Möbius Tool

LAB 02

Contacts

Maurizio Palmieri

PhD student of the Department of Information Engineering, University of Pisa

Office: Largo Lucio Lazzarino 1 - 56122 Pisa (PI), Italy

Email: maurizio.palmieri@ing.unipi.it

Overview

- Introduction to SAN in Möbius
- Example of Producer-Consumer
- Exercises

Stochastic Activity Networks (SAN)

The *Stochastic Activity Networks* are a wide-ranging and complex extension to Petri-Nets

Petri Net = places + transitions + enabling conditions + firing rules

Stochastic Petri Net = Petri Net + stochastic transition delay

Stochastic Activity Network = Stochastic Petri Net + stochastic transition outcome + advanced enabling condition + advanced firing rules

SAN in Möbius tool 1/2

NOTE: *the terms **activity**, **transition** and **action** will be used interchangeably*

- Transitions may be **timed** or **instantaneous**

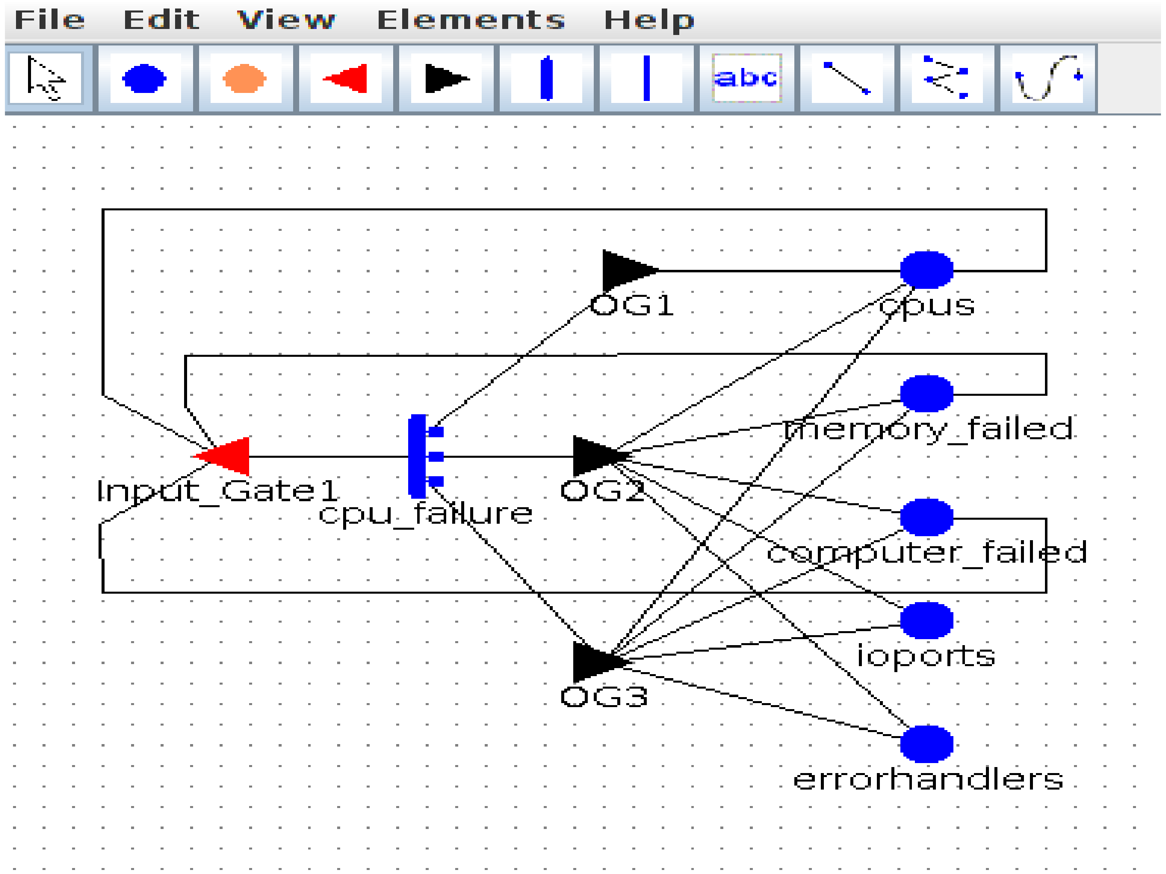
- **Enabling conditions** are defined with **input gates** associated with transitions

- **Firing rules:** user-defined functions specified in **input** or **output gates**

SAN in Möbius tool 2/2

Stochastic transition outcome: Alternative results of a transition can be specified as mutually exclusive **cases** associated with the transition

Each case has a probability defined by a function of the **marking** (it may be a constant)

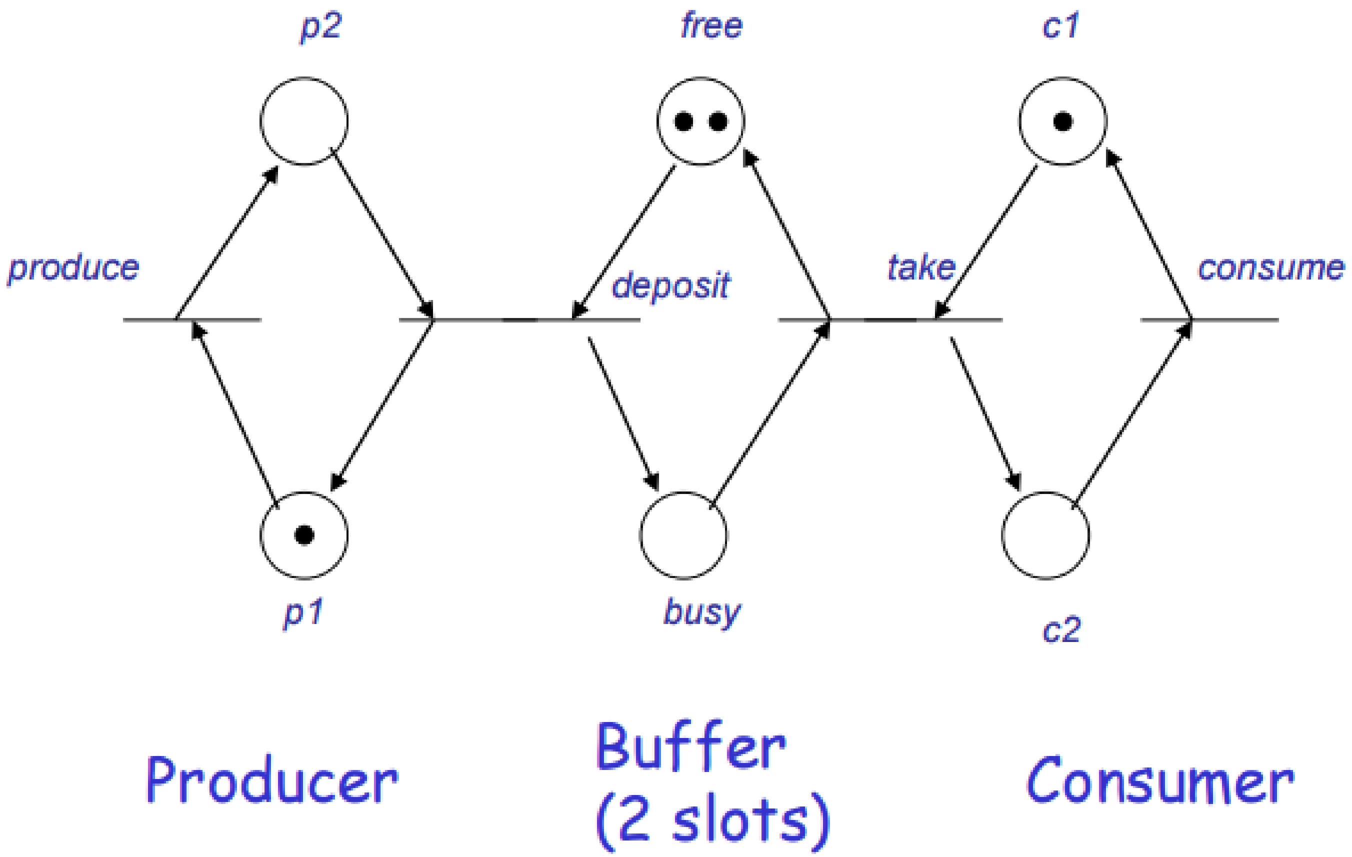


Elements of Möbius tool

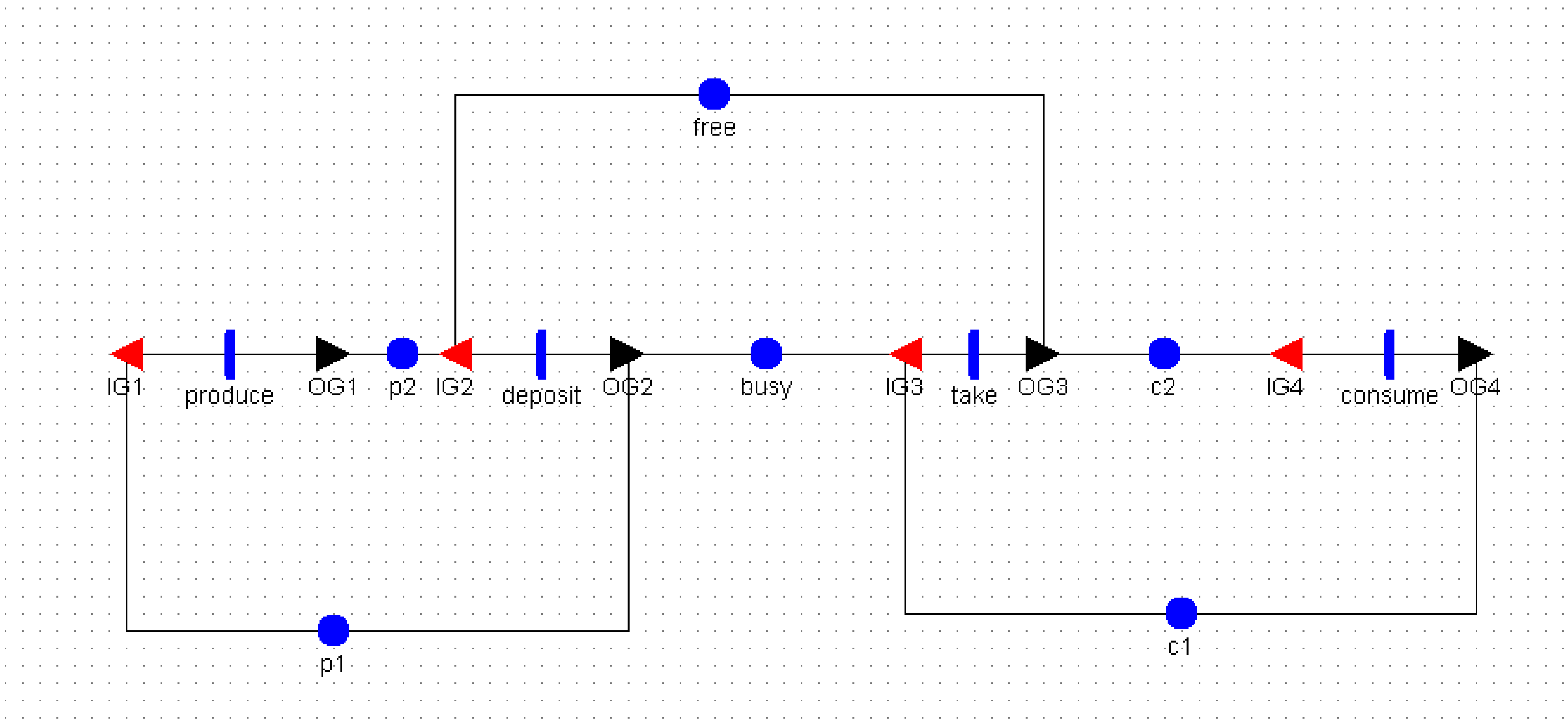
- **Atomic and Composed model, reward, study, transformer** and **solver**
  - are used in the same way they are used for Fault Tree analysis.

- Möbius allows to evaluate a Performance Variable
  - **Steady State**
  - **Transient (instant of time)**

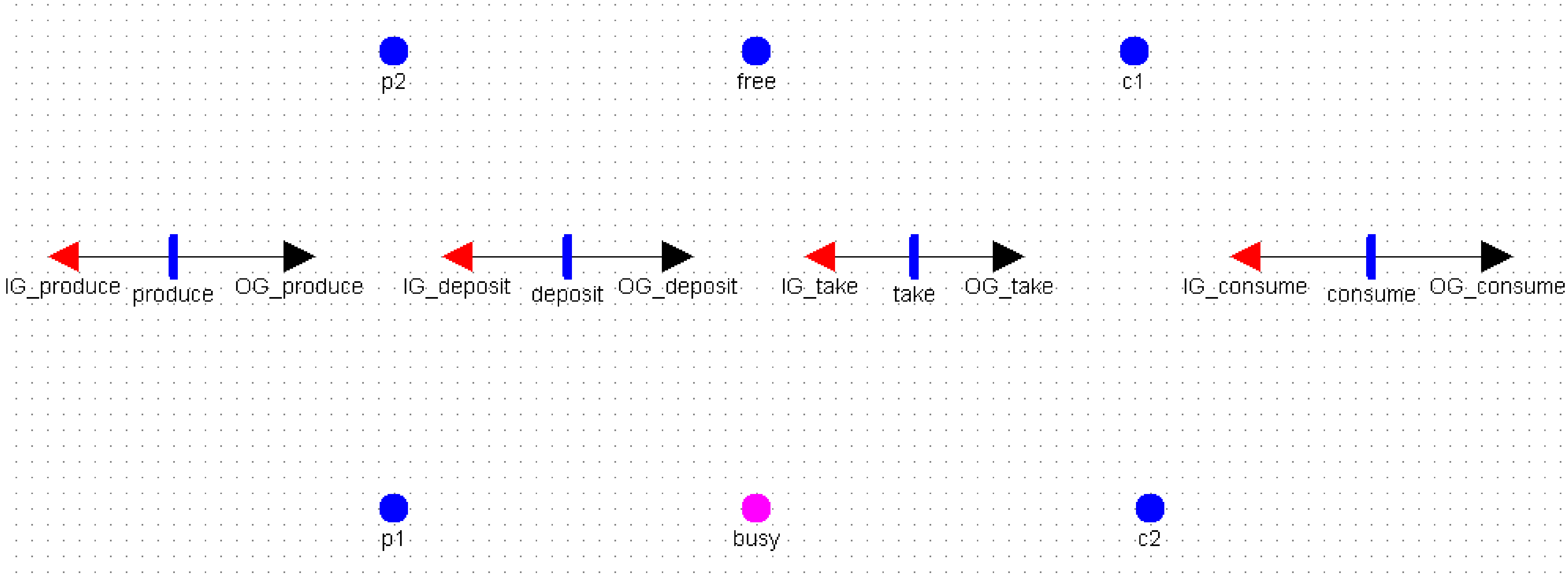
Example of Producer/Consumer



Atomic Model 1/2



Atomic model 2/2



Reward of Producer/Consumer

Evaluate the probabilty of having the buffer completely full during steady state phase

Reward function:

```
if ( P_C->busy->Mark() == 2) return 1;
else return 0;
```

Time:

Steady State with default values for parameters

Study for Producer/Consumer

- Create a base experiment with all rates with same value (for example 0.1)

- Then create a second experiment where the value of the rate of activities **before** place **busy** are higher than the others.
  - for example set them to 0.9 without changing the other.

- There are only 2 experiments so use a **Set Study.**

Transformer and Solver

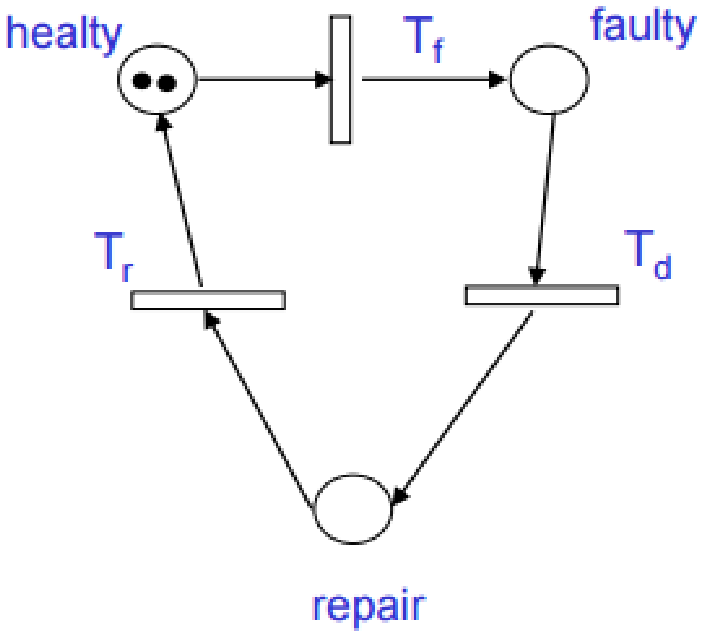
- Again use the State Space Generator (**NOT** Symbolic) as transformer

- Then, in order to evaluate the steady state behavior choose a Steady State solver.
  - For simplicity select the Direct Steady State Solver.

- The behavior of different solvers can be found in the Möbius wiki

Exercise 1

- Two identical CPUs
- **Failure of the CPU:** exponentially distributed with parameter λ
- **Fault detection:** exponentially distributed with parameter δ
- **CPU repair:** exponentially distributed with parameter μ



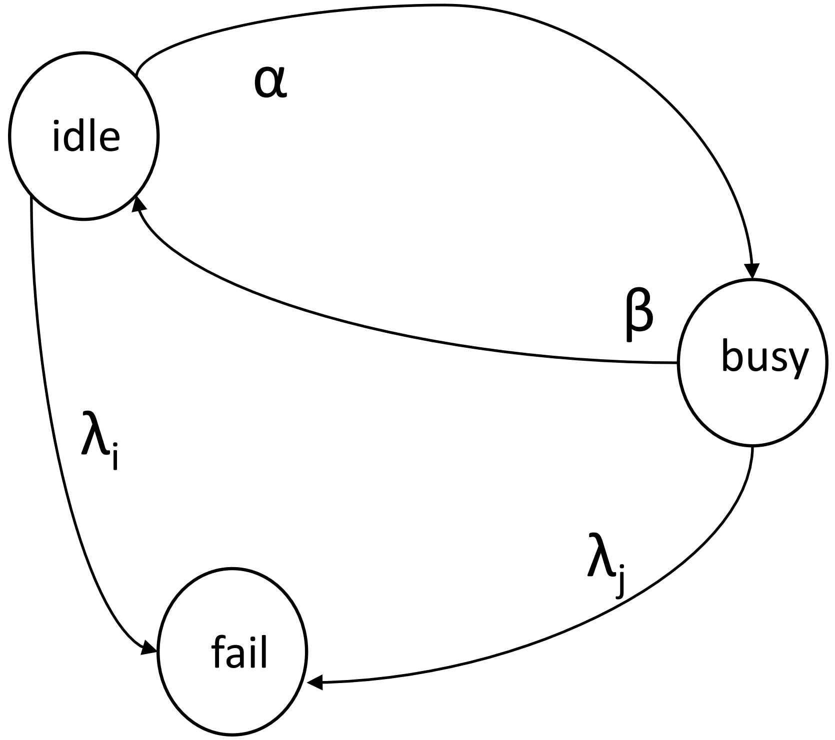
Evaluate Availability of the system during steady state phase

Exercise 2

- A computer is *idle*, *busy*, or *failed*;
- jobs arrive at a rate α ;
  - 1000
- jobs are completed at a rate β ;
  - 10000
- the computer fails at rate λ_i when idle;
  - 1.0E-7
- the computer fails at rate λ_j when busy.
  - From 1×10^{-6} to 5×10^{-6}

Evaluate reliability after 24

time units of operation



References

William H. Sanders and John F. Meyer, ``Stochastic Activity Networks:formal definitions and concepts'', in Lectures on formal methods andperformance analysis: first EEF/Euro summer school on trends in computer science, 2002.

https://www.mobius.illinois.edu/wiki/index.php/Möbius_Documentation

Thanks to prof. Andrea Domenici for previous version of the slides.