

Exercise 6

Generalized Stochastic Petri Nets (GSPNS)

Introduction to Laboratory Class 2

Fault-Tolerant Computer Systems

2012

Elements of GSPNs

- ▶ *Places* – Hold tokens.



Place



Place marked with 2 tokens

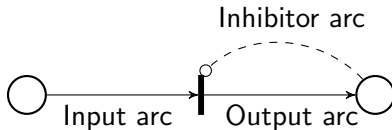
- ▶ *Timed transitions* – Fires at a random point in time when enabled.



- ▶ *Immediate transitions* – Fires immediate when enabled.



- ▶ *Input and output arcs* – Connect places with transitions
- ▶ *Inhibitor arcs* – Blocks the firing of a transition.

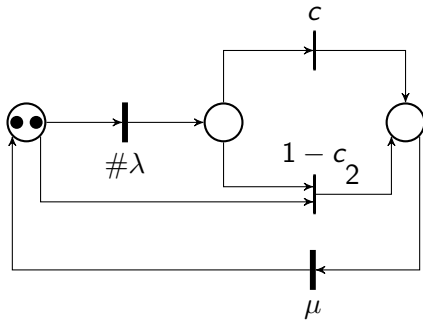


Hot stand-by system

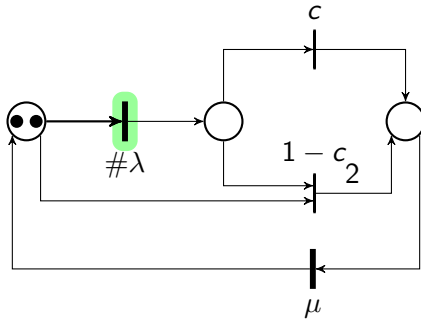
A fault-tolerant computer system consists of two modules working in active redundancy. The failure rate of the computer modules is λ and the repair rate is μ . The coverage for reconfigurations is c .

Define a Petri net model of the system.

Hot stand-by system

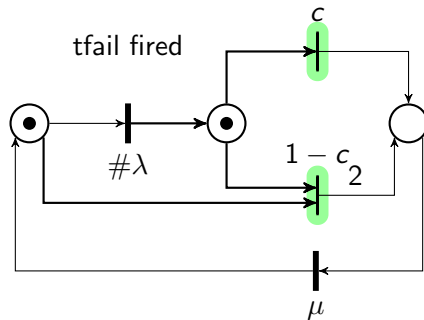


Hot stand-by system



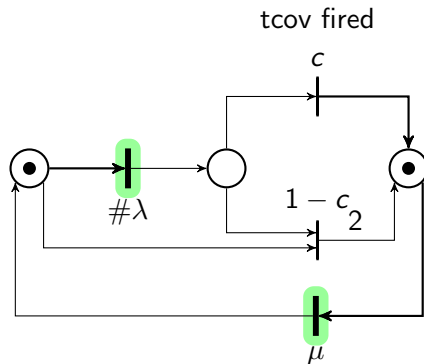
	Marking	Comment
$\Rightarrow$	$\{2, 0, 0\}$	Initial marking

Hot stand-by system



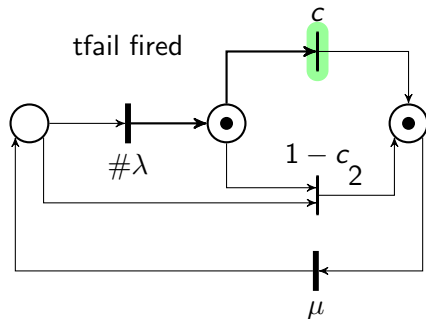
	Marking	Comment
	$\{2, 0, 0\}$	Initial marking
$\Rightarrow$	$\{1, 1, 0\}$	One module failed (vanishing marking)

Hot stand-by system



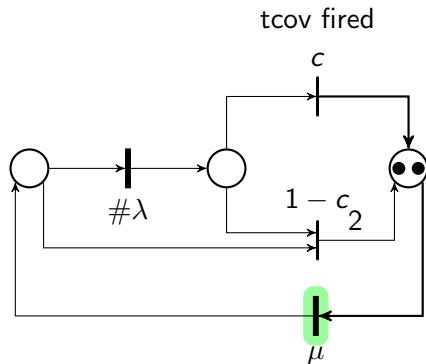
	Marking	Comment
	$\{2, 0, 0\}$	Initial marking
	$\{1, 1, 0\}$	One module failed (vanishing marking)
$\Rightarrow$	$\{1, 0, 1\}$	One module failed

Hot stand-by system



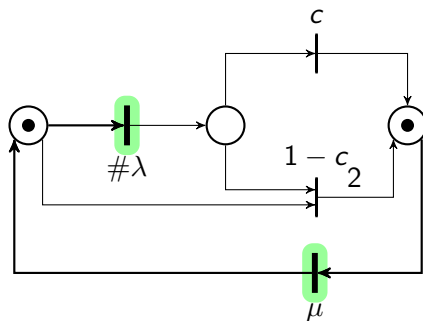
Marking	Comment
$\{2, 0, 0\}$	Initial marking
$\{1, 1, 0\}$	One module failed (vanishing marking)
$\{1, 0, 1\}$	One module failed
$\Rightarrow \{0, 1, 1\}$	Two modules failed (vanishing marking)

Hot stand-by system



Marking	Comment
$\{2, 0, 0\}$	Initial marking
$\{1, 1, 0\}$	One module failed (vanishing marking)
$\{1, 0, 1\}$	One module failed
$\{0, 1, 1\}$	Two modules failed (vanishing marking)
$\Rightarrow \{0, 0, 2\}$	Two modules failed

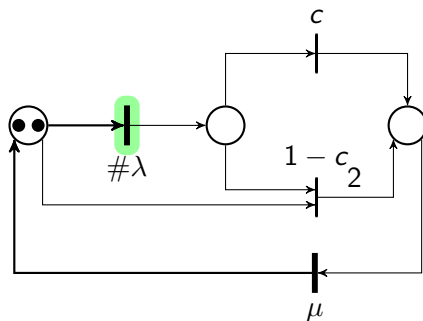
Hot stand-by system



repair fired

Marking	Comment
$\{2, 0, 0\}$	Initial marking
$\{1, 1, 0\}$	One module failed (vanishing marking)
$\Rightarrow \{1, 0, 1\}$	One module failed
$\{0, 1, 1\}$	Two modules failed (vanishing marking)
$\{0, 0, 2\}$	Two modules failed

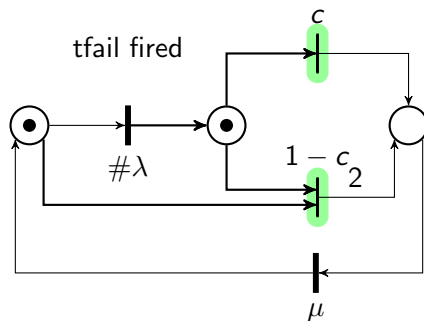
Hot stand-by system



repair fired

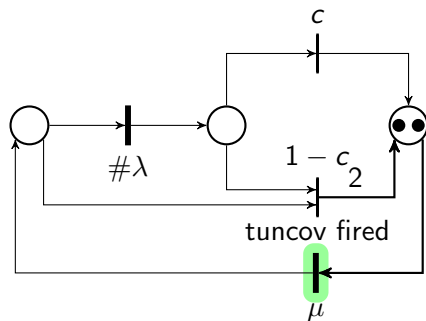
	Marking	Comment
$\Rightarrow$	$\{2, 0, 0\}$	Initial marking
	$\{1, 1, 0\}$	One module failed (vanishing marking)
	$\{1, 0, 1\}$	One module failed
	$\{0, 1, 1\}$	Two modules failed (vanishing marking)
	$\{0, 0, 2\}$	Two modules failed

Hot stand-by system



	Marking	Comment
	$\{2, 0, 0\}$	Initial marking
$\Rightarrow$	$\{1, 1, 0\}$	One module failed (vanishing marking)
	$\{1, 0, 1\}$	One module failed
	$\{0, 1, 1\}$	Two modules failed (vanishing marking)
	$\{0, 0, 2\}$	Two modules failed

Hot stand-by system



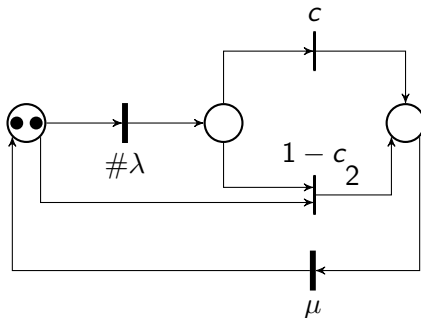
Marking	Comment
$\{2, 0, 0\}$	Initial marking
$\{1, 1, 0\}$	One module failed (vanishing marking)
$\{1, 0, 1\}$	One module failed
$\{0, 1, 1\}$	Two modules failed (vanishing marking)
$\Rightarrow \{0, 0, 2\}$	Two modules failed

GSPN Analysis

- ▶ Reachability graphs show possible firing sequences of GSPNs.
- ▶ An extended reachability graph contains
 - ▶ Tangible markings – Only timed transitions are enabled.
 - ▶ Vanishing markings – At least one immediate transition is enabled.
- ▶ A reduced reachability graph is created by transforming the extended reachability graph.
 - ▶ Transition rates are modified so that they account for the effects of tangible markings.

Hot stand-by system

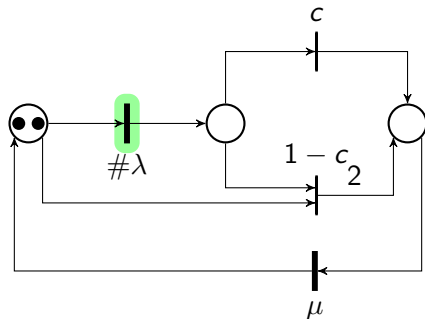
Extended reachability graph



Marking	Comment
$\{2, 0, 0\}$	Initial marking
$\{1, 1, 0\}$	One module failed (vanishing marking)
$\{1, 0, 1\}$	One module failed
$\{0, 1, 1\}$	Two modules failed (vanishing marking)
$\{0, 0, 2\}$	Two modules failed

Hot stand-by system

Extended reachability graph



2,0,0

1,1,0

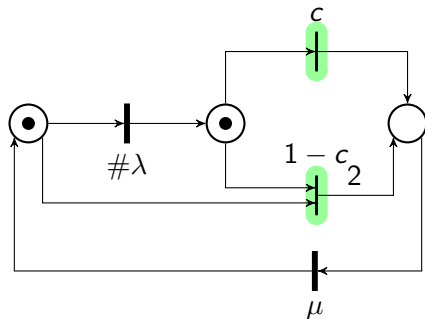
1,0,1

0,1,1

0,0,2

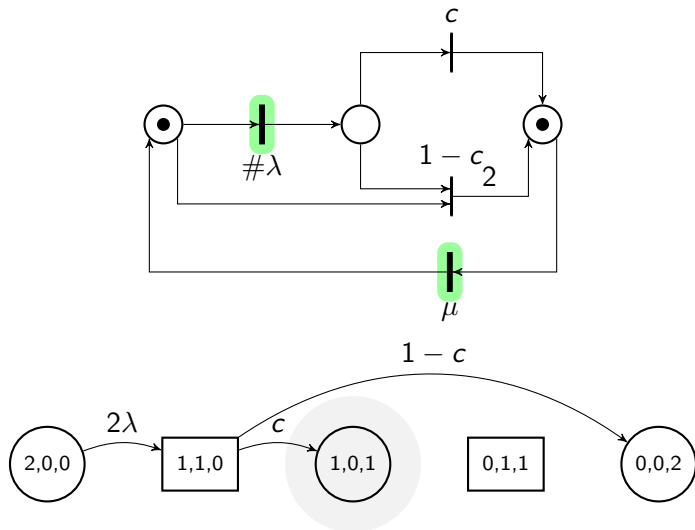
Hot stand-by system

Extended reachability graph



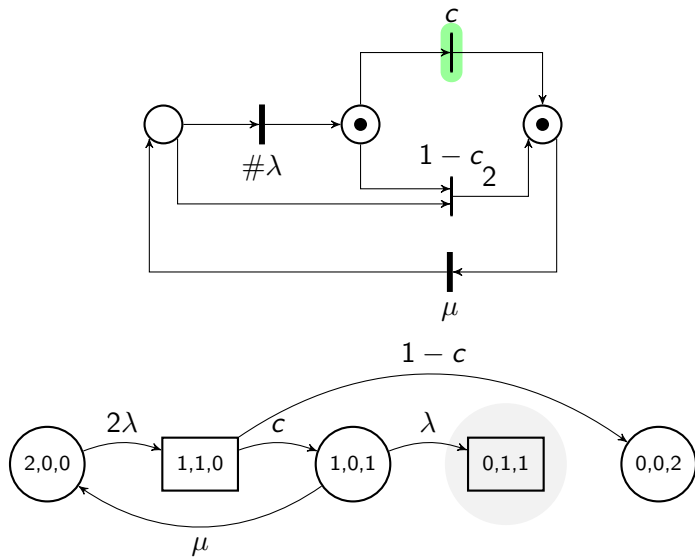
Hot stand-by system

Extended reachability graph



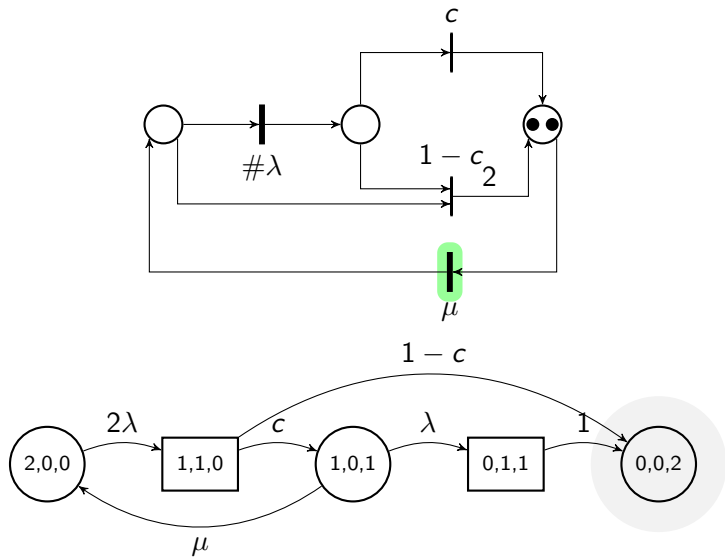
Hot stand-by system

Extended reachability graph



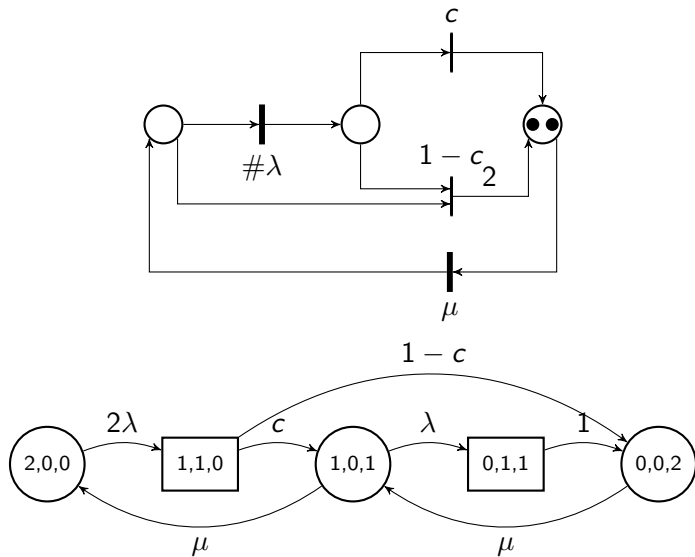
Hot stand-by system

Extended reachability graph



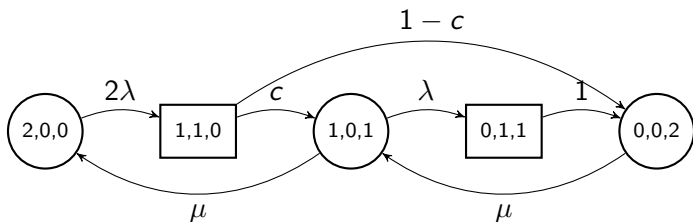
Hot stand-by system

Extended reachability graph

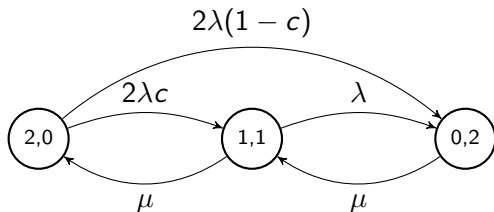


Hot stand-by system

Extended reachability graph:



Reduced reachability graph:



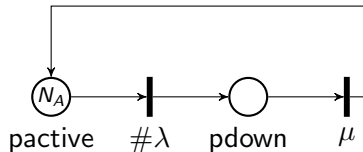
Cold stand-by system

A fault-tolerant computer system from start consists of N_A active modules and N_S spare modules. The failure rate for an active module is λ and the failure rate of the spare modules is negligible. The coverage for all reconfigurations is assumed to be ideal.

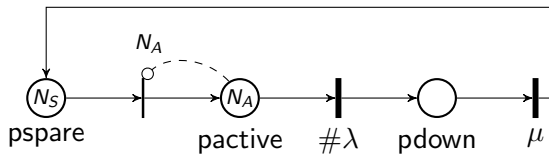
Define a GSPN model of the system.

Cold stand-by system

Hint: A hot stand-by system can be modelled with the following GSPN model:



Cold stand-by system



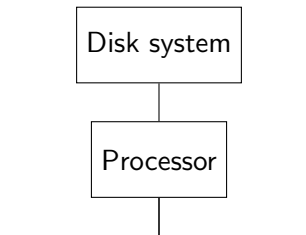
Laboratory class 2

Study the impact of repair on the availability.

- ▶ Evaluate two different file server configurations: a simplex and a duplex server.
- ▶ Use Markov models and GPSN models to calculate the steady-state availability.

Laboratory class 2

Simplex server

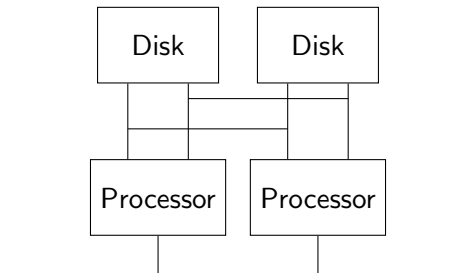


Repair policies:

1. Each unit has its own repair facility. The units fail independently of each other.
2. The units share one repair facility. The units fail independently of each other. Repairs are made using a *first-come first-served policy*. (Repairs are not preempted.)

Laboratory class 2

Duplex server



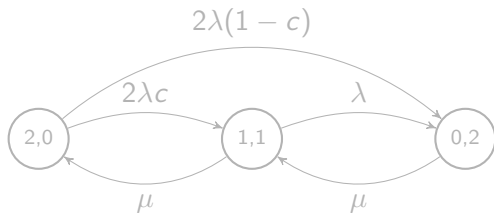
Laboratory class 2

Duplex server – Repair policies

1. One repair facility handles one processor or disk at a time.
 - ▶ Processor/disk failures do not occur if the system is down.
 - ▶ Processor repairs always are given priority over disk repairs.
2. One repair facility handles one processor or disk at a time.
 - ▶ Processor/disk failures may occur when the system is down
 - ▶ Processor repairs are given priority over disk repairs except when both disks have failed and one processor is working.
 - ▶ In that case, one disk is repaired first.
3. One repair facility for the processors and one for the disks.

SHARPE

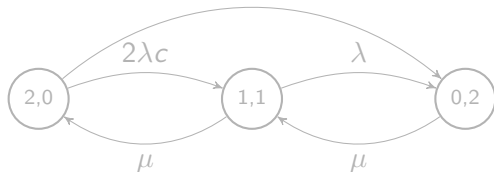
Markov chains



SHARPE

Markov chains

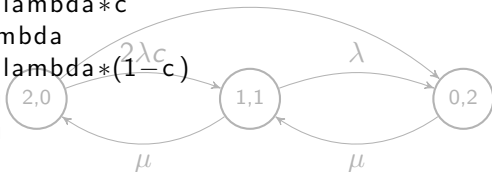
markov duplex_markov $2\lambda(1-c)$



SHARPE

Markov chains

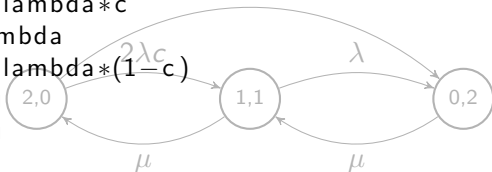
```
markov duplex_markov 2λ(1 - c)
s20 s11 2*lambda*c
s11 s02 lambda
s20 s02 2*lambda*(1-c)
s11 s20 mu
s02 s11 mu
end
```



SHARPE

Markov chains

```
markov duplex_markov 2λ(1 - c)
s20 s11 2*lambda*c
s11 s02 lambda
s20 s02 2*lambda*(1-c)
s11 s20 mu
s02 s11 mu
end
s20 1.0
end
```



SHARPE

Markov chains

```
1 markov name { (param_list) }
2 * Transitions and transition rates
3 <name name expression>
4 end
5 * Initial state probabilities
6 <name expression>
7 end
```

SHARPE

Steady-state probability for Markov models

- Calculate $\lim_{t \rightarrow \infty} A(t)$ for a state in a Markov model with

prob (system_name , state_eword {; arg_list })

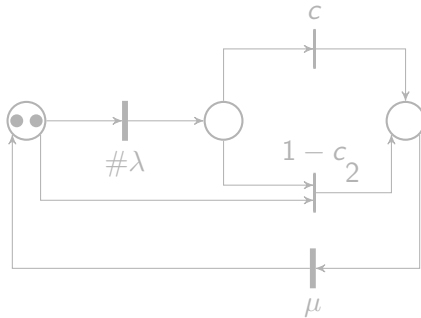
- Example: steady-state availability for the duplex system

```
func ss_avail_duplex_markov () \  
prob(duplex_markov , s20) + prob(duplex_markov , s11)
```

```
expr ss_avail_duplex_markov ()
```

SHARPE

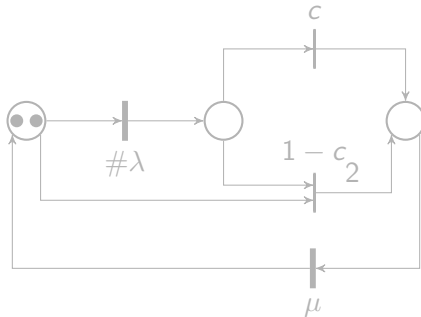
GSPNs



SHARPE

GSPNs

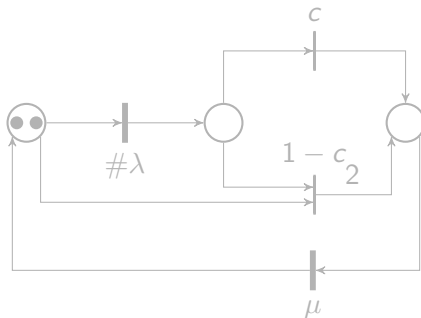
1 **gspn** duplex_petri



SHARPE

GSPNs

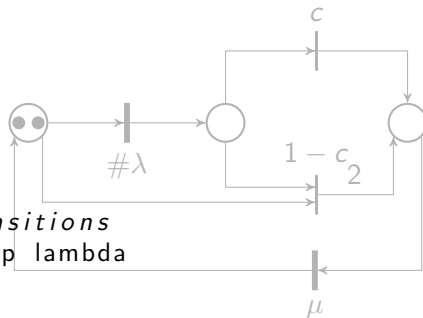
```
1 gspn duplex_petri
2
3 * places
4 pup 2
5 pdown 0
6 pdownim 0
7 end
```



SHARPE

GSPNs

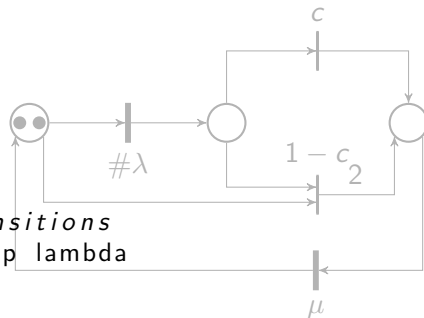
```
1 gspn duplex_petri
2
3 * places
4 pup 2
5 pdown 0
6 pdownim 0
7 end
8
9 * timed transitions
10 tfail dep pup lambda
11 trep ind mu
12 end
```



SHARPE

GSPNs

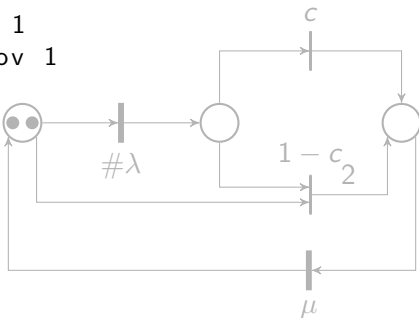
```
1 gspn duplex_petri
2
3 * places
4 pup 2
5 pdown 0
6 pdownim 0
7 end
8
9 * timed transitions
10 tfail dep pup lambda
11 trep ind mu
12 end
13
14 * immediate transitions
15 tcov ind c
16 tuncov ind (1-c)
17 end
```



SHARPE

GSPNs

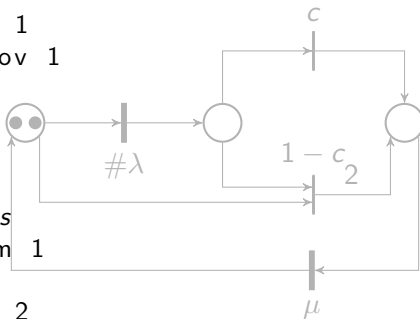
```
18 * input arcs
19 pup tfail 1
20 pdownim tcov 1
21 pdownim tuncov 1
22 pup tuncov 1
23 pdown trep 1
24 end
```



SHARPE

GSPNs

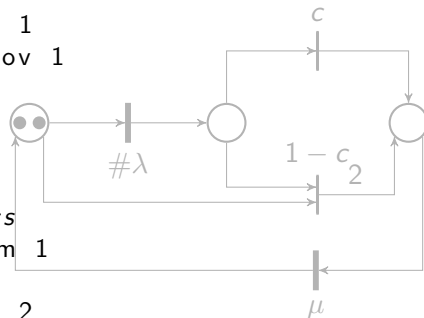
```
18 * input arcs
19 pup tfail 1
20 pdownim tcov 1
21 pdownim tuncov 1
22 pup tuncov 1
23 pdown trep 1
24 end
25
26 * output arcs
27 tfail pdownim 1
28 tcov pdown 1
29 tuncov pdown 2
30 trep pup 1
31 end
```



SHARPE

GSPNs

```
18 * input arcs
19 pup tfail 1
20 pdownim tcov 1
21 pdownim tuncov 1
22 pup tuncov 1
23 pdown trep 1
24 end
25
26 * output arcs
27 tfail pdownim 1
28 tcov pdown 1
29 tuncov pdown 2
30 trep pup 1
31 end
32
33 * inhibitor arcs
34 end
```



SHARPE

GSPNs

```
1  gspn name (param_list)
2  * section 1: places and initial numbers of tokens
3  <place_name expression>
4  end
5  * section 2: timed transition names, types and rates
6  <transition_name ind expression>
7  <transition_name dep place_name expression>
8  end
9  * section 3: immediate transition names and weights
10 <transition_name ind expression>
11 <transition_name dep place_name expression>
12 end
13 * section 4: place-to-transition arcs and multiplicity
14 <place name_transition_name expression>
15 end
16 * section 5: transition-to-place arcs and multiplicity
17 <transition_name place_name expression>
18 end
19 * section 6: inhibitor arcs and multiplicity
20 <place_name transition_name expression>
21 end
```

SHARPE

Steady-state probability for GSPNs

- Calculate the steady-state probability that a given state is empty using

```
preempty (system_name , place_eword {;arg_list })
```

- Example: steady-state availability for the duplex system

```
func ss_avail_duplex () \  
1 - preempty(duplex_petri , pup)
```

```
expr ss_avail_duplex ()
```

Laboratory class 2

Preparations for the lab

- ▶ Read the lab pm.
- ▶ Draw Markov models and GSPN models for the different repair policies.