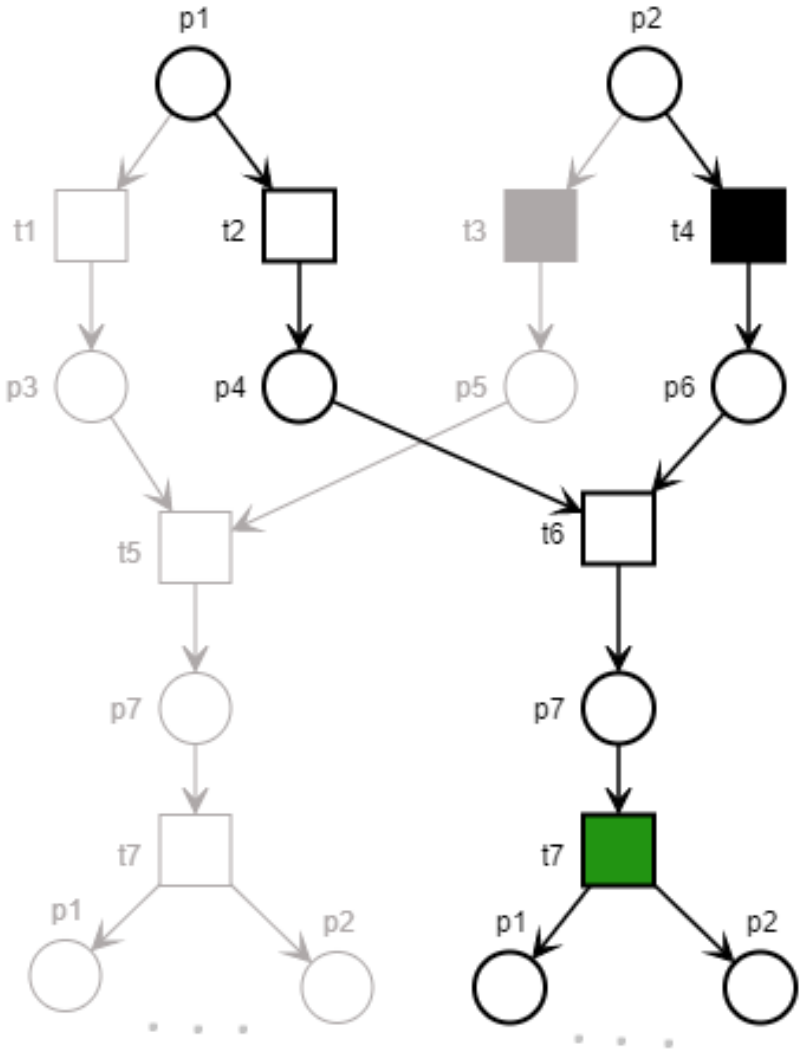
Simulation of an execution



Simulation of an execution

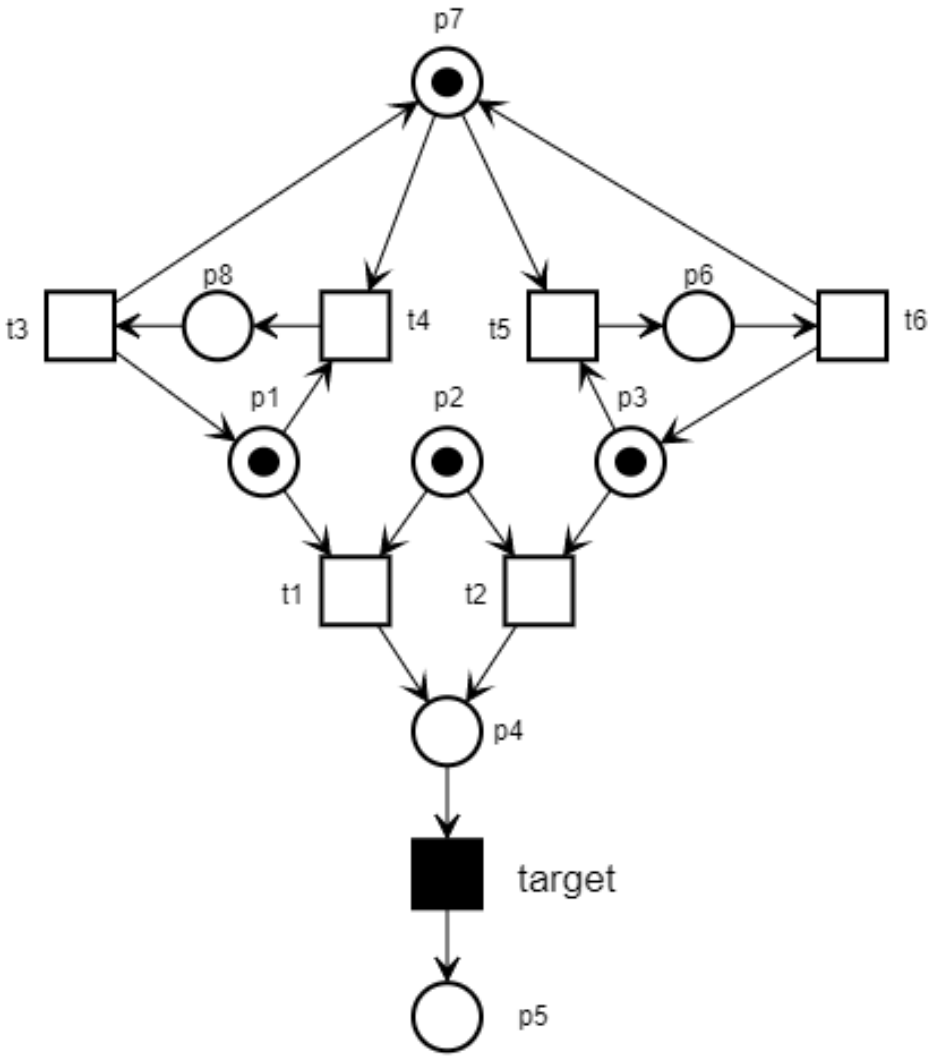


Simulation of an execution



Critical example

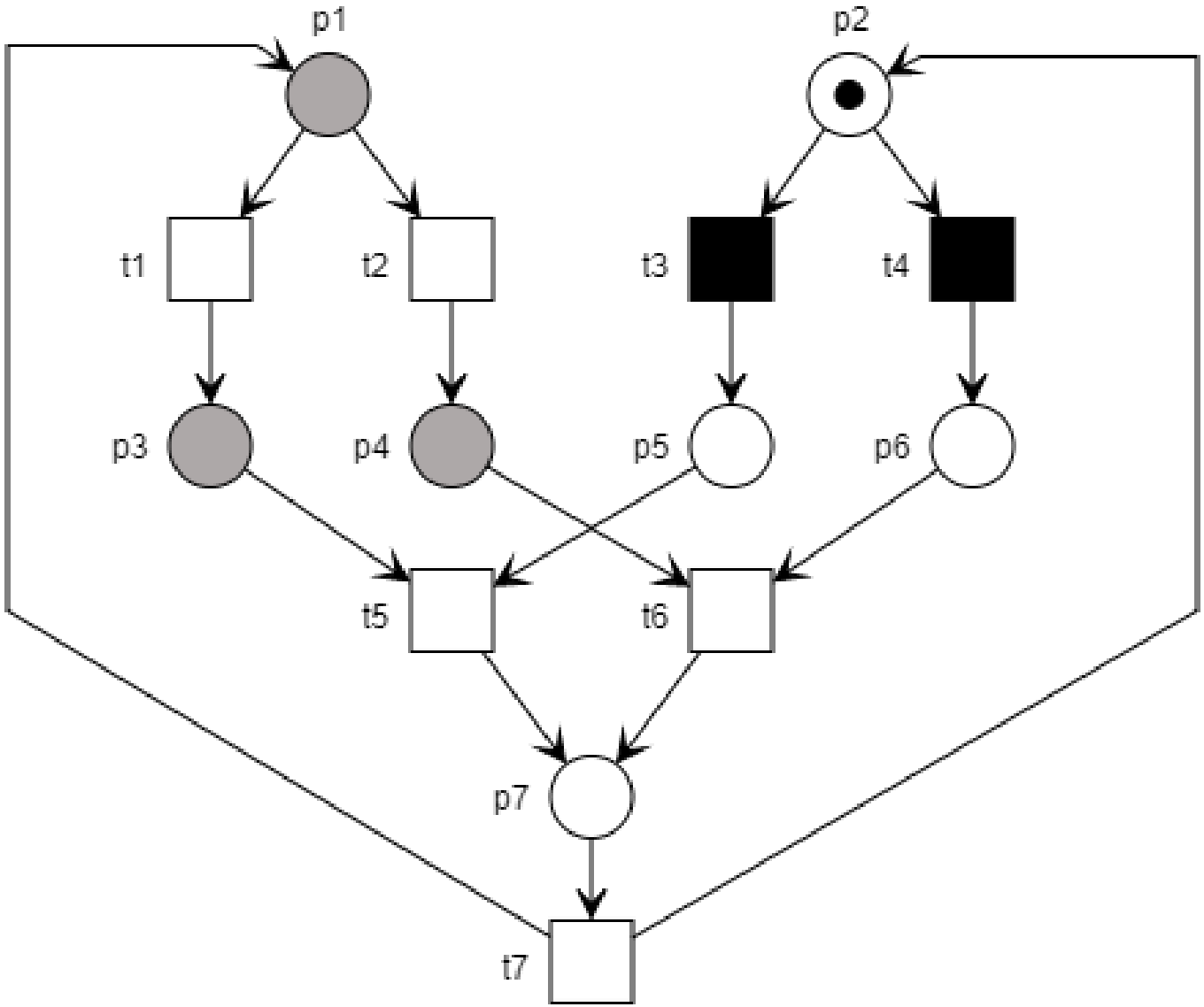
Solved with backtracking



Ongoing works and future developments

- Correctness and complexity of the algorithm
- Partial observability
- Implementability of the strategy

Partial observability



Implementable strategies