COMP201 – Software Engineering I
Lecture 10
Petri Nets

Lecture: T. Carroll

Email: Thomas.Carroll2@Liverpool.ac.uk

Office: G.14

See Vital for all notes

Today

Overview

- Recap of Petri Net Basics
- Modelling Using Petri Nets

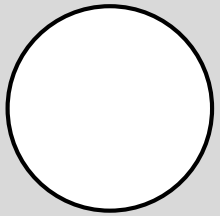
Petri Net Basics

What is a Petri Net?

- A Model that uses Places, Tokens, Arcs and Transitions to **model processes**
- Presence of tokens denotes availability of resources (**the state** of the petri net)
- Transitions **Fire** and the state changes – different transitions can then fire

Petri Net Components

- Places
  - **Resources**/Conditions
  - Can hold Tokens
  - Infinite capacity (unless labelled)



- Tokens
  - **Availability** of a Resource
  - Condition is satisfied

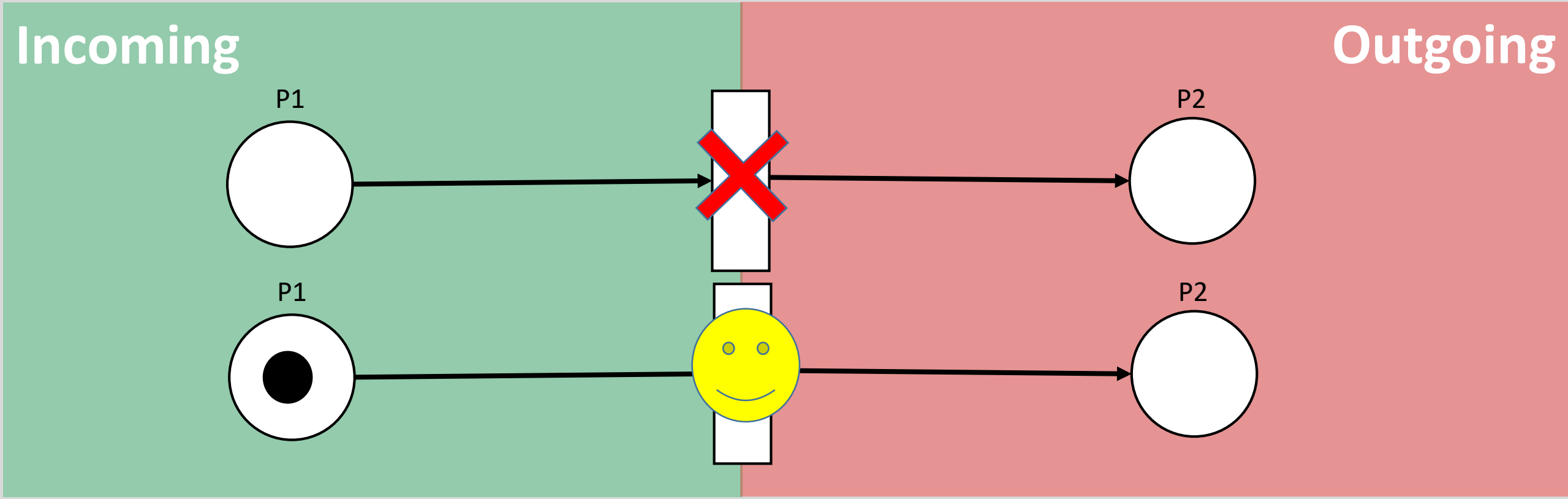
- Transitions
  - **Processes**
  - Consumes Tokens
  - Generates NEW Tokens

- Arcs
  - Shows **dependencies**
  - Connects Places with Transitions
  - Tokens travel along arcs
  - Unlabelled arc has capacity 1
  - Labelled arcs have the labelled capacity

Is The Transition Enabled?

A transition must be **enabled** for it to fire

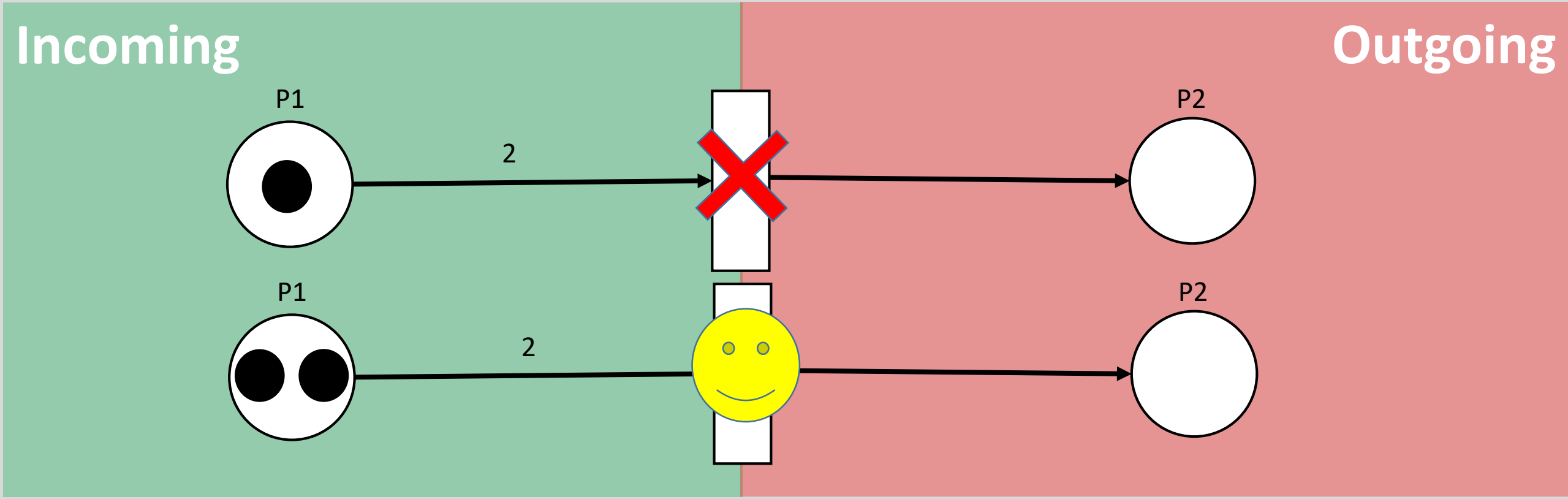
Each incoming place must have tokens **equal to or greater than** arc capacity



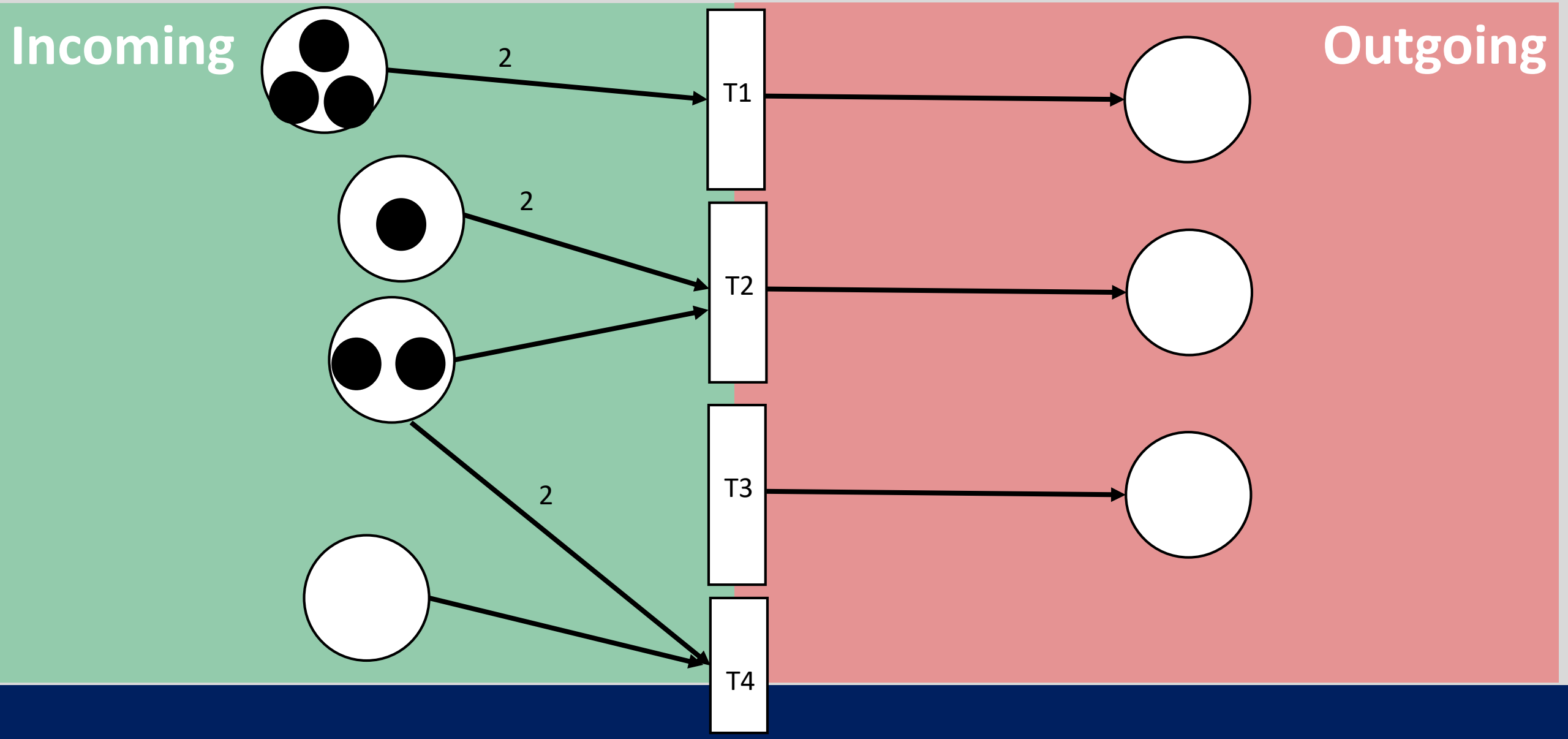
Is The Transition Enabled?

A transition must be **enabled** for it to fire

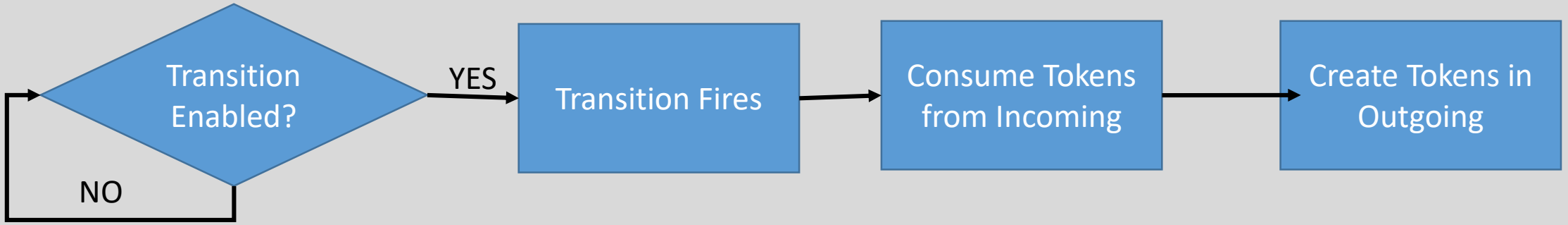
Each incoming place must have tokens **equal to or greater than** arc capacity

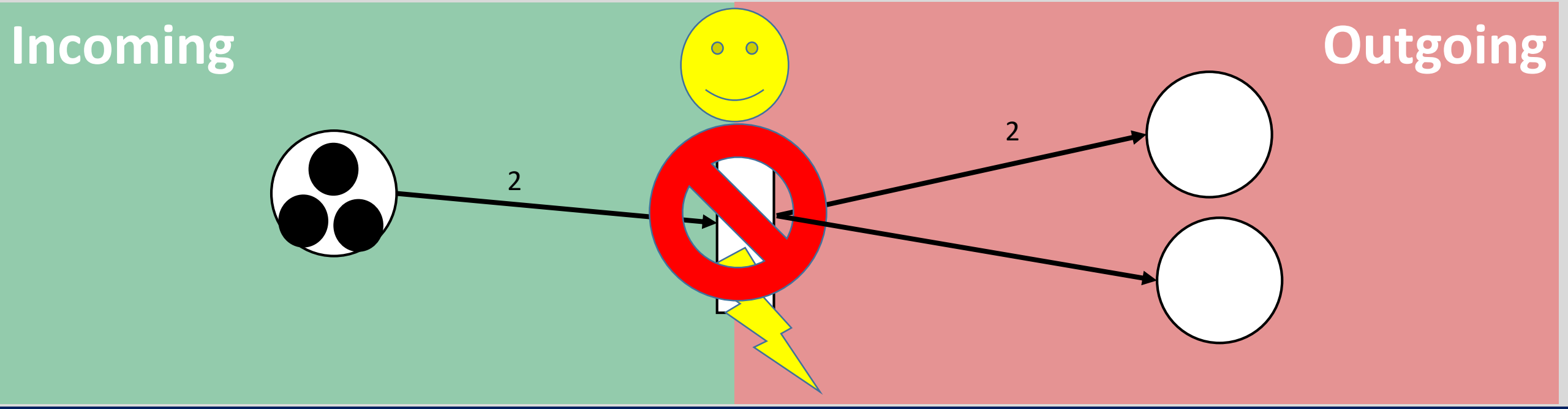


Exercise: Are the transitions enabled?

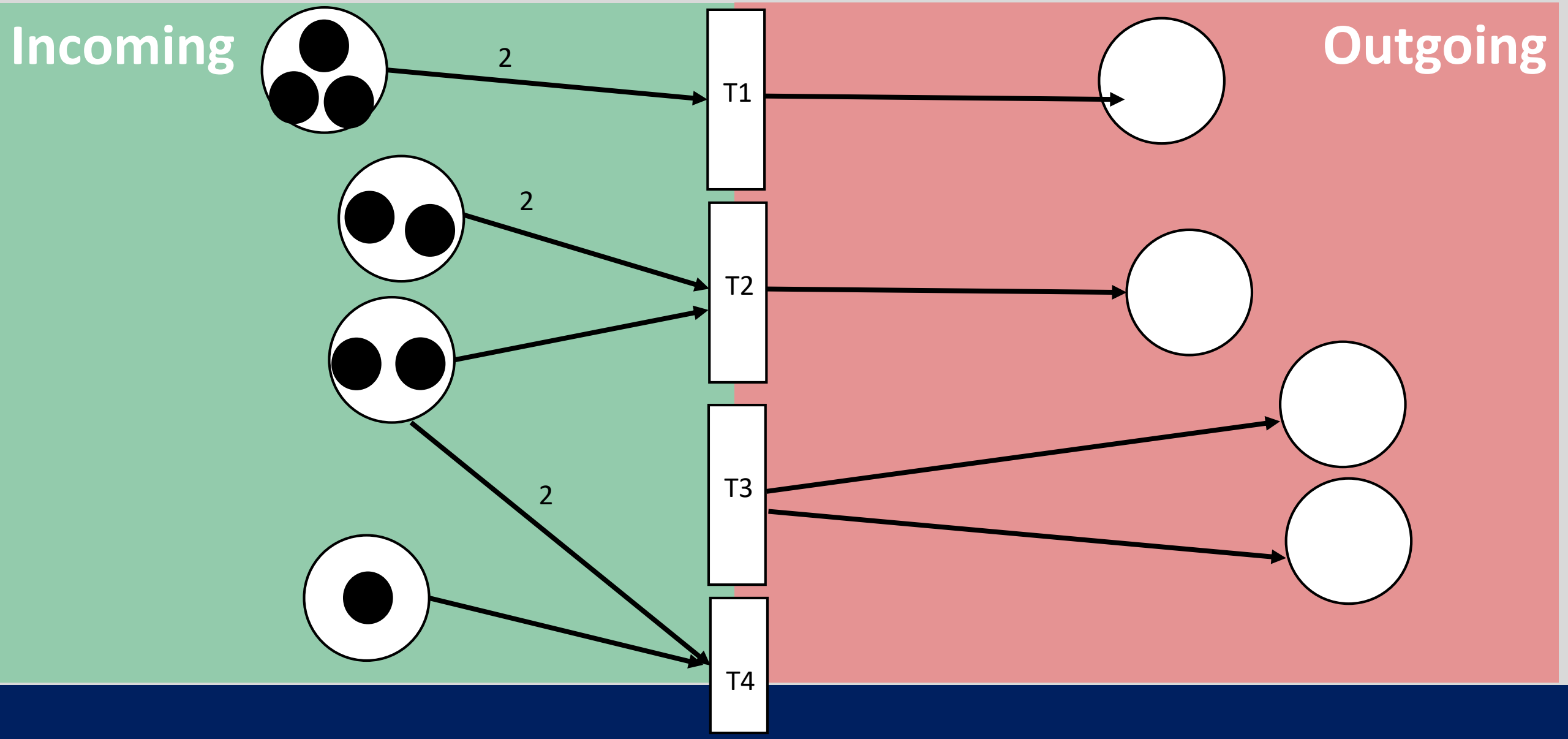


Transitions Firing

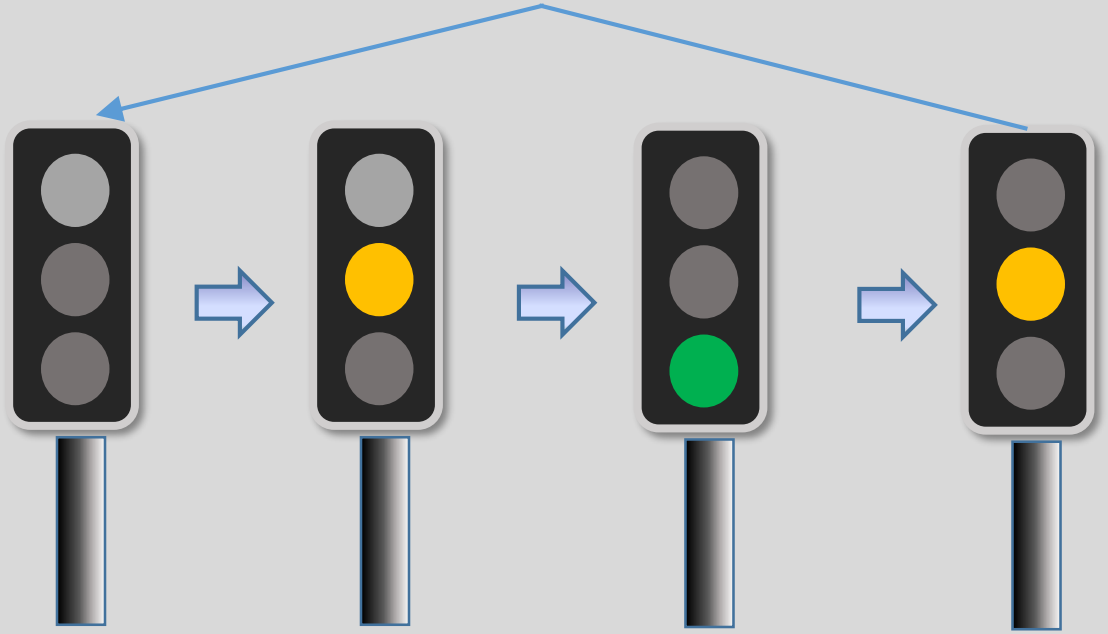




Exercise: Effect of Firing



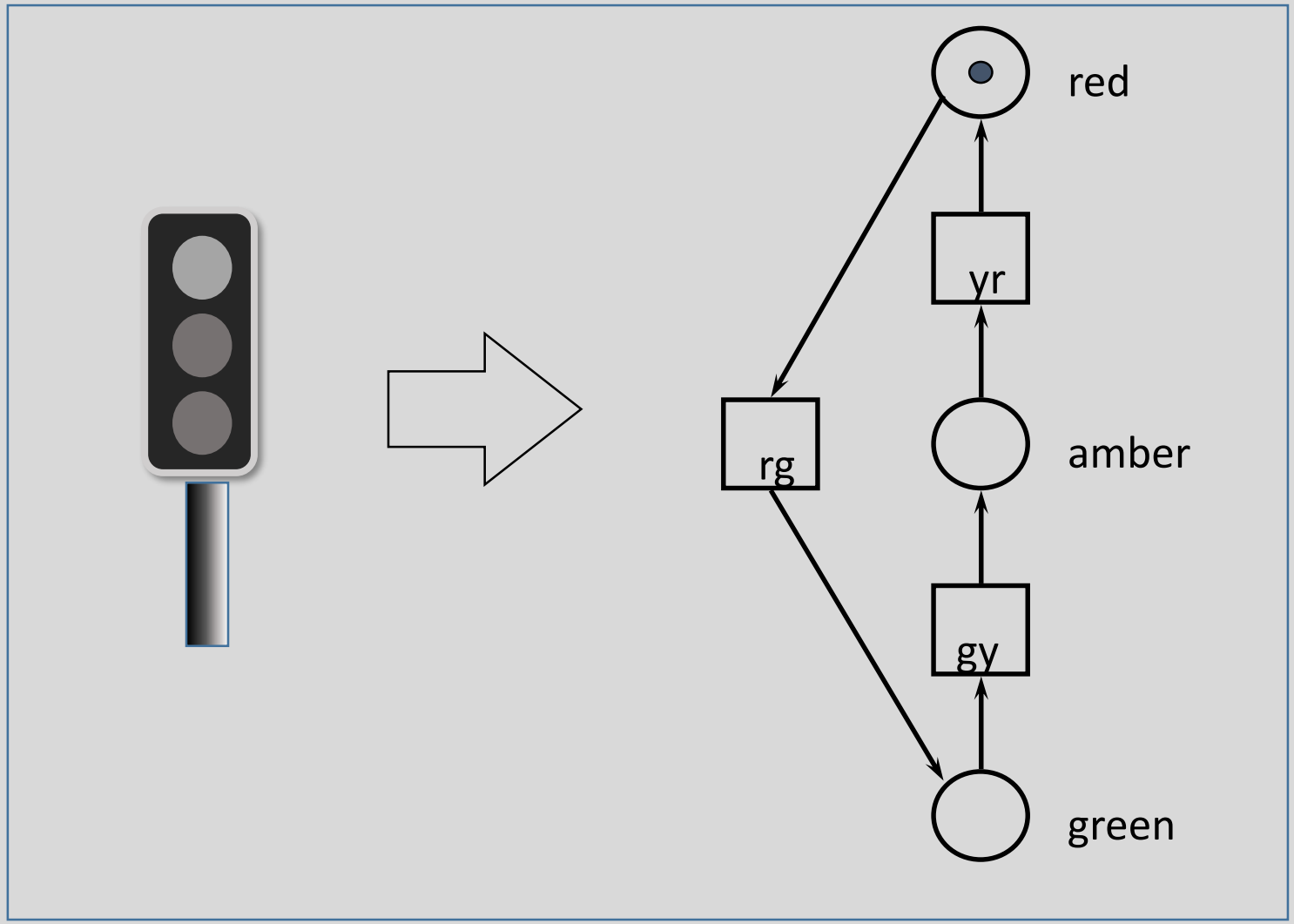
Modelling a Traffic Light



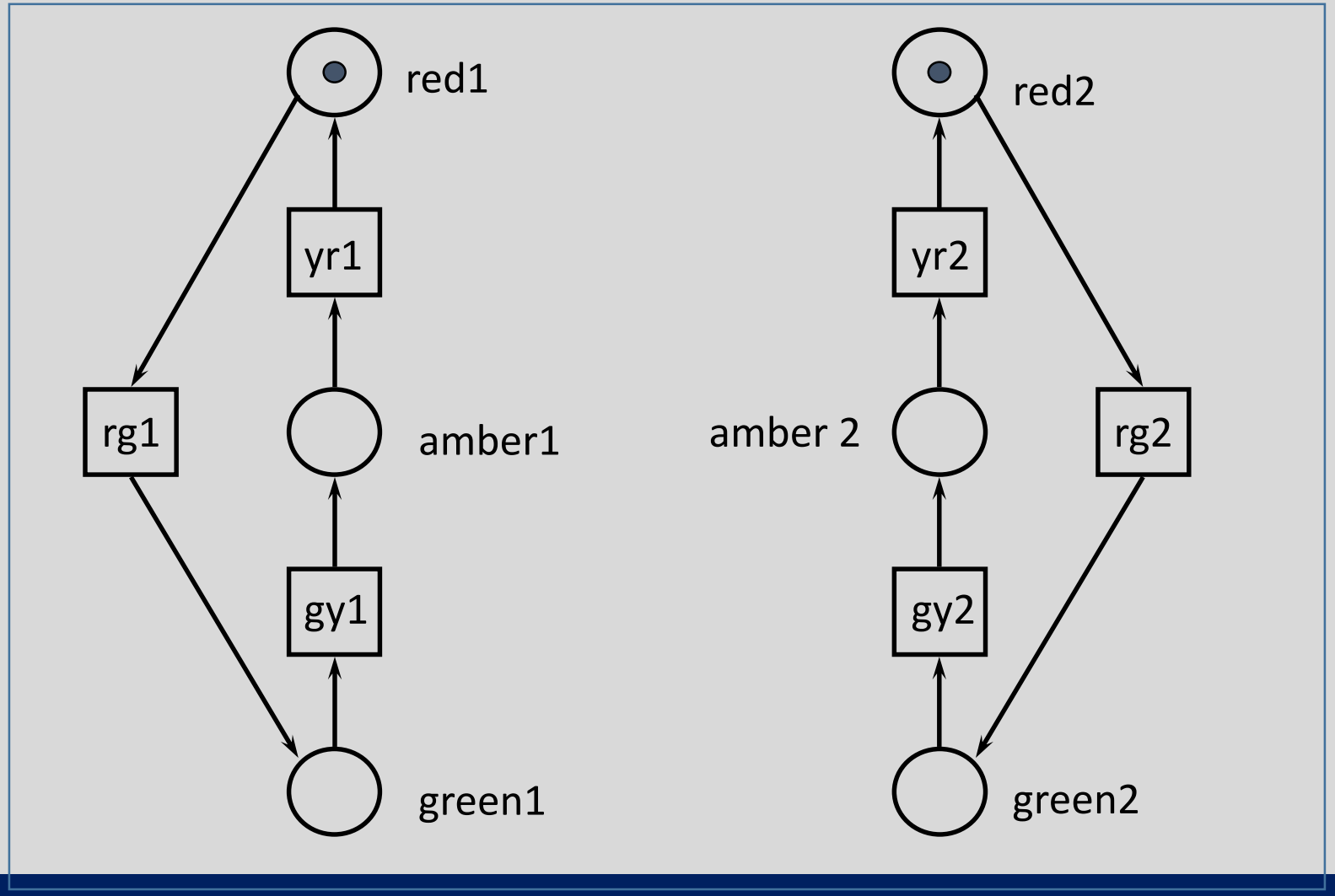
Modelling Two Traffic Lights

- Imagine that we are designing a traffic light system for a crossroads junction (i.e. with two sets of (simplified) lights).
- An informal specification of a traffic light junction:
  - A single traffic light turns from “Red” to “Green” to “Amber” and then back to “Red” (we’ll ignore “red and amber” for now).
  - There are two sets of lights. When one of the traffic lights is “Amber” or “Green”, the other must be “Red”.
- As a first step, we may decide to model the system as a Petri net. This allows us to make sure the specification is rigorously defined and reduces potential ambiguities later.
- We can also prove properties about the model if we wish.

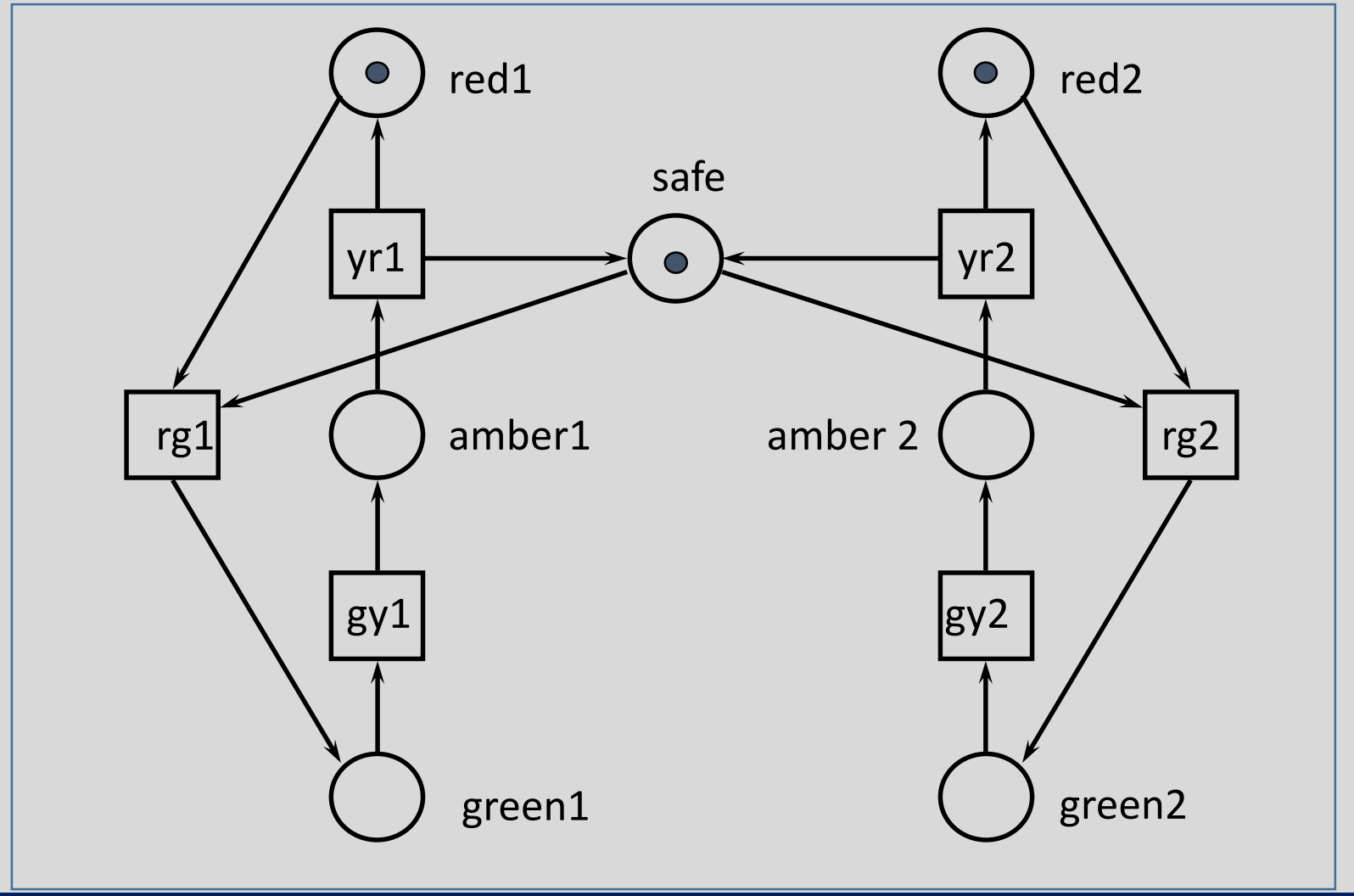
Example: Traffic Light



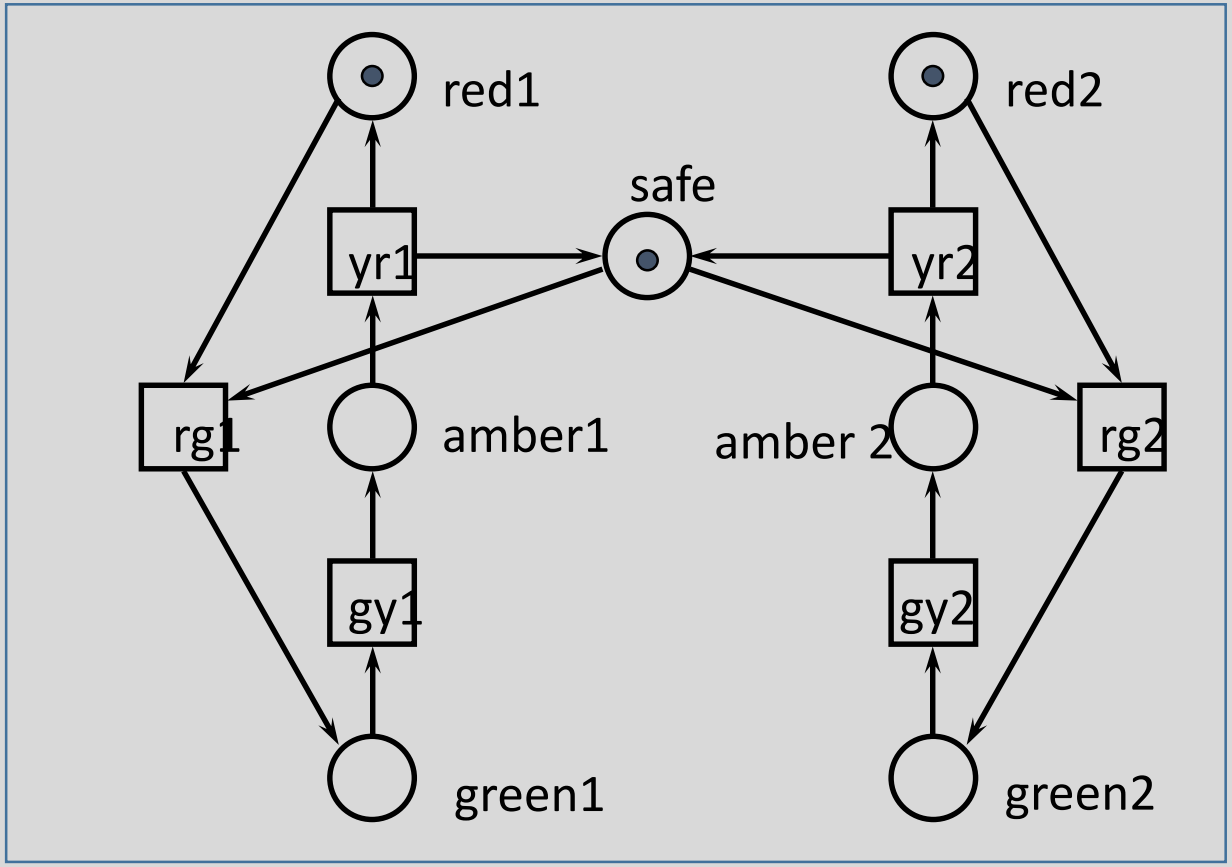
Two Traffic Lights

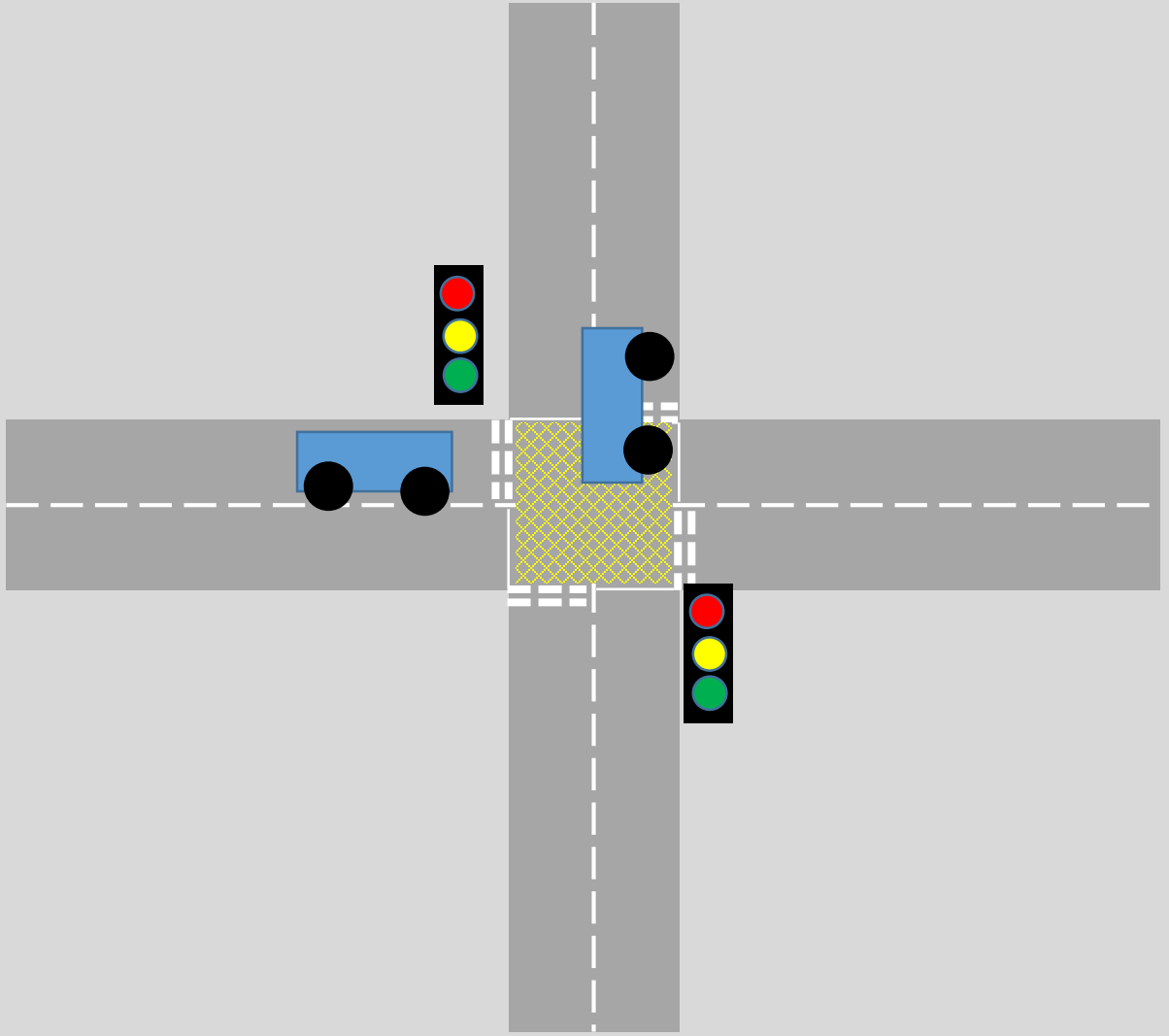


Two Safe Traffic Lights



Can you use the Petri Net to guarantee that the cars wont crash?





Lecture Key Points

- Petri nets have Arcs, Places and Transitions.
- Petri nets are **non-deterministic** and thus may be used to model discrete distributed systems.
- They have a well defined semantics and many variations and extensions of Petri nets exist.
- The state or marking of a net is an assignment of tokens to places.
- For those interested, the book “Fundamentals of Software Engineering” (Prentice Hall) by C. Ghezzi, M. Jazayeri and D. Mandrioli has an extensive example of using Petri nets for an elevator system.
- **Next lecture: More modelling Traffic Light and Extensions of Petri Nets with Time and Colour**