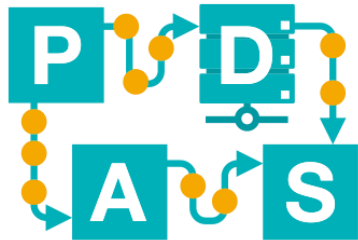


Petri Nets

Benedikt Knopp

BPI-Instruction 3



Chair of Process
and Data Science

RWTHAACHEN
UNIVERSITY

Agenda

Today:

- How does a Petri net work?
- How does a workflow work?
- How to reason about workflow properties?
- How to model systems and processes with Petri nets?
- How to use ProM for filtering event logs?

Next week:

- Alpha Miner
- Model Quality

Definition of Petri Nets

Exercise 1

A Petri net system (P, T, F, m_0) consists of a Petri net (P, T, F) and a distinguished marking m_0 , the *initial marking*.

Draw the following Petri net $N = (P, T, F)$ with:

$$P = \{p_1, p_2\}, T = \{t_1, t_2\}$$

$$F = \{(p_1, t_1), (t_1, p_2), (p_2, t_1), (p_2, t_2), (t_2, p_2), (t_2, p_1)\}$$

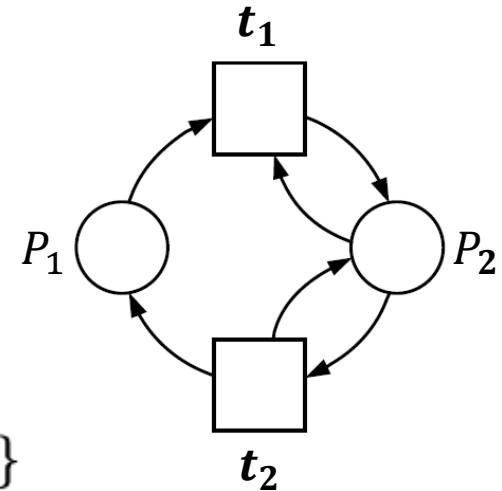
Definition of Petri Net

Exercise 1 Solution

Draw the following Petri net:

$$P = \{p_1, p_2\}, T = \{t_1, t_2\}$$

$$F = \{(p_1, t_1), (t_1, p_2), (p_2, t_1), (p_2, t_2), (t_2, p_2), (t_2, p_1)\}$$



Definition of Petri Net

Exercise 2

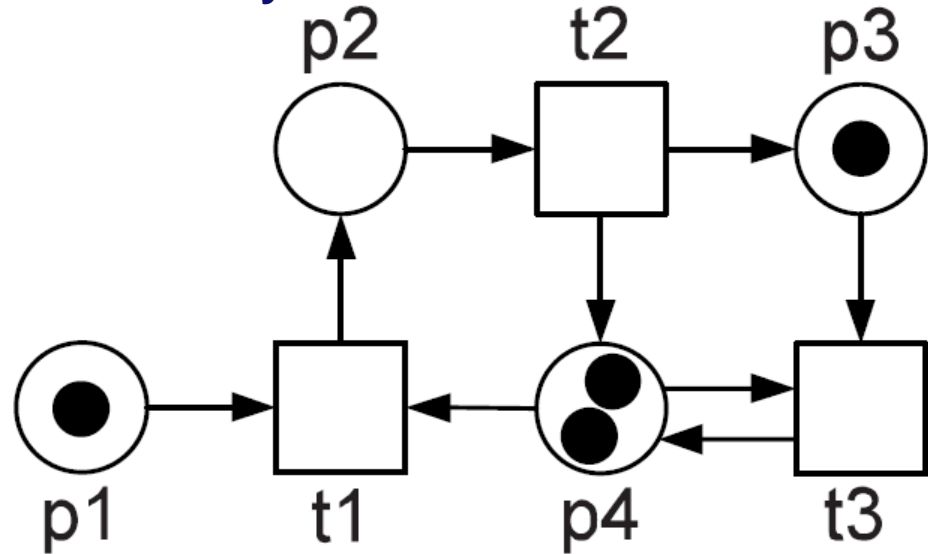
Define the following Petri net system formally:

$P = ?$

$T = ?$

$F = ?$

$m_0 = ?$



Definition of Petri Net

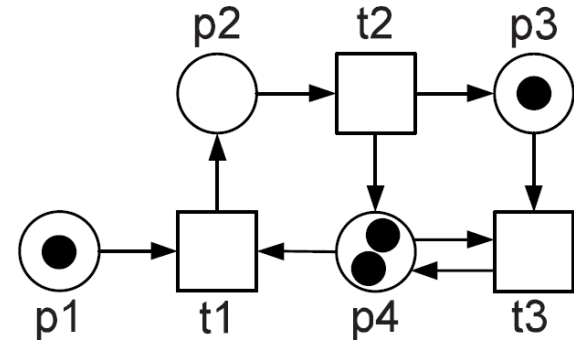
Exercise 2 Solution

$$P = \{p1, p2, p3, p4\}$$

$$T = \{t1, t2, t3\}$$

$$F = \{(p1, t1), (t1, p2), (p2, t2), (t2, p4), (t2, p3), \\ (p3, t3), (t3, p4), (p4, t3), (p4, t1)\}$$

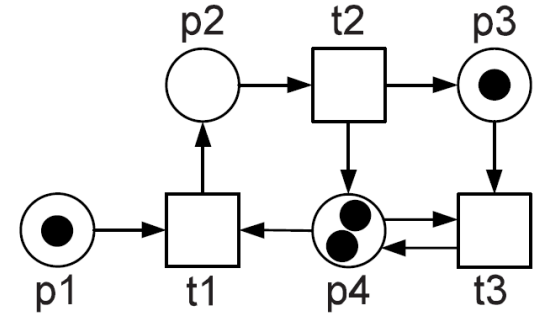
$$m_0 = [p1, p3, p4^2]$$



Definition of Petri Net

Exercise 3

1. Give the preset and postset of each transition.
2. Which transitions are all enabled at $m_0 = [p1, p3, p4^2]$?
3. Give all reachable markings.
4. What are the reachable final markings?



Definition of Petri Net

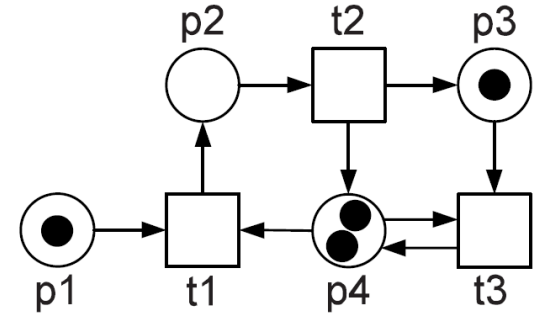
Exercise 3 Solution

1. Give the preset and postset of all transitions.

$\bullet t_1 = \{p1, p4\}$, $t_1^\bullet = \{p2\}$

$\bullet t_2 = \{p2\}$, $t_2^\bullet = \{p3, p4\}$

$\bullet t_3 = \{p3, p4\}$, $t_3^\bullet = \{p4\}$

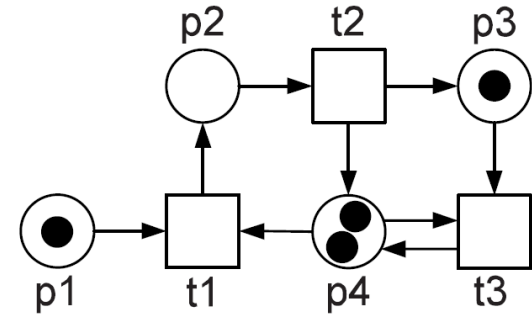


Definition of Petri Net

Exercise 3 Solution

2. Which transitions are enabled at $m_0 = [p1, p3, p4^2]$?

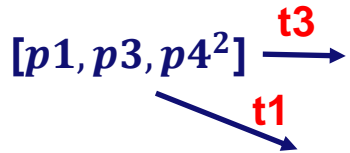
t_1 and t_3



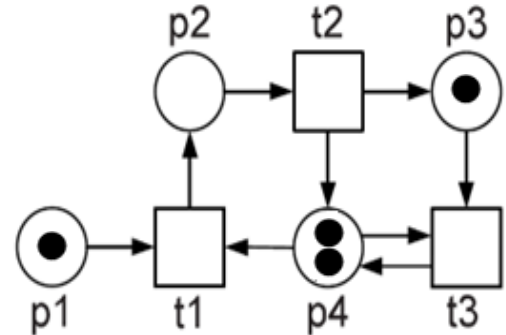
Definition of Petri Net

Exercise 3 Solution

3. Give all reachable markings.



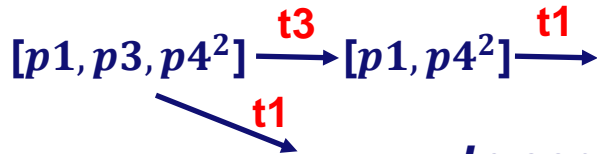
Start by writing down the initial marking and enabled transitions.



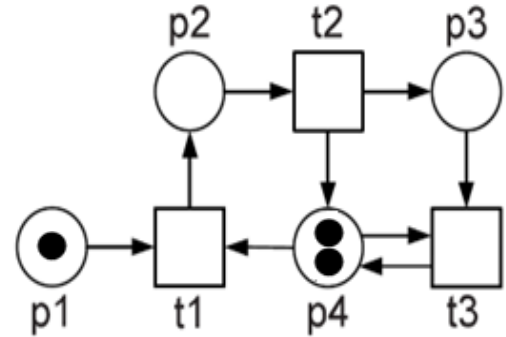
Definition of Petri Net

Exercise 3 Solution

3. Give all reachable markings



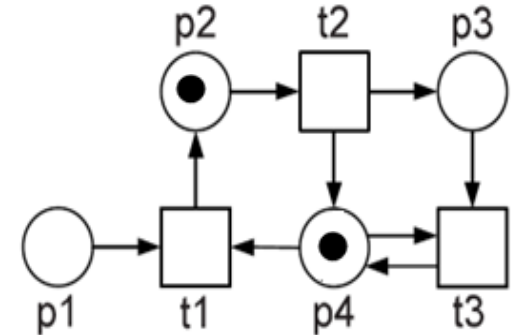
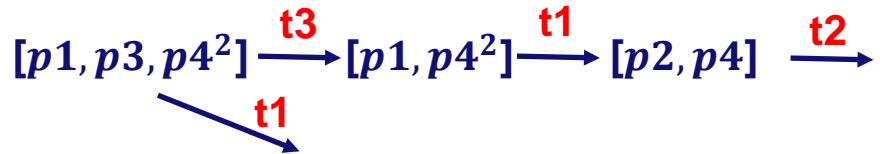
In each step, “play the token game” for one open transition branch. Draw the resulting marking and again all enabled transitions.



Definition of Petri Net

Exercise 3 Solution

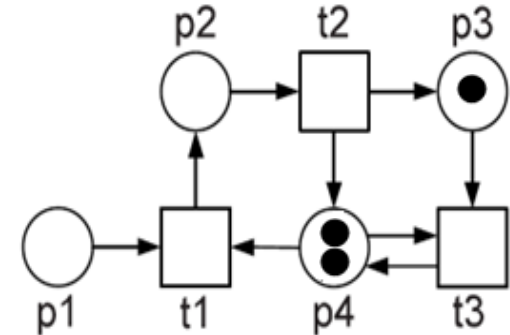
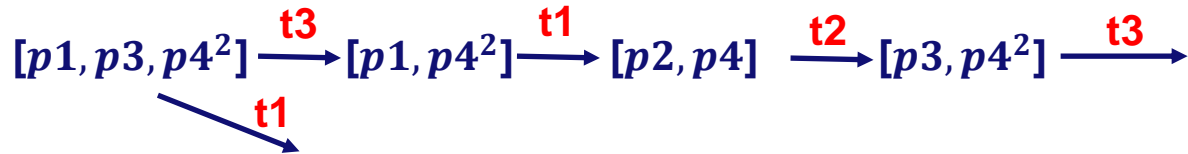
3. Give all reachable markings



Definition of Petri Net

Exercise 3 Solution

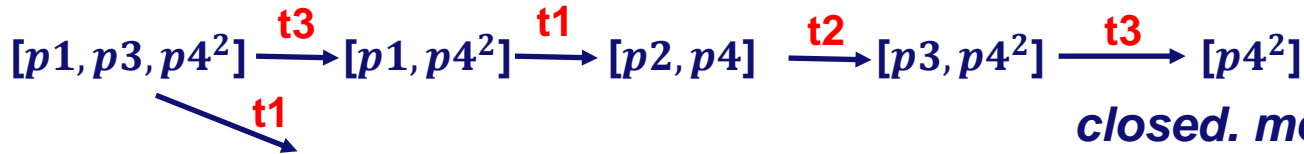
3. Give all reachable markings



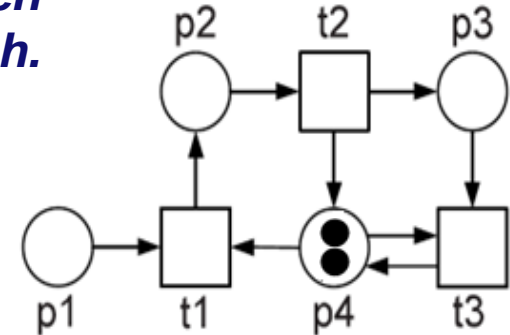
Definition of Petri Net

Exercise 3 Solution

3. Give all reachable markings



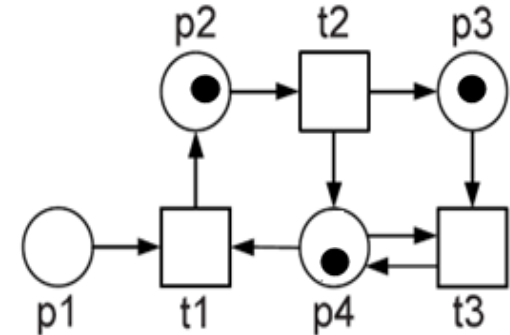
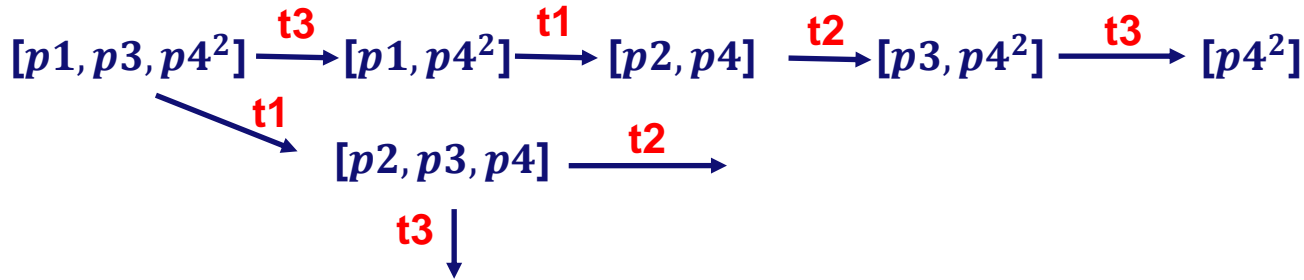
*closed. move
to open
branch.*



Definition of Petri Net

Exercise 3 Solution

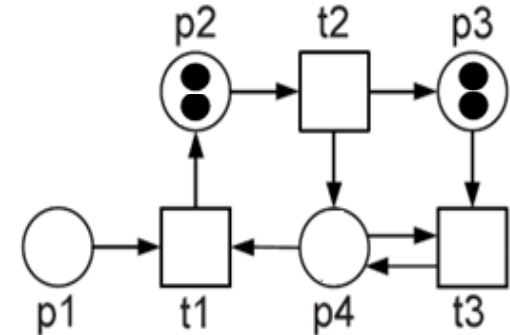
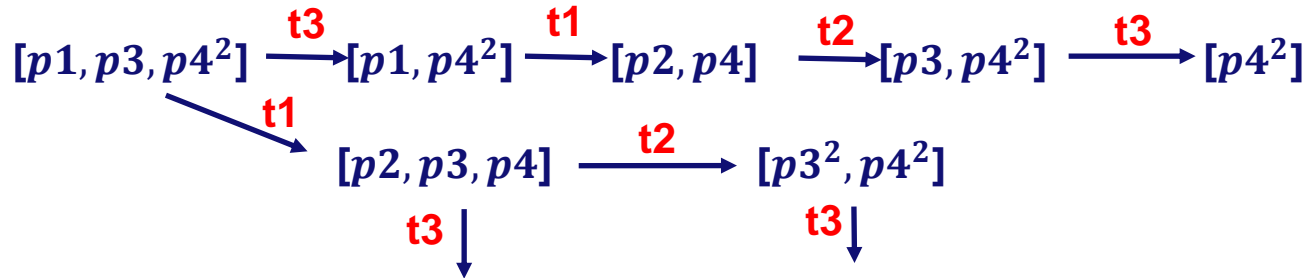
3. Give all reachable markings



Definition of Petri Net

Exercise 3 Solution

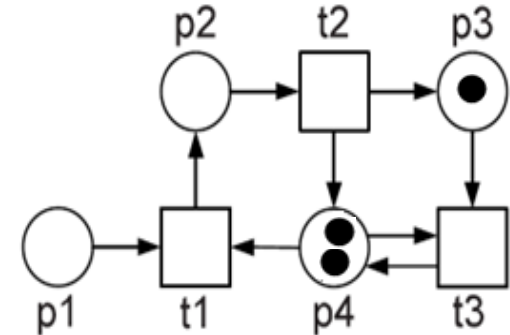
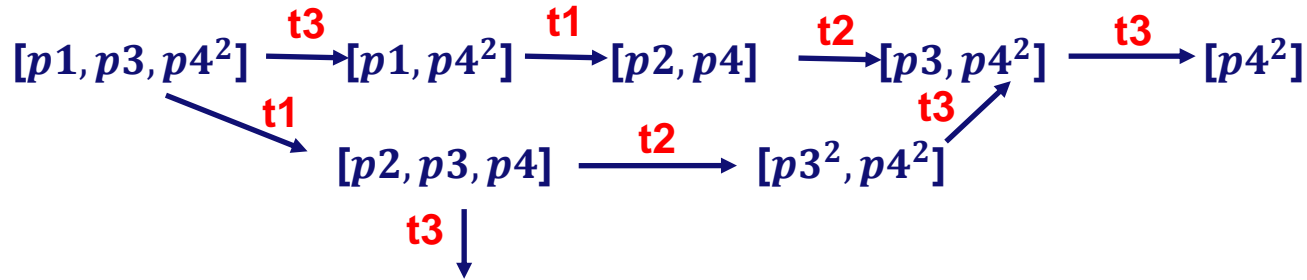
3. Give all reachable markings



Definition of Petri Net

Exercise 3 Solution

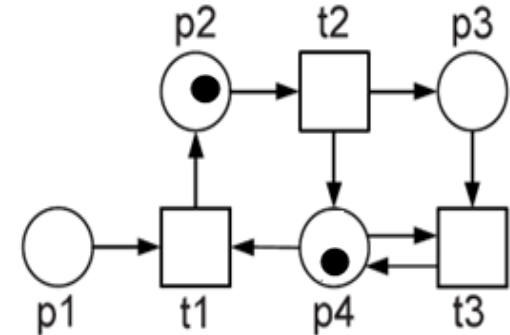
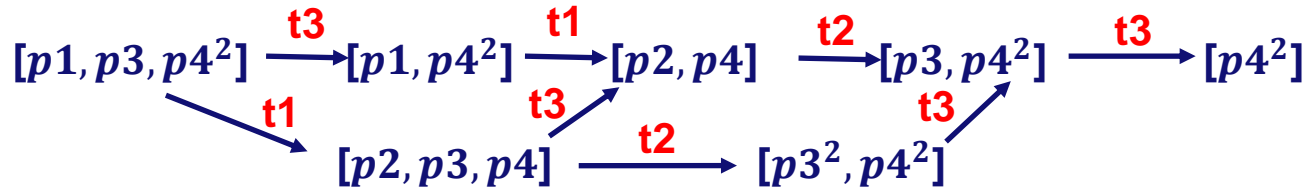
3. Give all reachable markings



Definition of Petri Net

Exercise 3 Solution

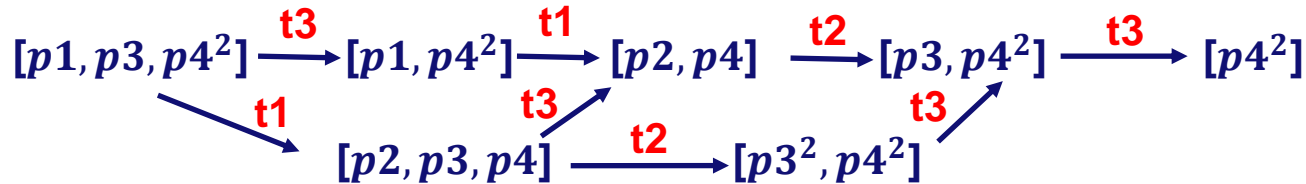
3. Give all reachable markings



Definition of Petri Net

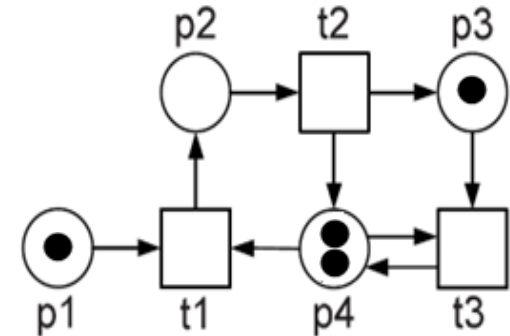
Exercise 3 Solution

3. Give all reachable markings



4. What are the reachable final markings?

$[p4^2]$



Modeling with Petri nets

Exercise 4

Consider the following scenario:

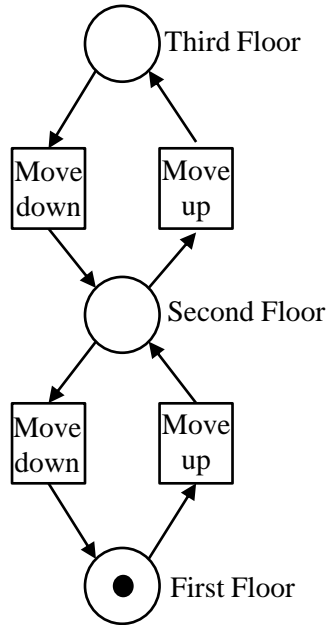
A building with three floors has an elevator. The elevator may move up and down between the floors and stop at any floor. Initially, the elevator resides on the first floor.

Model the behavior of the elevator as a Petri net. What do the places in your model express? What does a token in a place stand for?

Modeling with Petri nets

Exercise 4 (Solution)

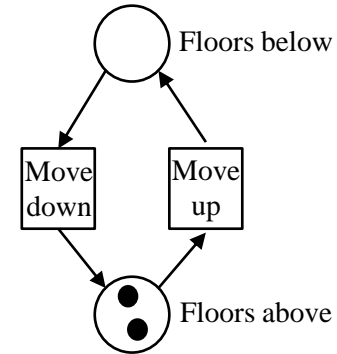
Two suggestions:



System Modeling Choices: What is a...

place? a location, constraint, queue, state...

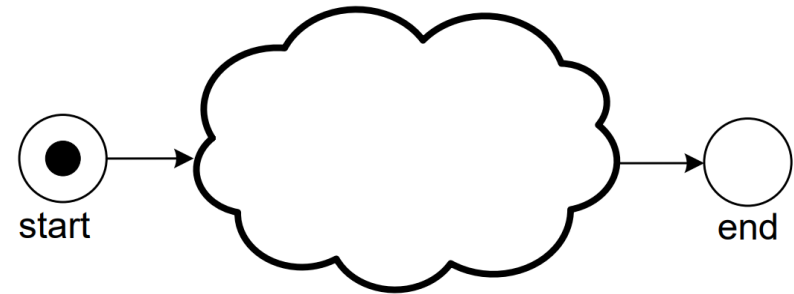
token? a resource, case, message, a state value...



Workflow Nets

Subclass of Petri nets

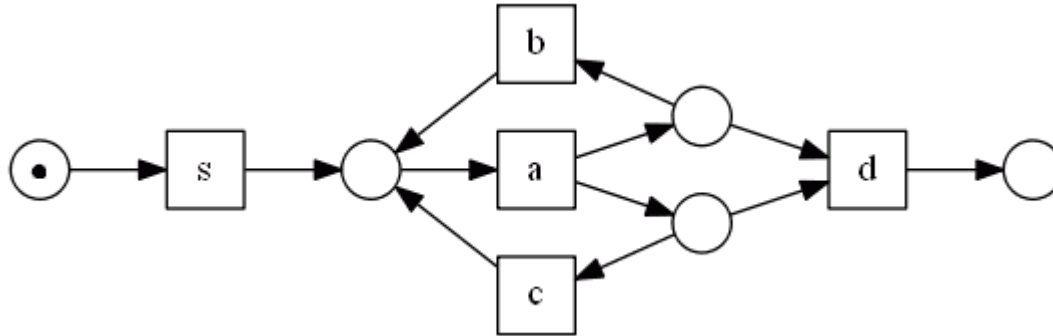
- One source place (no incoming arc)
- One sink place (no outgoing arc)
- All other nodes are on a path from source to sink



Workflow Nets

Exercise 5

Is this Petri net a workflow net?



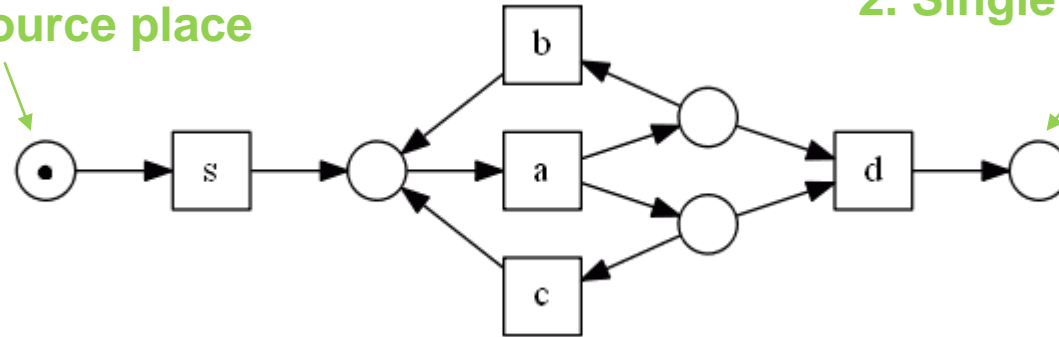
Workflow Nets

Exercise 5 Solution

Is this Petri net a workflow net?

1. Single source place

2. Single sink place



Workflow Nets

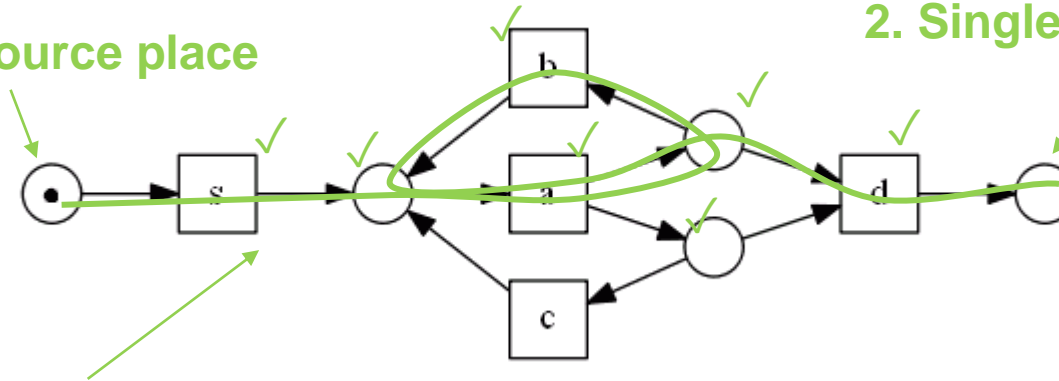
Exercise 5 Solution

Is this Petri net a workflow net?

1. Single source place

2. Single sink place

3. All other nodes are on a path from source to sink



Workflow Nets

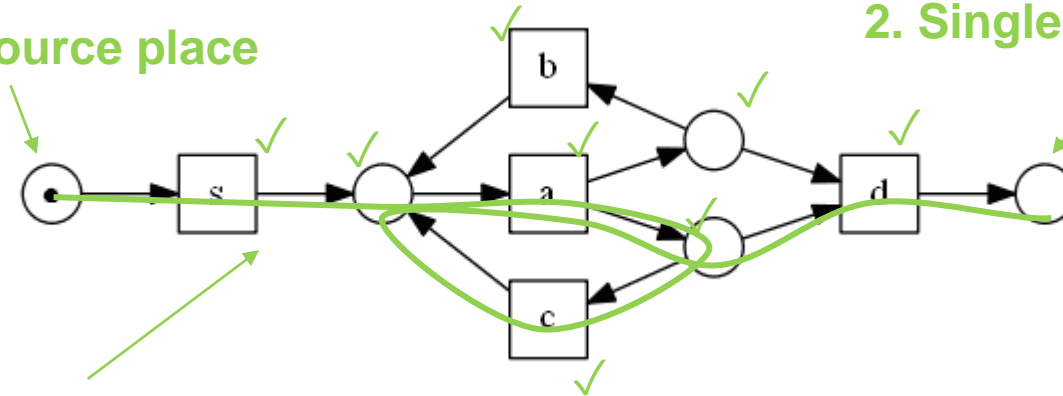
Exercise 5 Solution

Is this Petri net a workflow net?

1. Single source place

2. Single sink place

3. All other nodes are on a path from source to sink



Modeling Workflows

Exercise 6

Consider the following scenario:

A carpenter sells standard goods from a catalog on the one hand, and on the other hand takes orders for handmade unique pieces. After an order is placed, the carpenter sends a confirmation to the customer. Afterwards, two things are worked on concurrently: Firstly, a delivery date is scheduled with a shipping company. At the same time, the order is picked from stock in case of a catalog order, and manufactured in case of a customized order. Finally, the furniture is shipped to the customer.

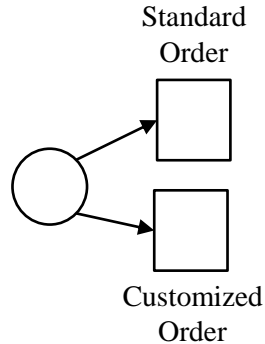
Model the scenario with a workflow.

Are there multiple ways of doing this?

Modeling Workflows

Exercise 6 Solution

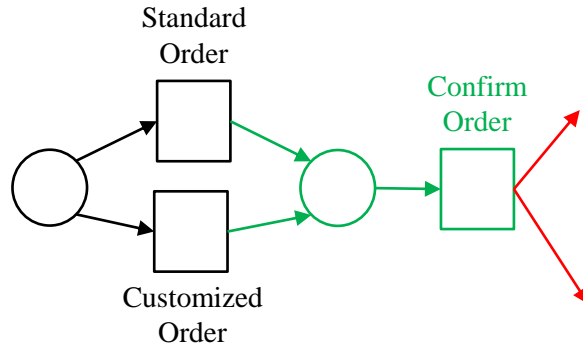
A carpenter sells standard goods from a catalog on the one hand, and on the other hand takes orders for handmade unique pieces.



Modeling Workflows

Exercise 6 Solution

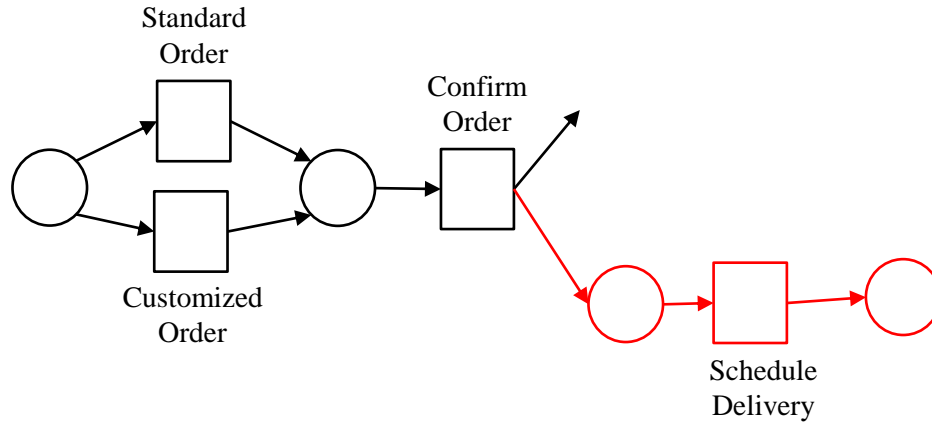
After an order is placed, the carpenter sends a confirmation to the customer. Afterwards, two things are worked on concurrently.



Modeling Workflows

Exercise 6 Solution

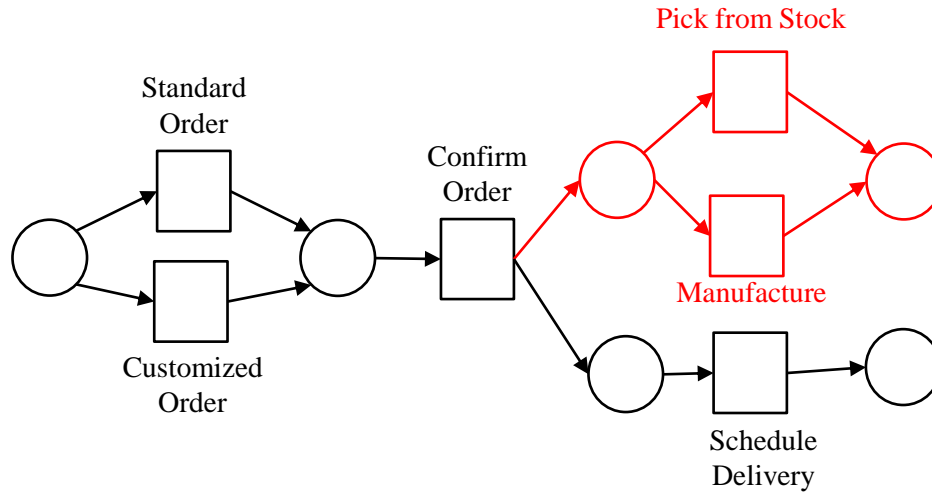
Firstly, a **delivery date is scheduled** with a shipping company.



Modeling Workflows

Exercise 6 Solution

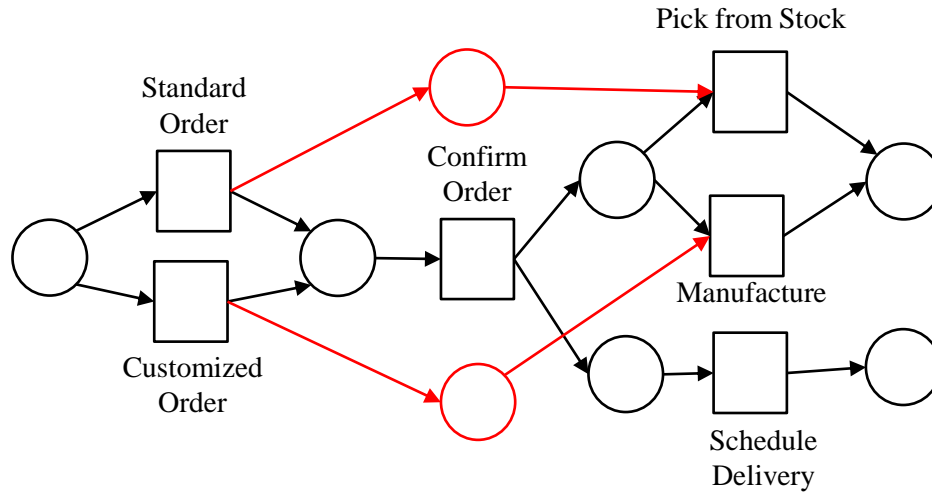
At the same time, the order is **picked from stock** in case of a catalog order, and **manufactured** in case of a customized order.



Modeling Workflows

Exercise 6 Solution

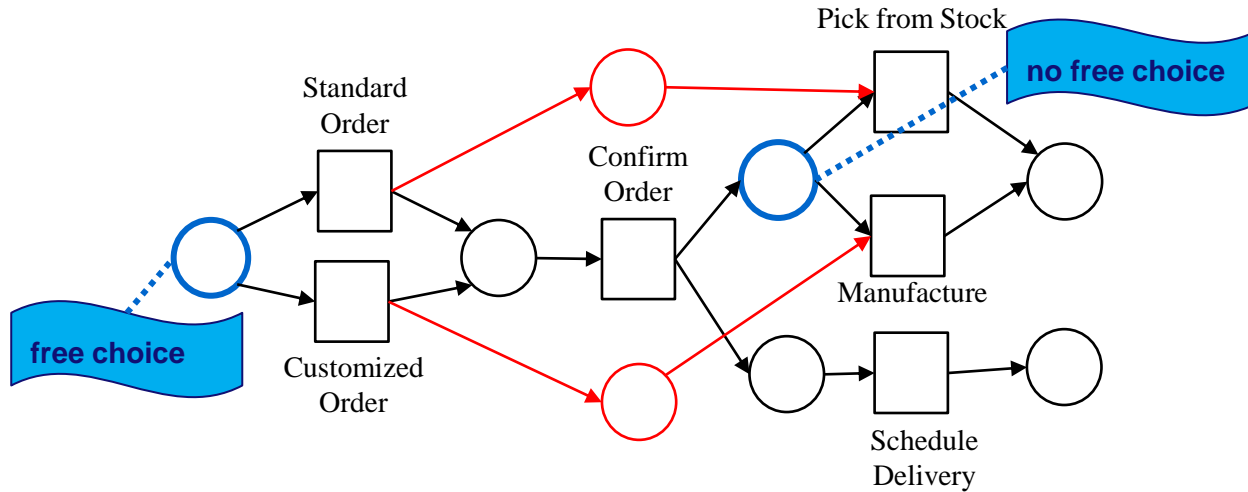
At the same time, the order is picked from stock **in case of a catalog order**, and manufactured **in case of a customized order**.



Modeling Workflows

Exercise 6 Solution

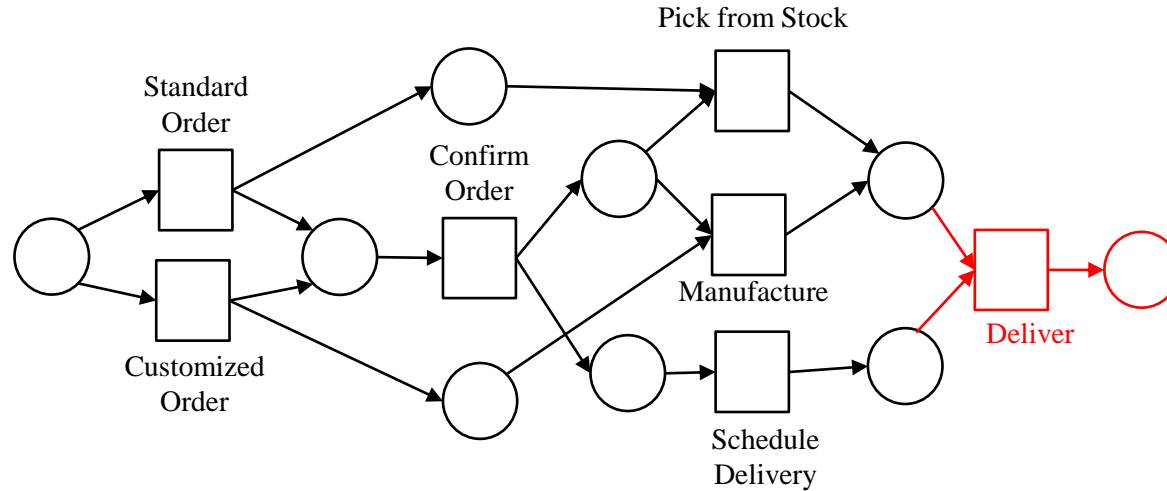
At the same time, the order is picked from stock **in case of a catalog order**, and manufactured **in case of a customized order**.



Modeling Workflows

Exercise 6 Solution

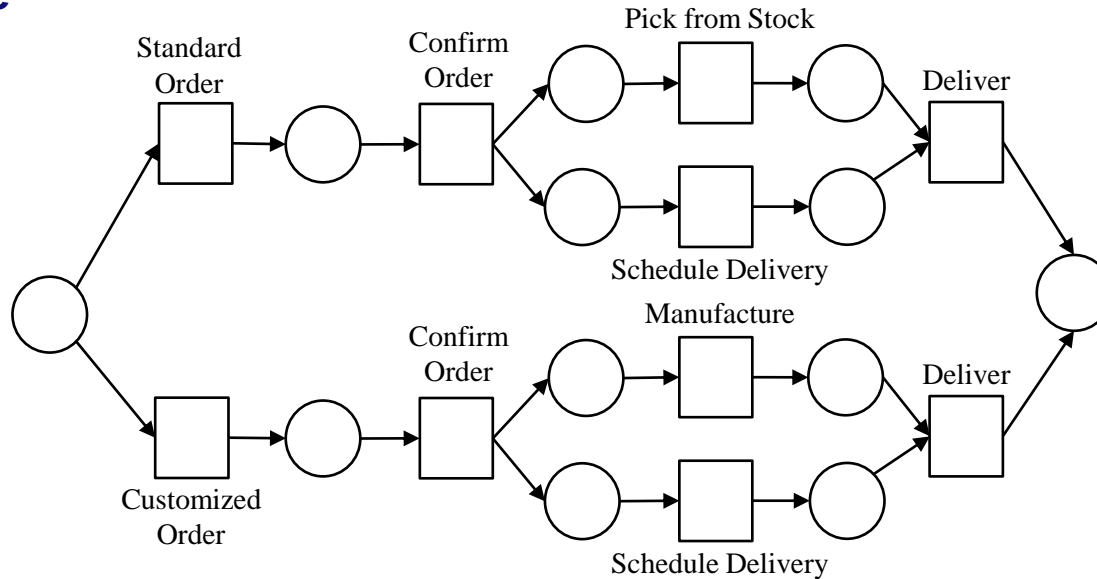
Finally, the furniture is **shipped** to the customer.



Modeling Workflows

Exercise 6 Solution

**One possible
alternative:**



**Next week:
Comparison.**

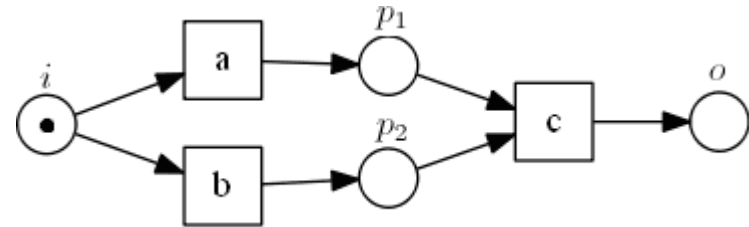
Soundness of Workflow Nets

- **Safeness:** Places cannot hold multiple tokens at the same time.
- **Proper completion:** If the sink place is marked, all other places are empty.
- **Option to complete:** From any reachable marking, it is possible to reach a marking which marks only the sink place. (this criterion actually implies proper completion)
- **Absence of dead parts:** For any transition there is a firing sequence enabling it.

Sound Workflow Nets

Exercise 7

Is this Petri net a sound workflow net? Why?

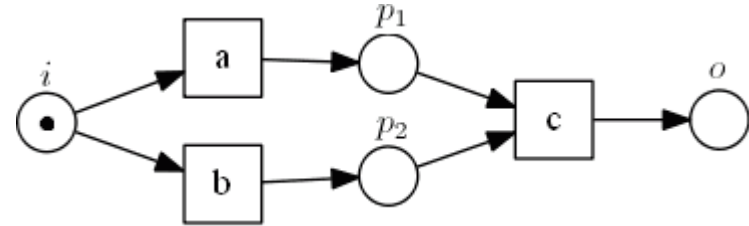


Sound Workflow Nets

Exercise 7 Solution

Is a workflow net

Option to complete is violated

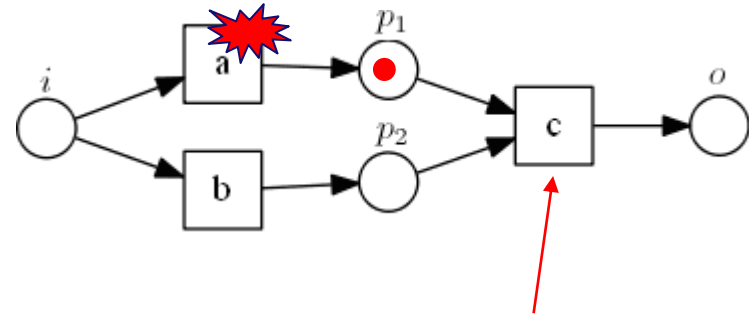


Sound Workflow Nets

Exercise 7 Solution

Is a workflow net

Option to complete is violated

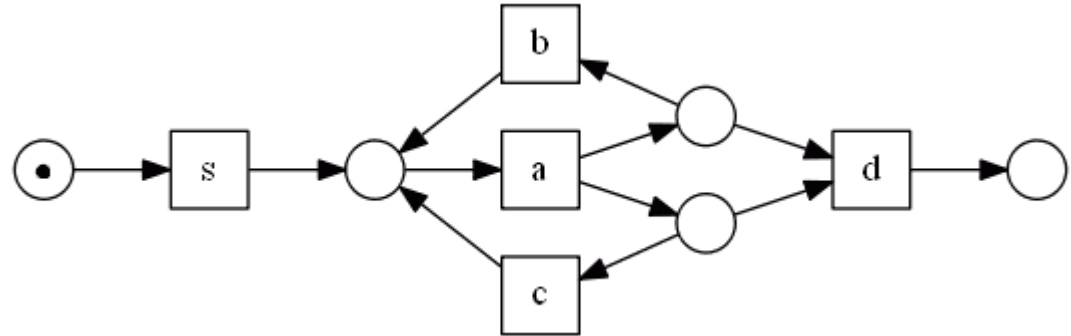


Is not enabled, thus, we can not reach the marking $[o]$

Sound Workflow Nets

Exercise 8

Is this Petri net a sound workflow net? Why?

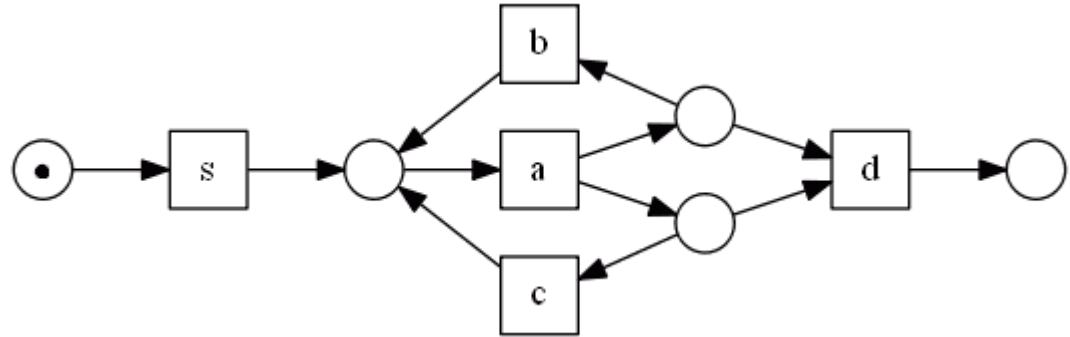


Sound Workflow Nets

Exercise 7 Solution

**Safeness &
Proper completion
are violated**

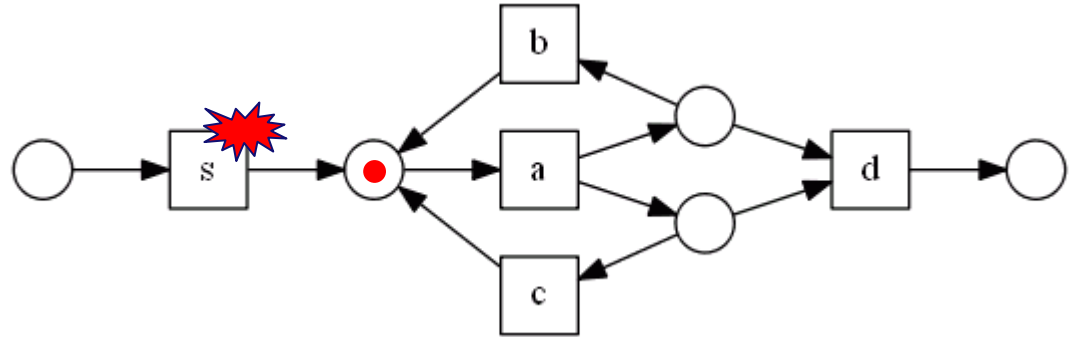
(e.g., repeatedly looping transitions a and b)



Sound Workflow Nets

Exercise 7 Solution

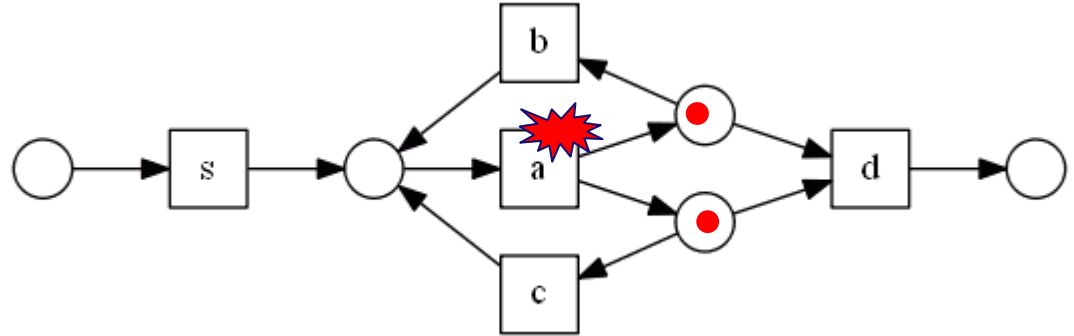
**Safeness &
Proper completion
are violated
(e.g., repeatedly looping transitions a and b)**



Sound Workflow Nets

Exercise 7 Solution

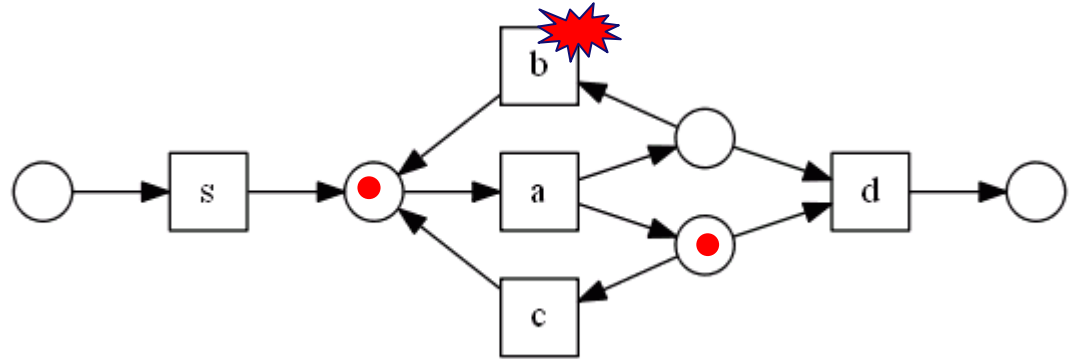
**Safeness &
Proper completion
are violated
(e.g., repeatedly looping transitions a and b)**



Sound Workflow Nets

Exercise 7 Solution

**Safeness &
Proper completion
are violated
(e.g., repeatedly looping transitions a and b)**

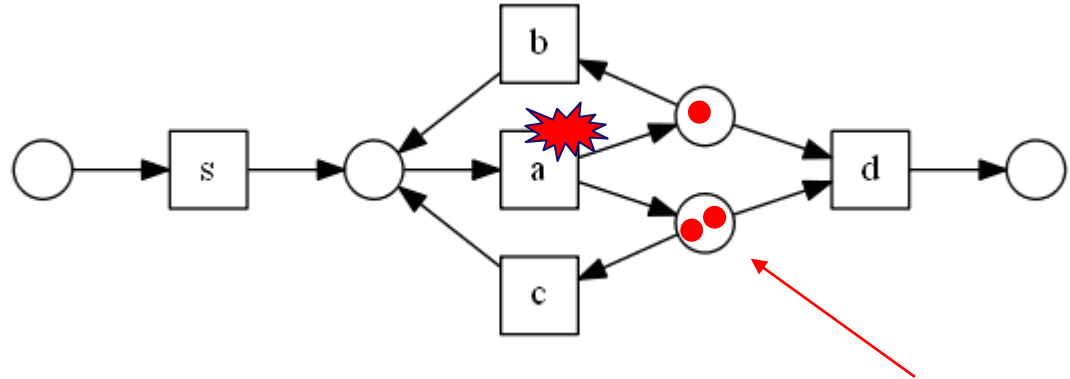


Sound Workflow Nets

Exercise 7 Solution

**Safeness &
Proper completion
are violated**

(e.g., repeatedly looping transitions a and b)



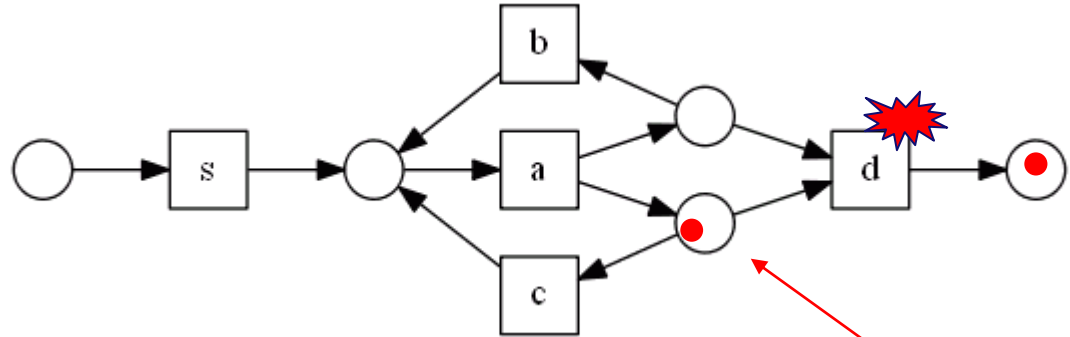
**More than
one token**

Sound Workflow Nets

Exercise 7 Solution

**Safeness &
Proper completion
are violated**

(e.g., repeatedly looping transitions a and b)

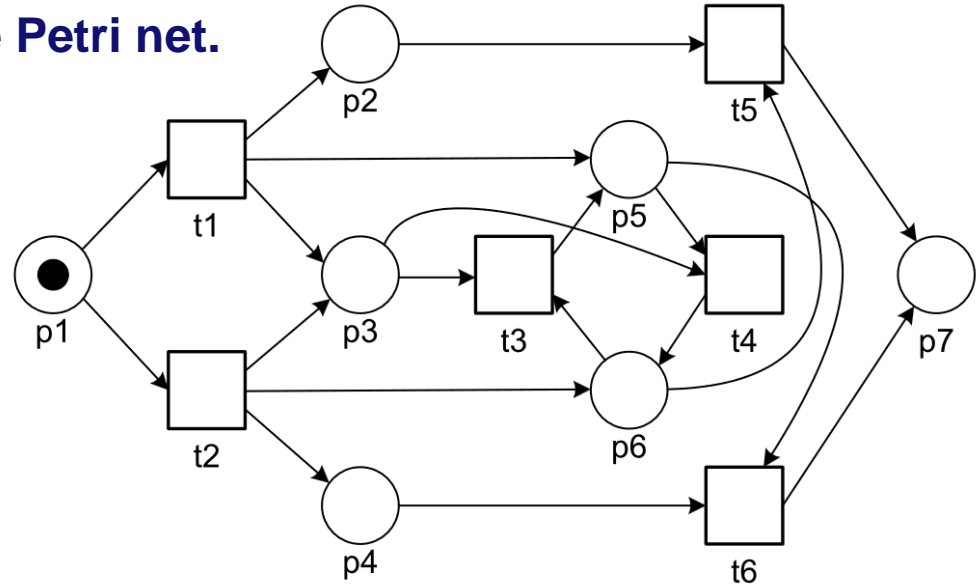


**Is not
empty**

Sound Workflow Nets

Exercise 8

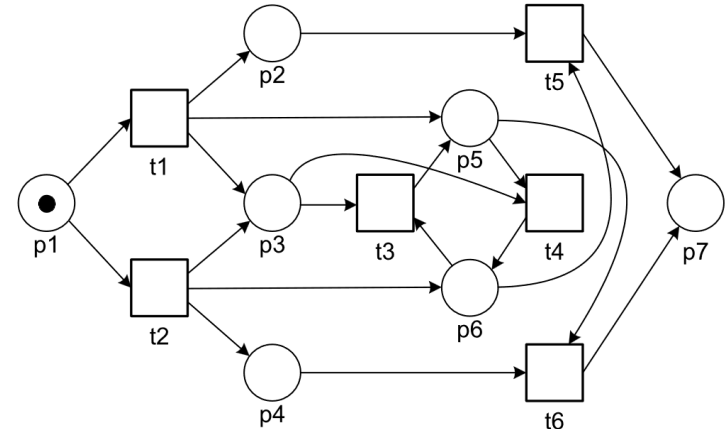
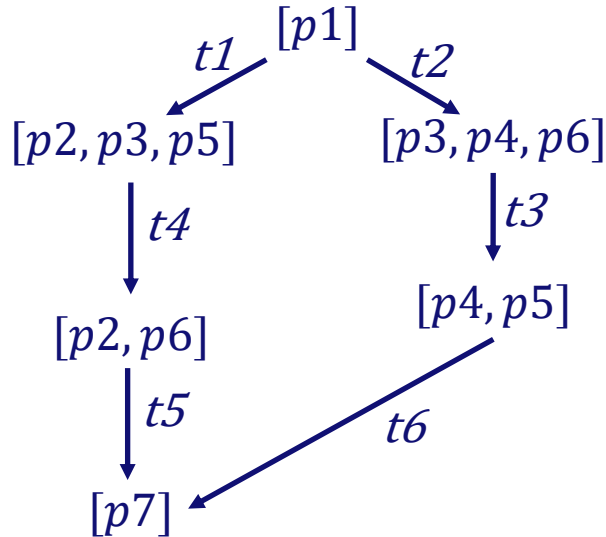
**Draw the reachability graph of the Petri net.
Is it a sound workflow net?**



Sound Workflow Nets

Exercise 8 (Solution)

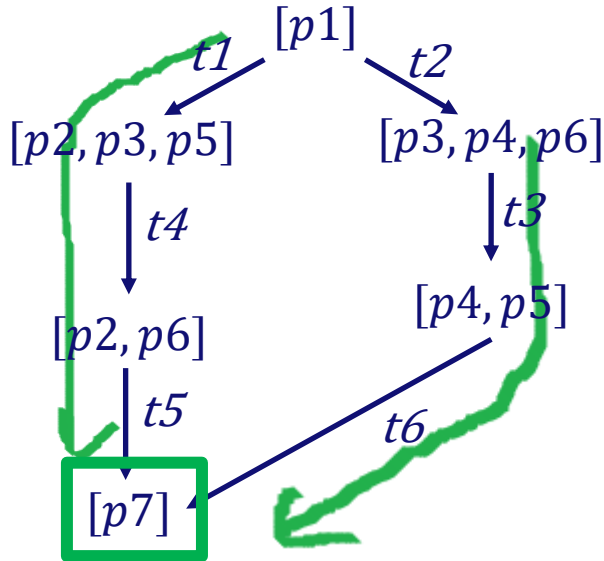
Draw the reachability graph of the Petri net.
Is it a sound workflow net?



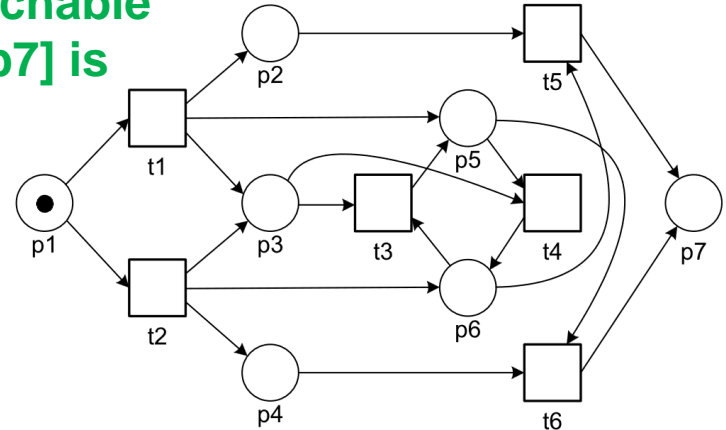
Sound Workflow Nets

Exercise 8 (Solution)

Draw the reachability graph of the Petri net.
Is it a sound workflow net?



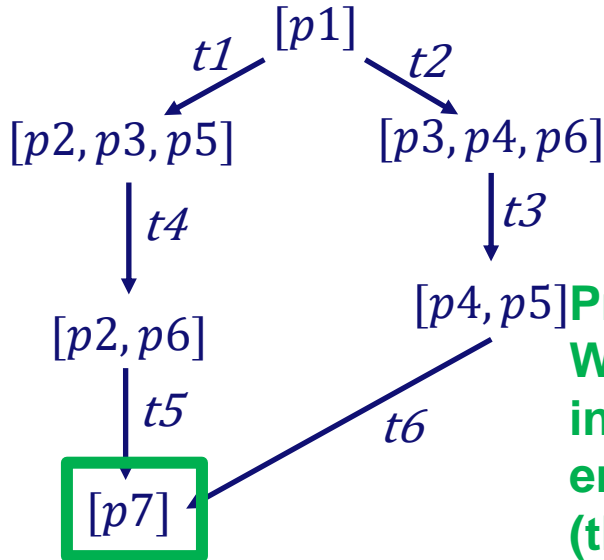
Option to Complete:
From all reachable
markings, $[p7]$ is
reachable.



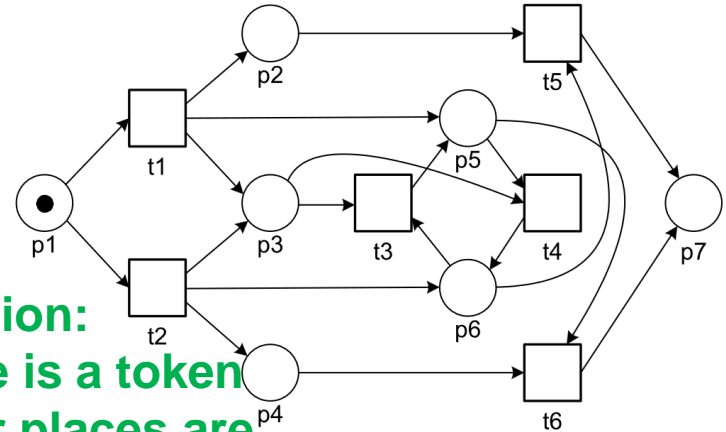
Sound Workflow Nets

Exercise 8 (Solution)

Draw the reachability graph of the Petri net.
Is it a sound workflow net?



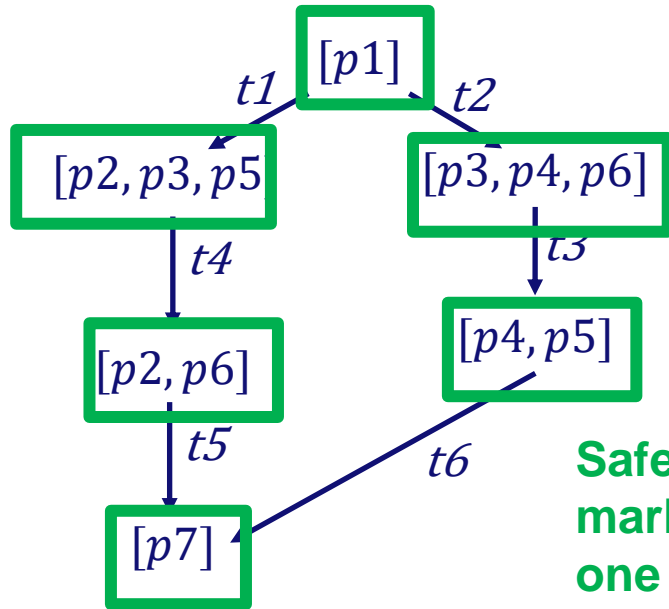
Proper completion:
Whenever there is a token
in $[p7]$, all other places are
empty.
(this is also implied by the
Option to Complete)



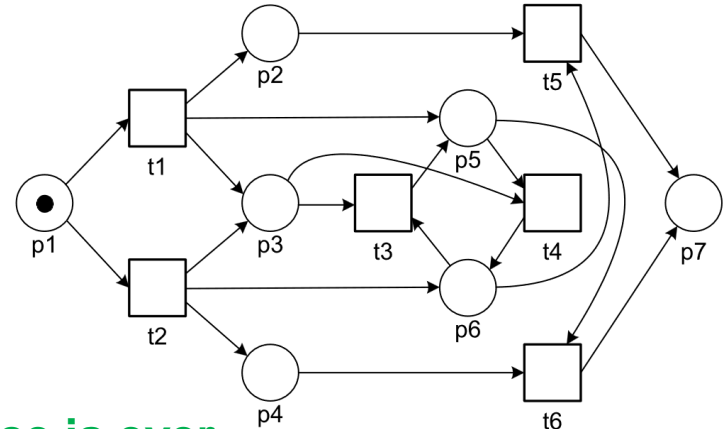
Sound Workflow Nets

Exercise 8 (Solution)

Draw the reachability graph of the Petri net.
Is it a sound workflow net?



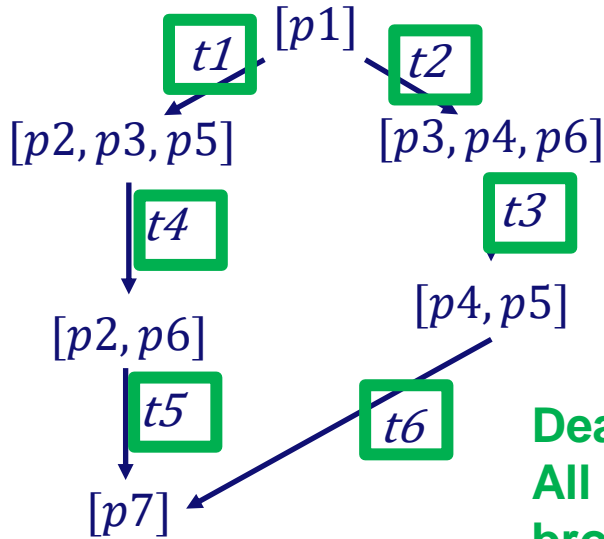
Safeness: No place is ever marked with more than one token.



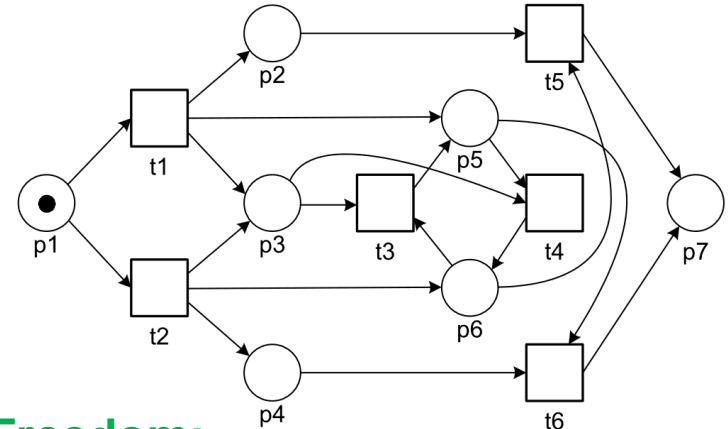
Sound Workflow Nets

Exercise 8 (Solution)

Draw the reachability graph of the Petri net.
Is it a sound workflow net?



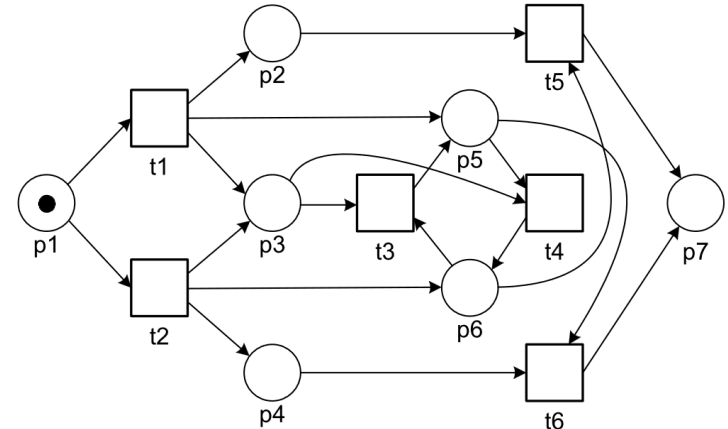
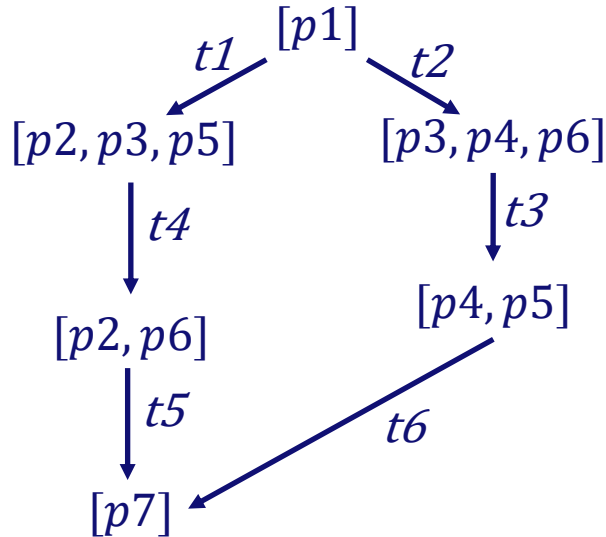
Dead Transition Freedom:
All transitions can be brought to firing.



Sound Workflow Nets

Exercise 8 (Solution)

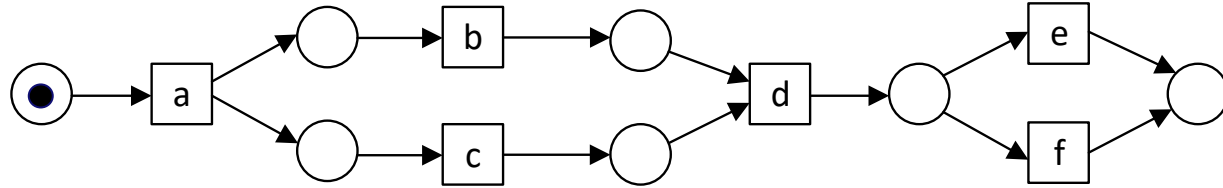
Draw the reachability graph of the Petri net.
Is it a sound workflow net? **Yes.**



Complete Traces

Complete Traces: also known as traces **accepted** by the model or **language** of the model.

Model 1



Criterion:

$\langle a_1, \dots, a_n \rangle$ is a complete trace if there is a firing sequence leading from the initial marking to the final marking where the labels of the transitions in the firing sequence are a_1, \dots, a_n in the order of their occurrence.

Example for Model 1:

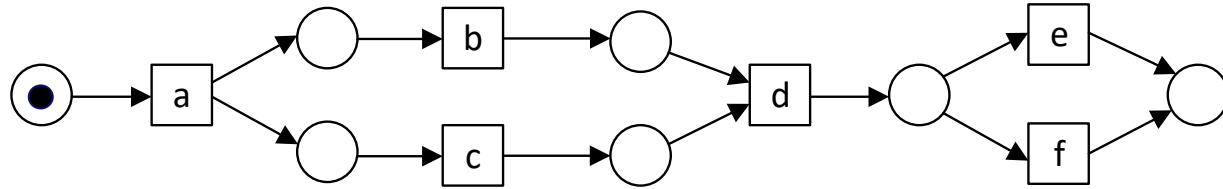
$\langle a, b, c, d, e \rangle$, $\langle a, c, b, d, f \rangle$

Complete Traces - Count

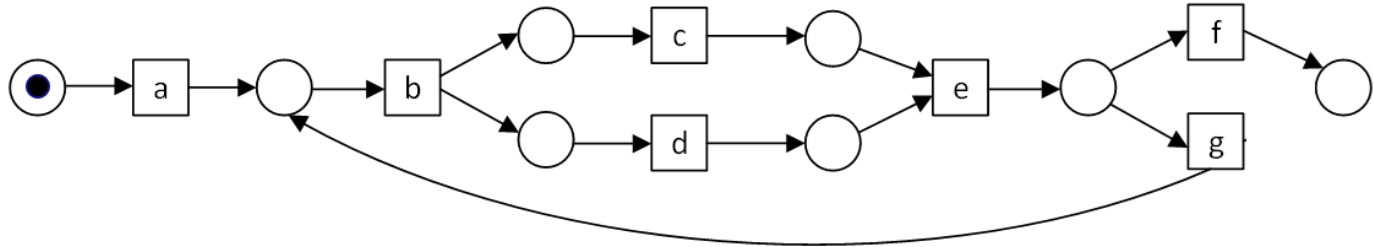
Exercise 9

How many complete traces are possible in the following models?

Model 1



Model 2



Complete Traces - Count

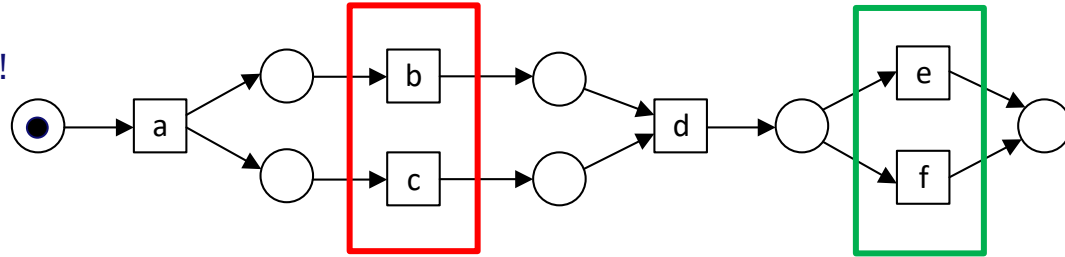
Exercise 9 (Solution)

How many complete traces are possible based on the following models?

$\{b, c\} \rightarrow$ in any order $\rightarrow 2!$

$\{e, f\} \rightarrow$ two choices $\rightarrow 2$

Model 1: $2! \times 2! = 4$



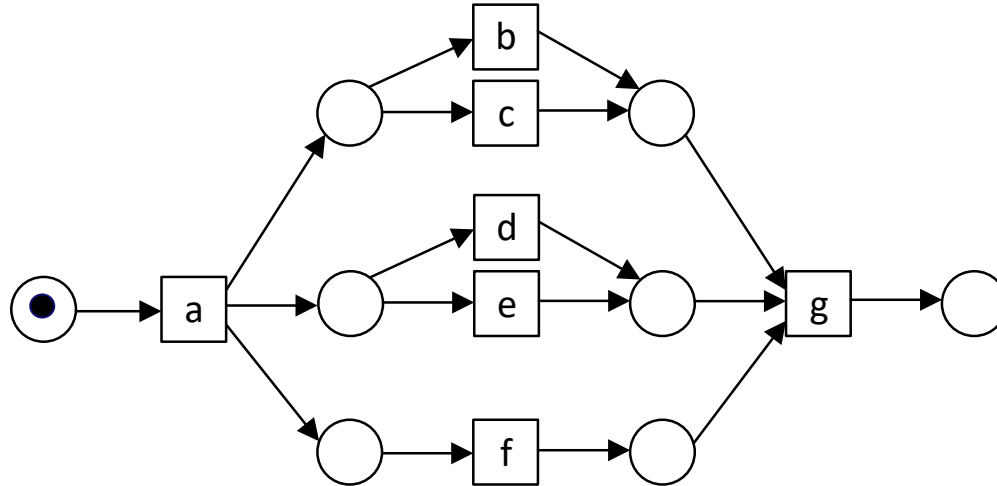
There are infinitely many. For example:

$\langle a, (b, c, e, g)^n, f \rangle$ for all $n \geq 1$.

Complete Traces - Count

Exercise 10

How many complete traces are possible in the following model?



Complete Traces - Count

Exercise 10 (Solution)

How many complete traces are possible based on the following model?

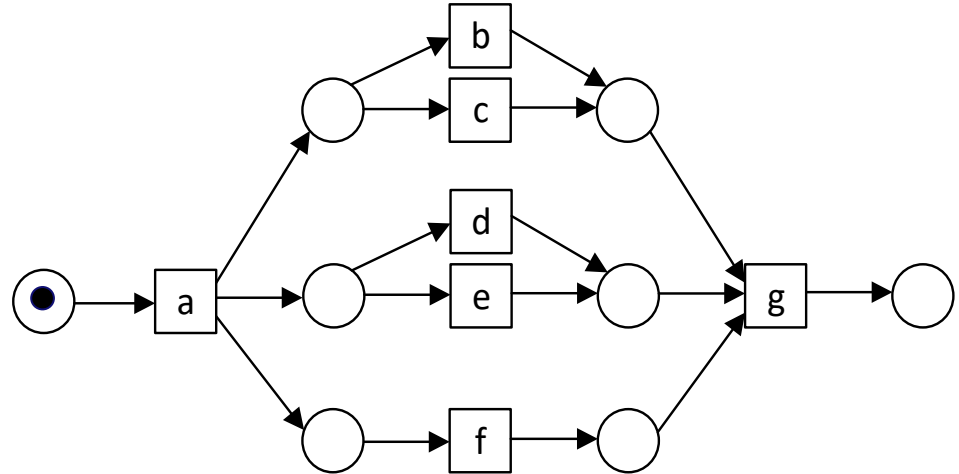
$\{b, d, f\} \rightarrow$ in any order $\rightarrow 3!$

$\{b, e, f\} \rightarrow$ in any order $\rightarrow 3!$

$\{c, e, f\} \rightarrow$ in any order $\rightarrow 3!$

$\{c, d, f\} \rightarrow$ in any order $\rightarrow 3!$

$$3! \times 4 = 24$$



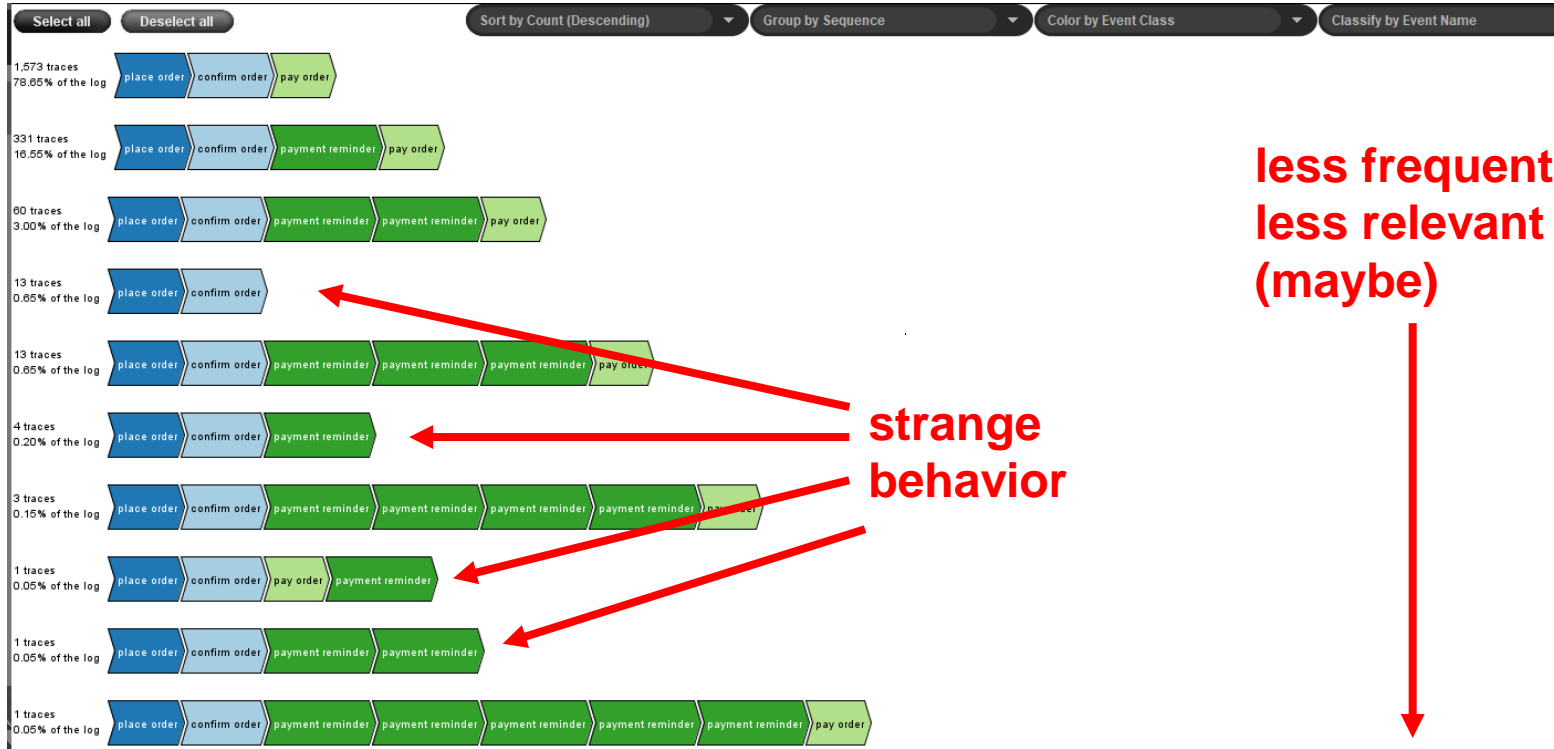
Event Data Preprocessing in

Two Plugins for Filtering / Noise Handling



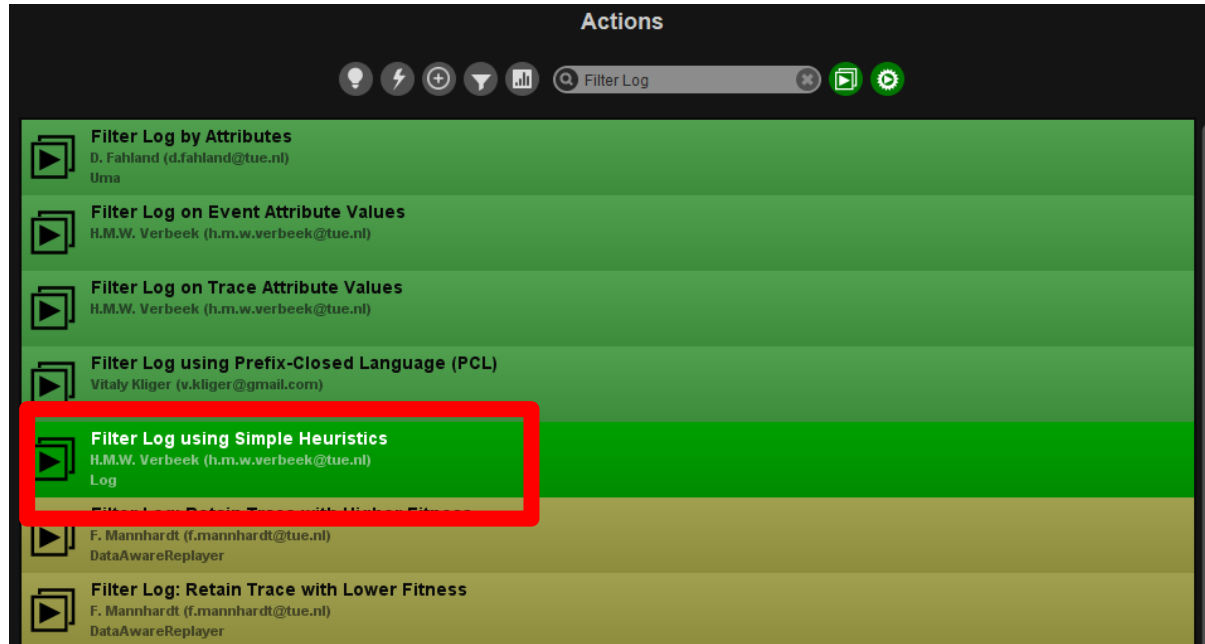
Event Data Preprocessing in ProM

Two Plugins for Filtering / Noise Handling



Event Data Preprocessing in ProM

Two Plugins for Filtering / Noise Handling

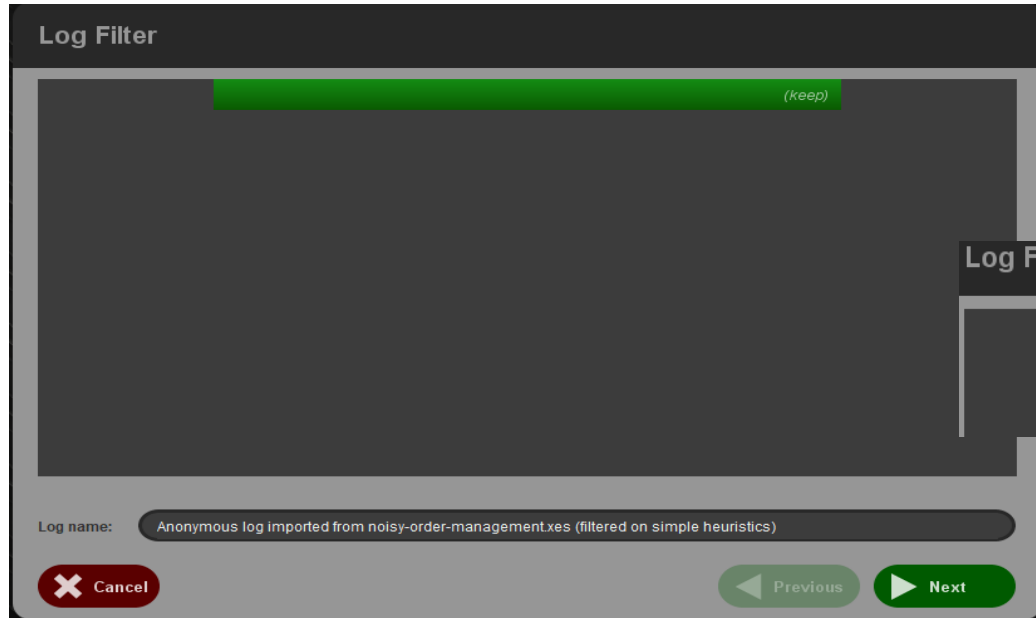


Event Data Preprocessing in ProM

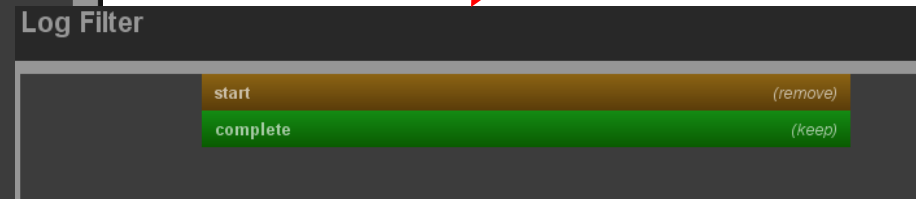
Two Plugins for Filtering / Noise Handling

Filter Events based on Lifecycle Information.

In this log, there are no lifecycle information at events.



If there are, it will look like this.



Event Data Preprocessing in ProM


Two Plugins for Filtering / Noise Handling

Only keep instances (traces / cases) with a certain start or end event.
Note: The activities based on which we filter out (here: confirm order, payment reminder) are still included in the remaining traces.




Log Filter

Start events
Only instances starting with a green event will be used.

place order+

Select top percentage:  80

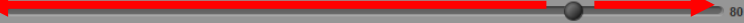
Log name: Anonymous log imported from noisy-order-management.xes (filtered on simple heuristics) (filtered on simple heuristics)

 Cancel  Previous  Next



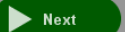
Log Filter

End events
Only instances ending with a green event will be used.

confirm order+
pay order+
payment reminder+

Select top percentage:  80

Log name: Anonymous log imported from noisy-order-management.xes (filtered on simple heuristics) (filtered on simple heuristics)

 Cancel  Previous  Next

Event Data Preprocessing in ProM

Two Plugins for Filtering / Noise Handling

Filter on the **event level** (not case level!)

Log Filter

Event filter

Only green events will be used.

confirm order+

pay order+

payment reminder+

place order+

Select top percentage: 80

Log name: Anonymous log imported from noisy-order-management.xes (filtered on simple heuristics)

Cancel

Previous

Finish

Event Data Preprocessing in ProM

Two Plugins for Filtering / Noise Handling

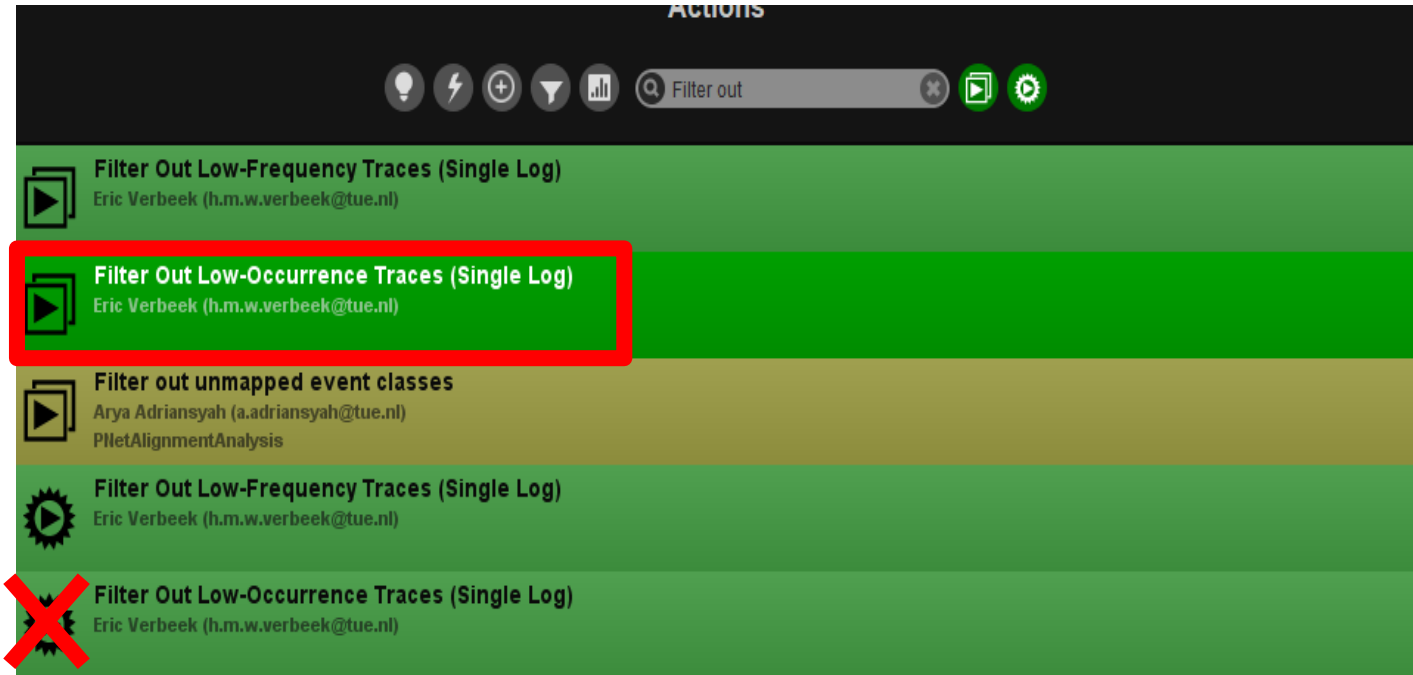
Result:



Next:
Filter out
infrequent
traces.

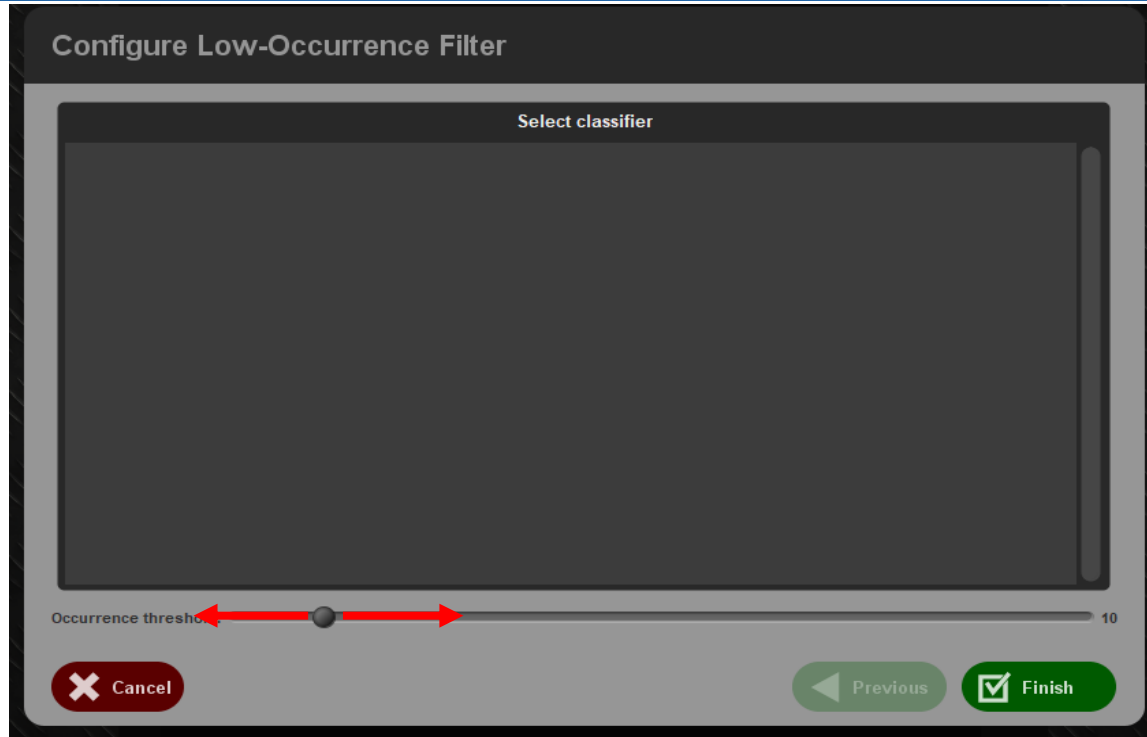
Event Data Preprocessing in ProM

Two Plugins for Filtering / Noise Handling



Event Data Preprocessing in ProM

Two Plugins for Filtering / Noise Handling



**Discard
traces for
which that
variant
occurs less
than 10
times.**

Event Data Preprocessing in ProM

Two Plugins for Filtering / Noise Handling

Result:

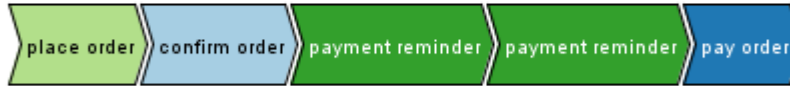
1,573 traces
79.56% of the log



331 traces
16.74% of the log



60 traces
3.03% of the log



13 traces
0.66% of the log

