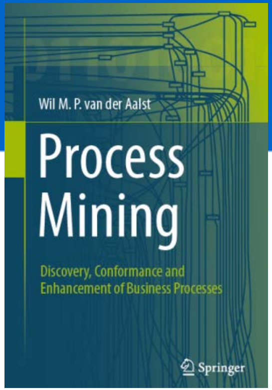
Process Mining: Data Science in Action

Transition Systems and Petri Net Properties

prof.dr.ir. Wil van der Aalst

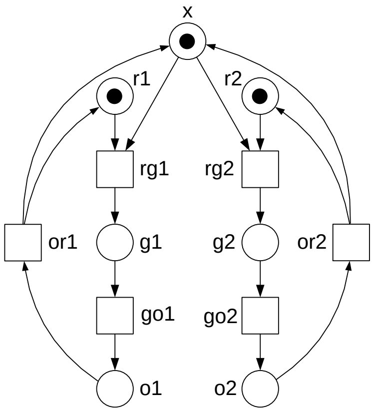
www.processmining.org



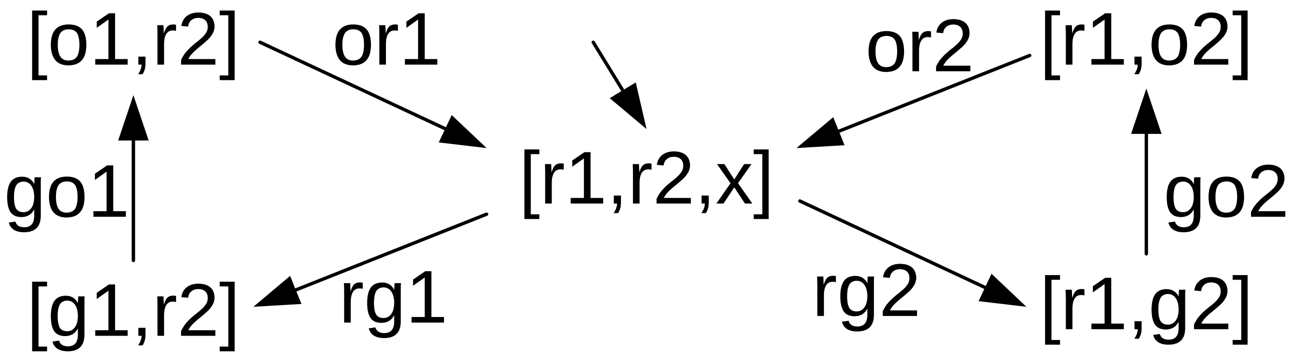
TU/e Technische Universiteit **Eindhoven** University of Technology

Where innovation starts

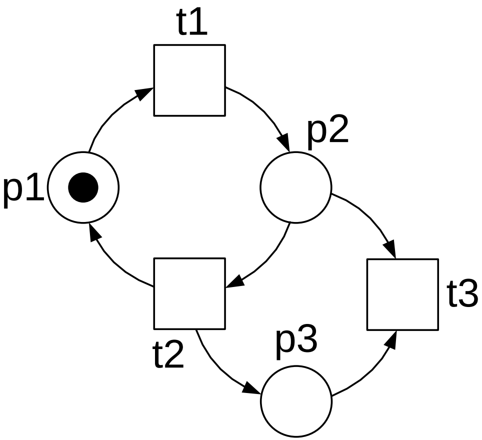
Reachability graph of a Petri net



The reachability graph is a transition system with one initial state (initial marking) and no explicit final marking.



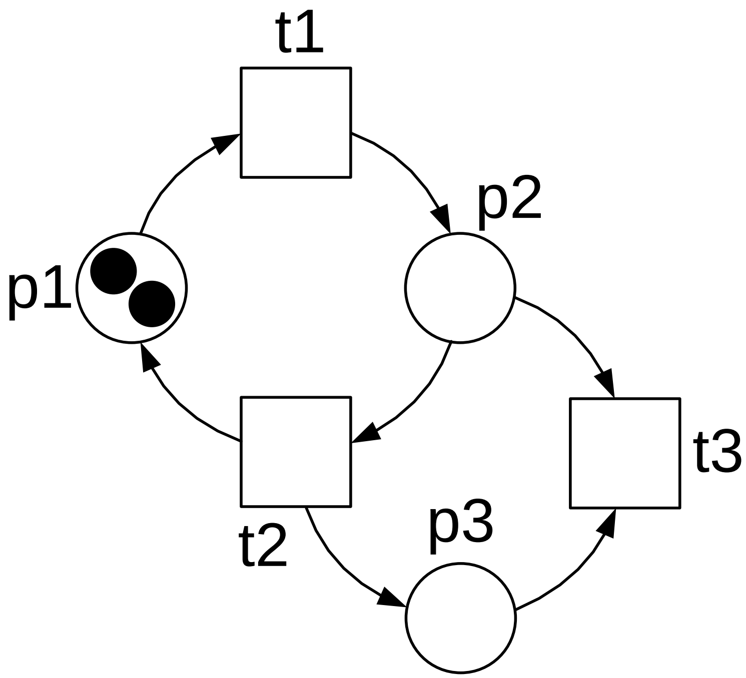
Reachability graph may be infinite



$$\begin{tikzcd}
{[p1]} \arrow[r, "t1"] & {[p2]} \arrow[r, "t2"] & {[p1,p3]} \arrow[r, "t1"] & {[p2,p3]} \arrow[r, "t2"] \arrow[d, "t3"'] & {[p1,p3^2]} \arrow[r, "t1"] & {[p2,p3^2]} \arrow[r, "t2"] \arrow[d, "t3"'] & \ldots \\
 & & & {[\ ]} & & {[p3]} &
\end{tikzcd}$$

Boundedness

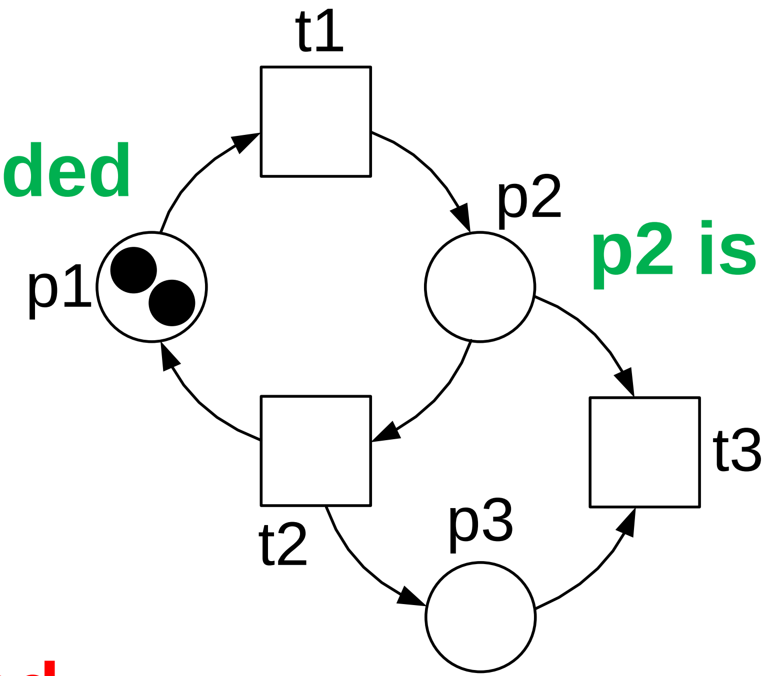
- A place *p* is *k*-bounded if there is no reachable marking with more than *k* tokens in *p*.
- A Petri net is *k*-bounded if all places are *k*-bounded.
- A place/Petri net is bounded if there exists such a *k*.



Boundedness

p1 is 2-bounded

p2 is 2-bounded

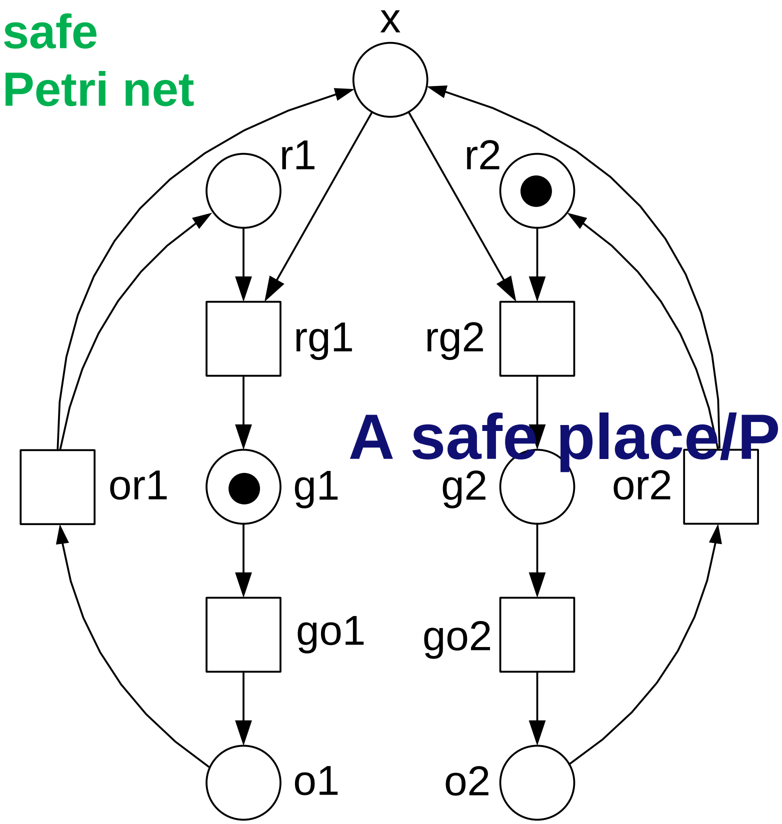


The whole Petri net is unbounded

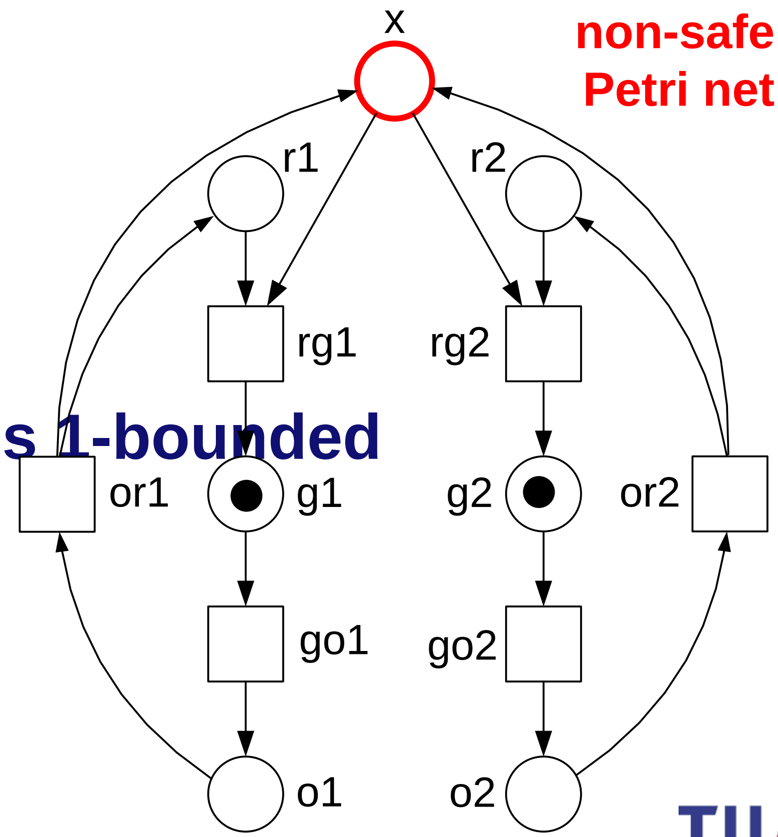
p3 is unbounded

Safeness (= 1-boundedness)

safe Petri net



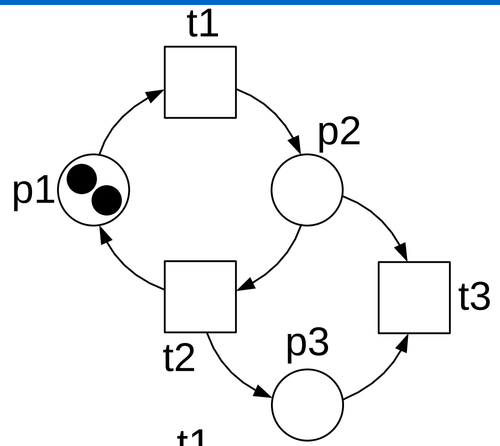
non-safe Petri net

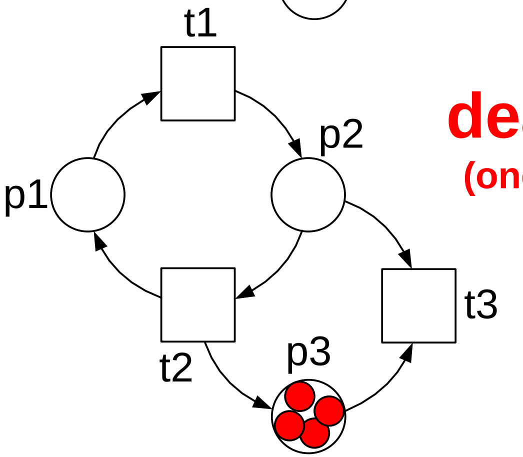


A safe place/Petri net is 1-bounded

Deadlock

- A marking is **dead** if no transition is enabled in it.
- A Petri net has a potential **deadlock** if there is a reachable dead marking.
- A Petri net is **deadlock-free** if each reachable marking enables at least one transition.



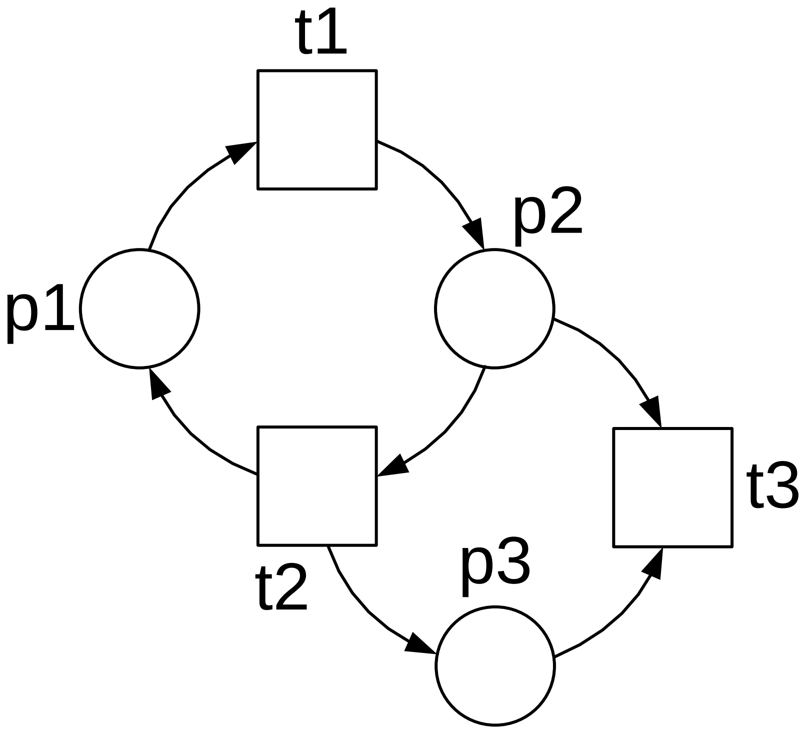


deadlock
(one of many)

Question

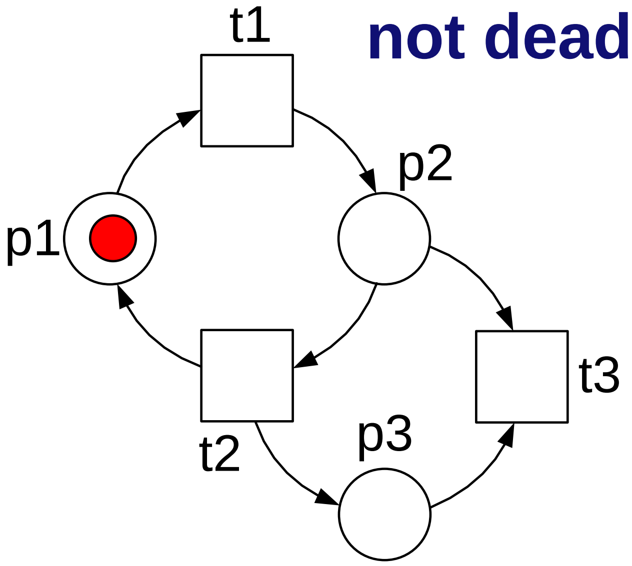
Provide (if possible) an initial making such that

- **the Petri net is unbounded and not deadlock-free,**
- **the Petri net is bounded and not deadlock-free,**
- **the Petri net is unbounded and deadlock-free, and**
- **the Petri net is bounded and deadlock-free.**

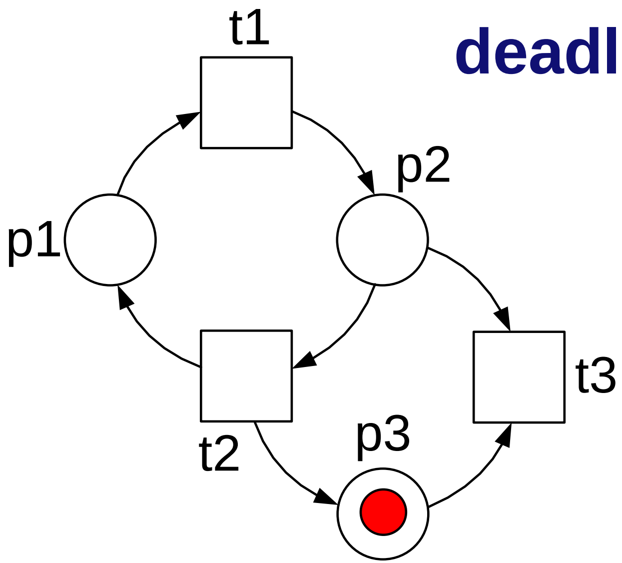


Answer

unbounded and not deadlock-free

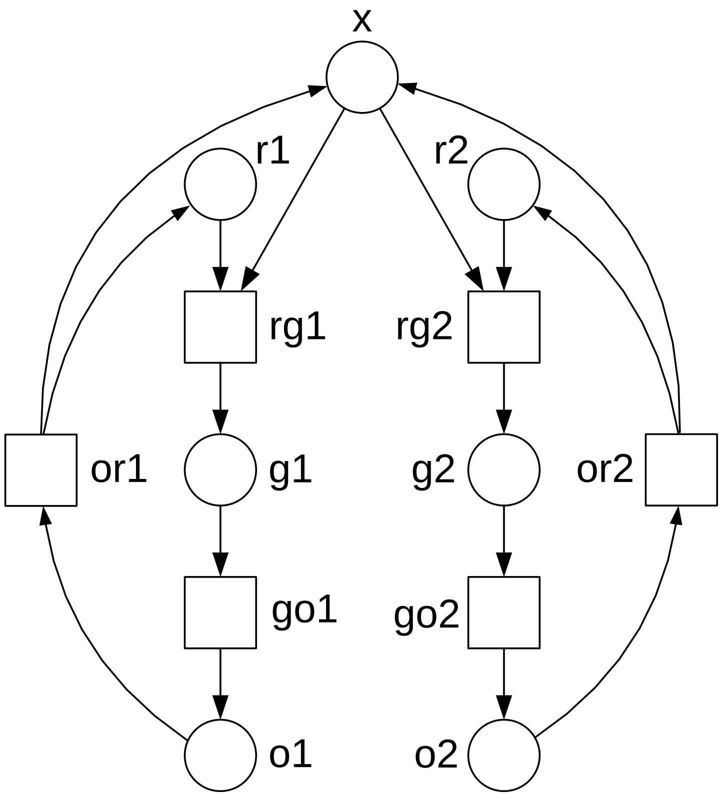


bounded but not deadlock-free



The net will always have a deadlock independent of the initial marking!

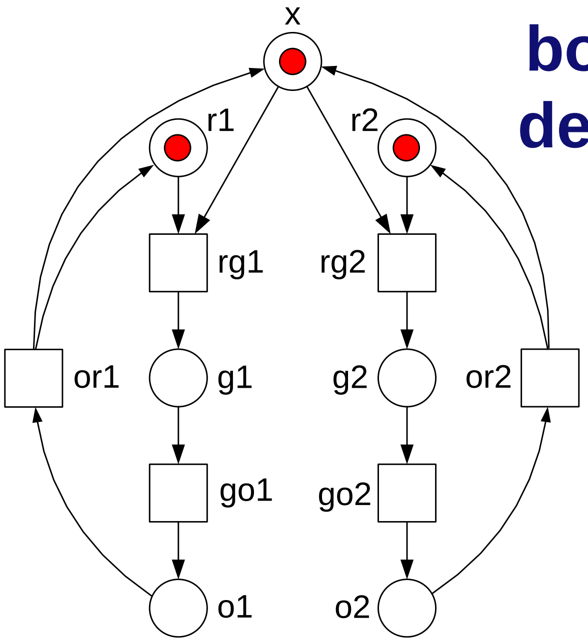
Question



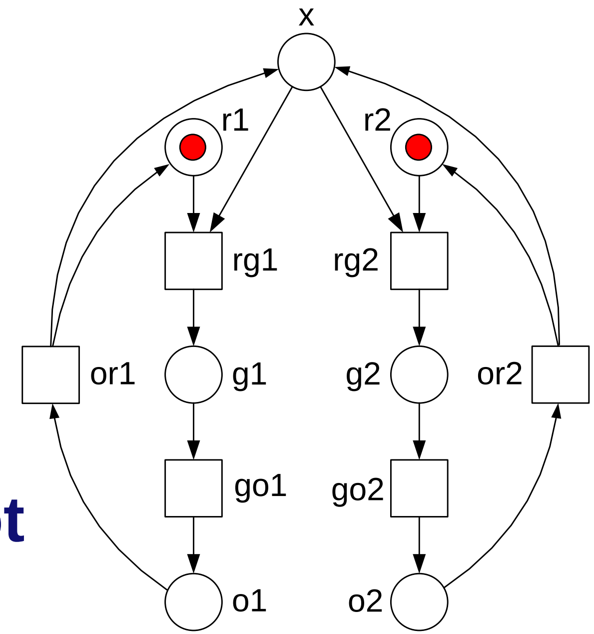
Provide (if possible) an initial making such that

- **the Petri net is unbounded and not deadlock-free,**
- **the Petri net is bounded and not deadlock-free,**
- **the Petri net is unbounded and deadlock-free, and**
- **the Petri net is bounded and deadlock-free.**

Answer



bounded and deadlock-free

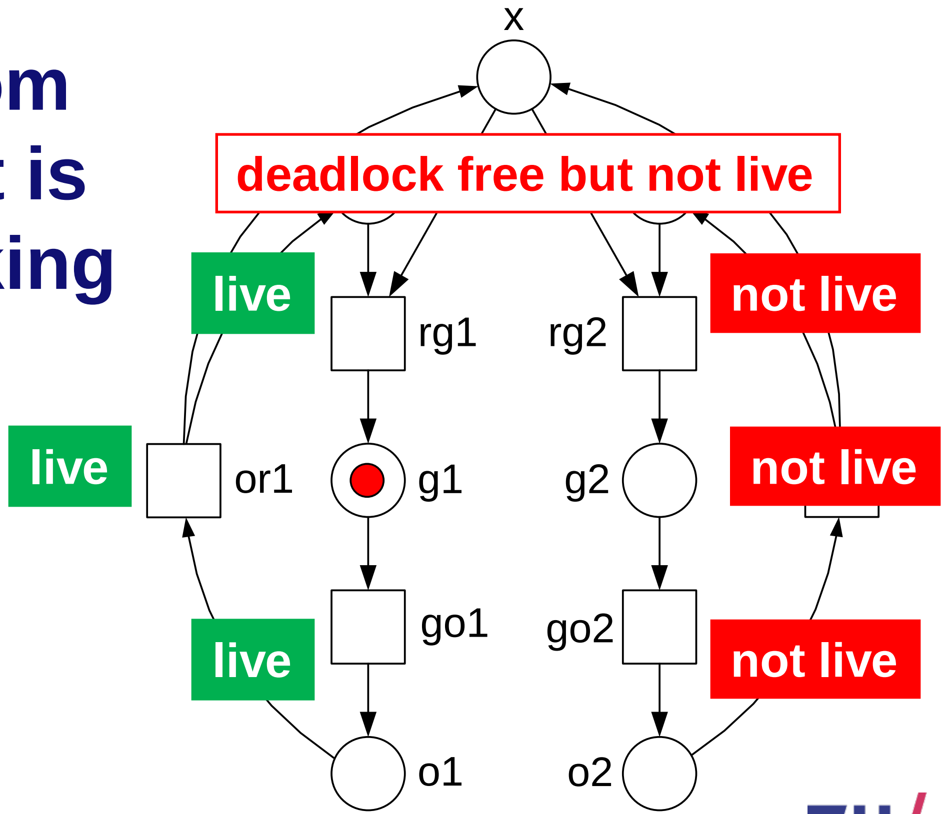


bounded, but not deadlock-free

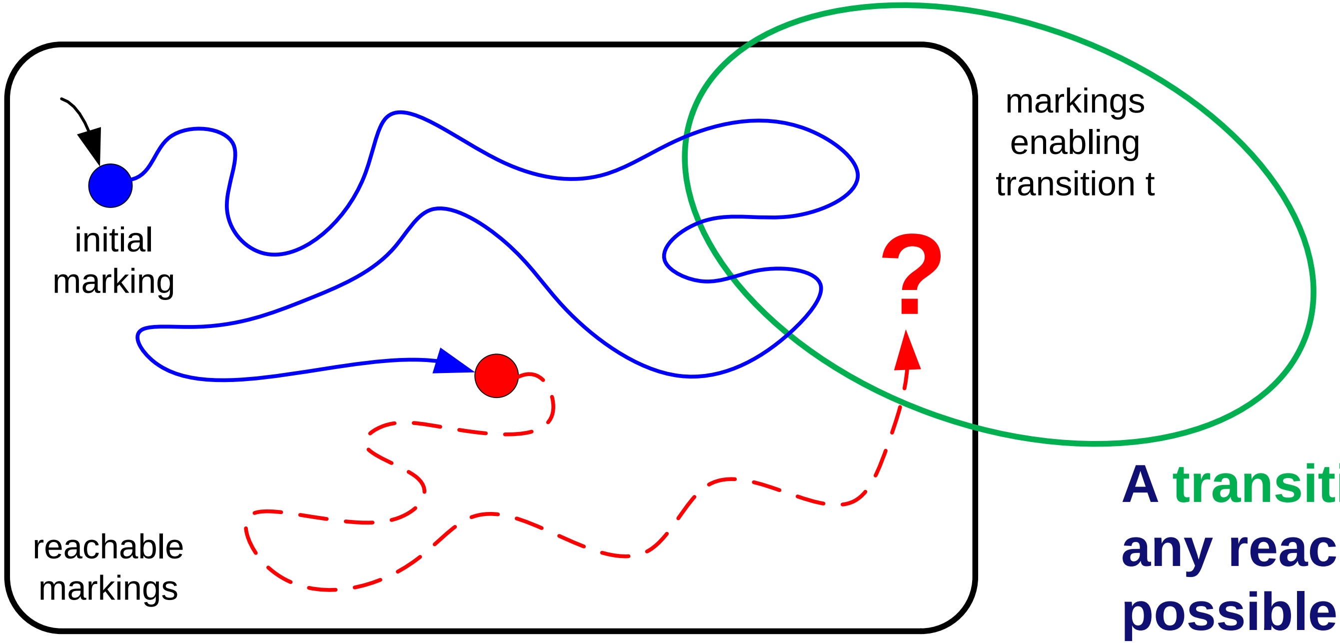
The net will always be bounded independent of the initial marking!

Liveness

- A transition *t* is live if from any reachable marking it is possible to reach a marking that enables *t*.
- A Petri net is live if all transitions are live.
- A Petri net that is live is deadlock-free.



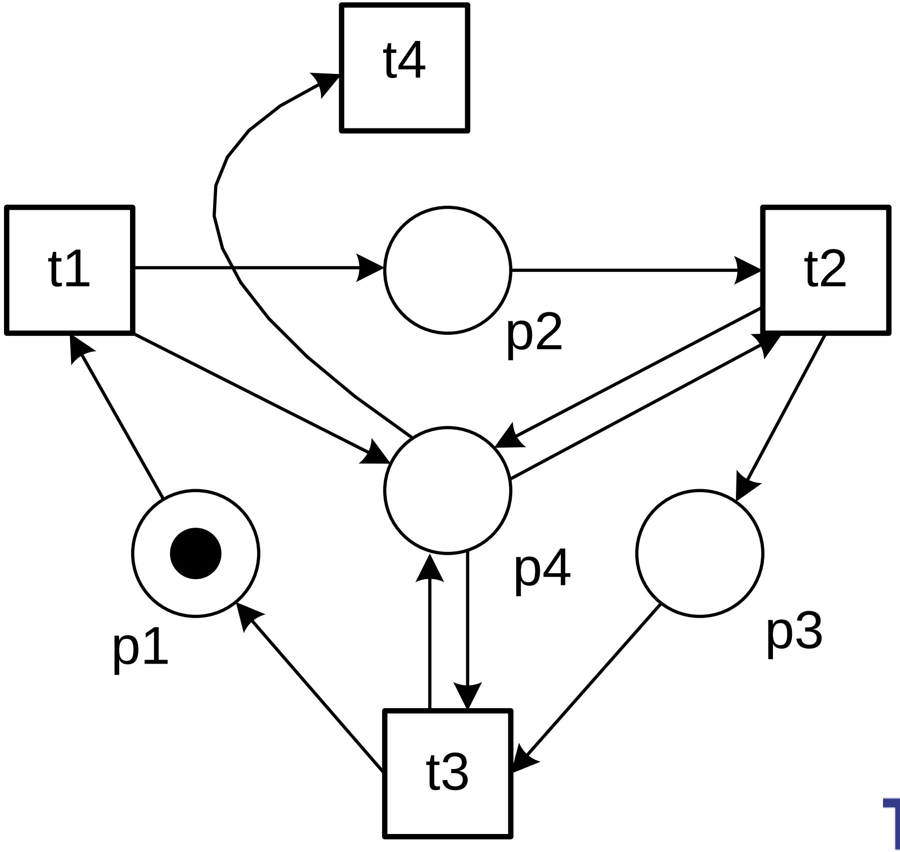
Understanding liveness



A transition *t* is live if from any reachable marking it is possible to reach a marking that enables *t*.
A Petri net is live if all transitions are live.

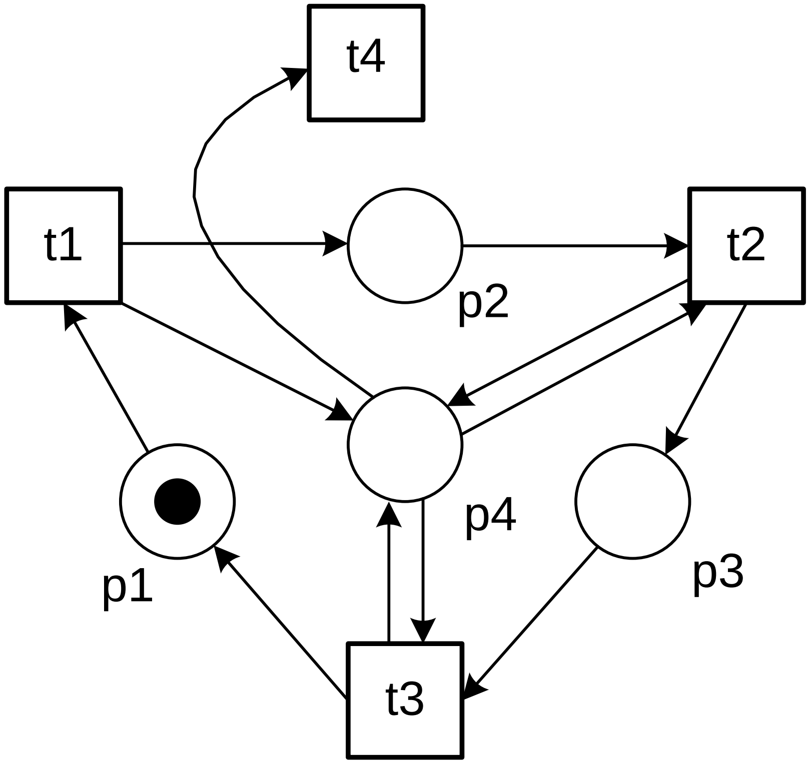
Question

- **Is the Petri net bounded?**
- **Is the Petri net safe?**
- **Is the Petri net deadlock free?**
- **Is the Petri net live?**



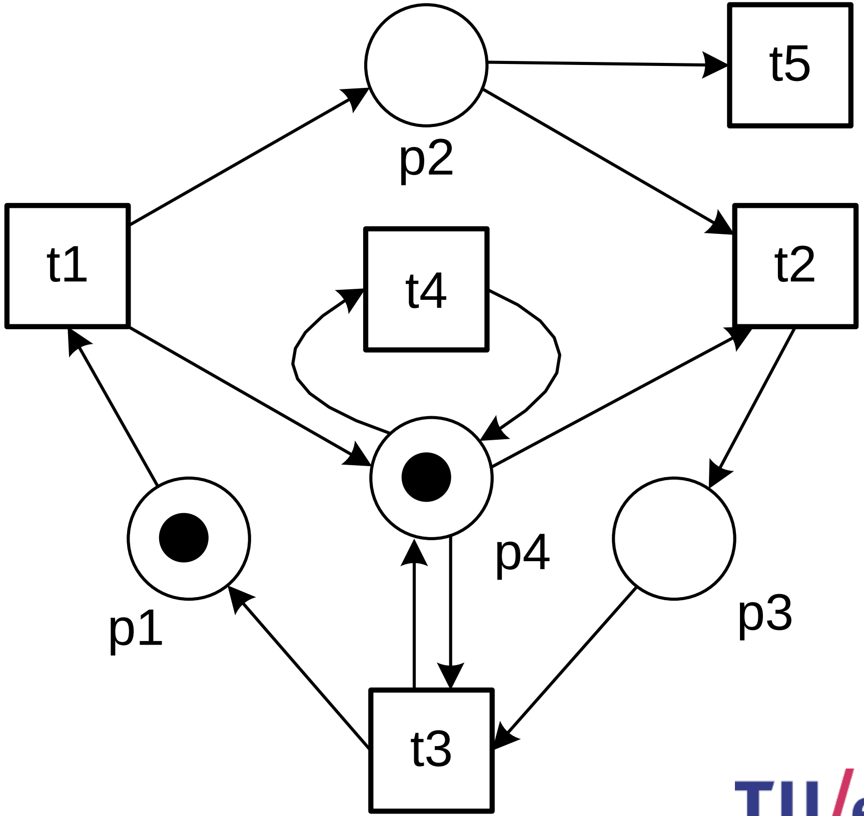
Answer

- **The Petri net is not bounded because any number of tokens can be put in *p4*.**
- **Hence, also not safe (= 1-bounded).**
- **The Petri net is not deadlock free, e.g., [p2] is reachable.**
- **Hence, also not live.**



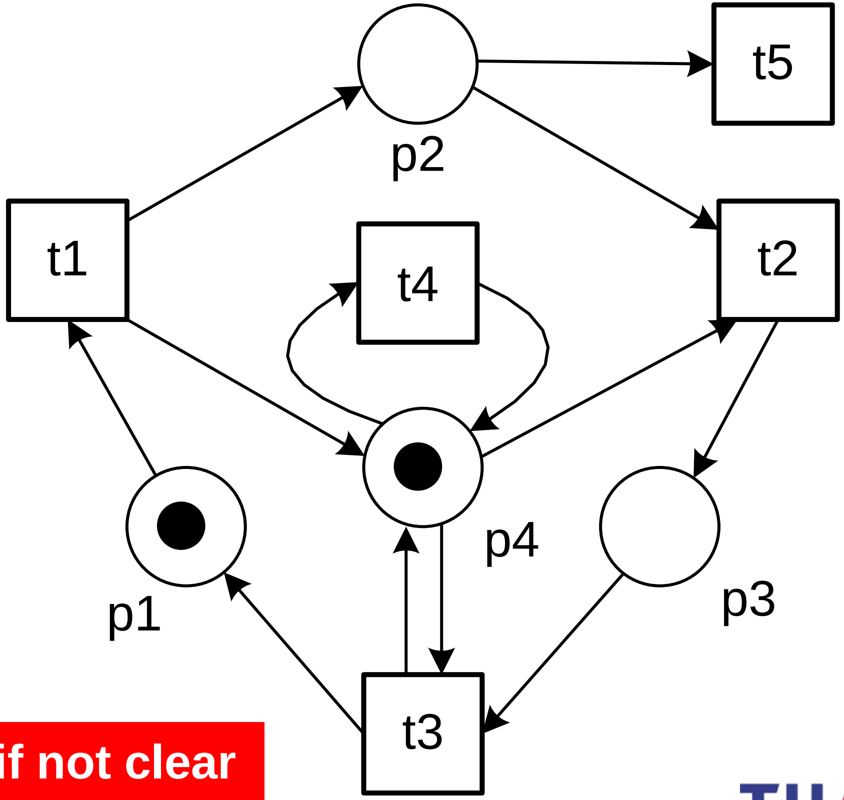
Question

- **Is the Petri net bounded?**
- **Is the Petri net safe?**
- **Is the Petri net deadlock free?**
- **Is the Petri net live?**



Answer

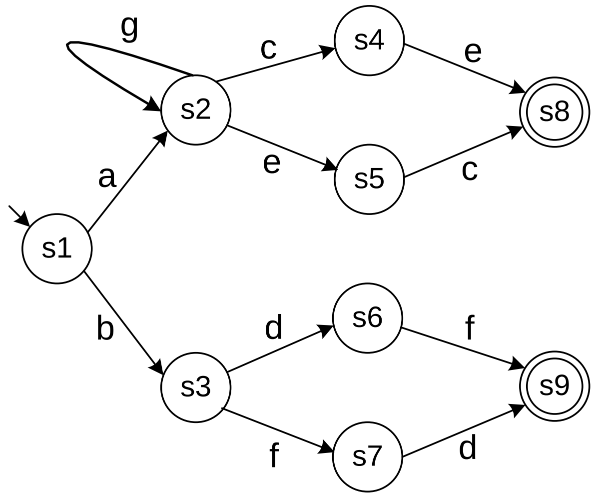
- The Petri net is **bounded**, but **not safe** (*p4* is 2-bounded).
- The Petri net is **deadlock-free**, but **not live** (after firing *t5* only *t4* can fire).



draw reachability graph if not clear

Transition systems

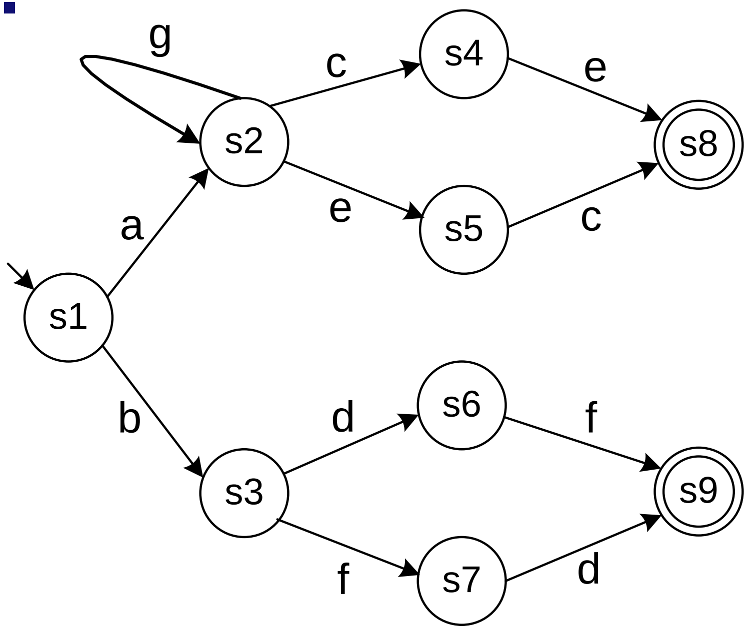
- A reachability graph is a special kind of transition system.
- Firing sequences correspond to paths in the transition system.
- A transition system is composed of states and transitions.
- There may be 1 or more initial states and 0 or more final states (more general than Petri nets).



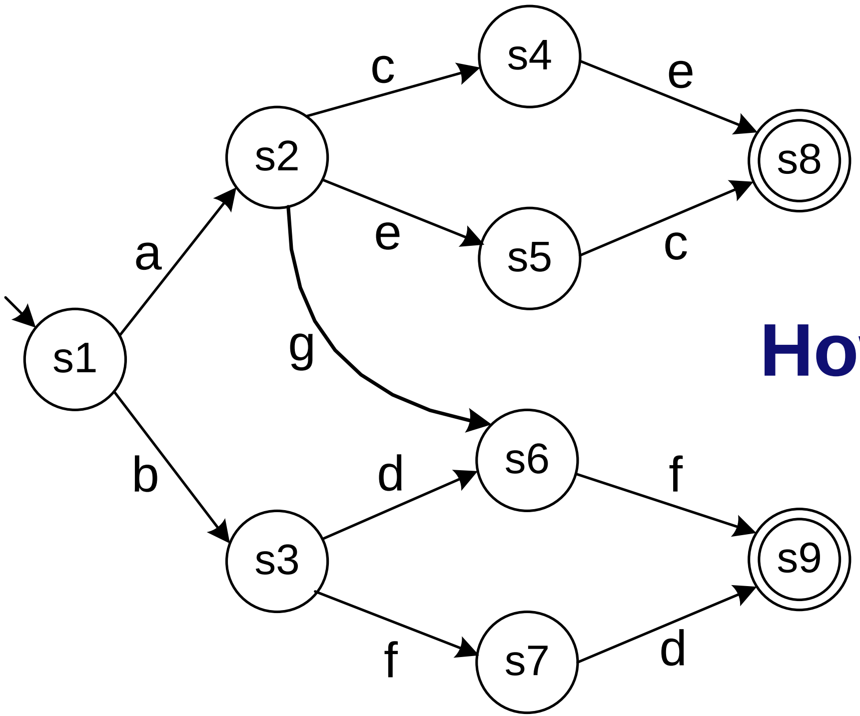
One initial state: *s1*
Two final states: *s8* and *s9*

Complete traces

- ⟨a,g,g,c,e⟩ is a **complete** trace.
- ⟨b,d,f⟩ is a **complete** trace.
- ⟨a,g,g,g,g⟩ is an **incomplete** trace.
- ⟨b,f⟩ is an **incomplete** trace.
- The transition system has **infinitely** many (in)complete traces.



Question

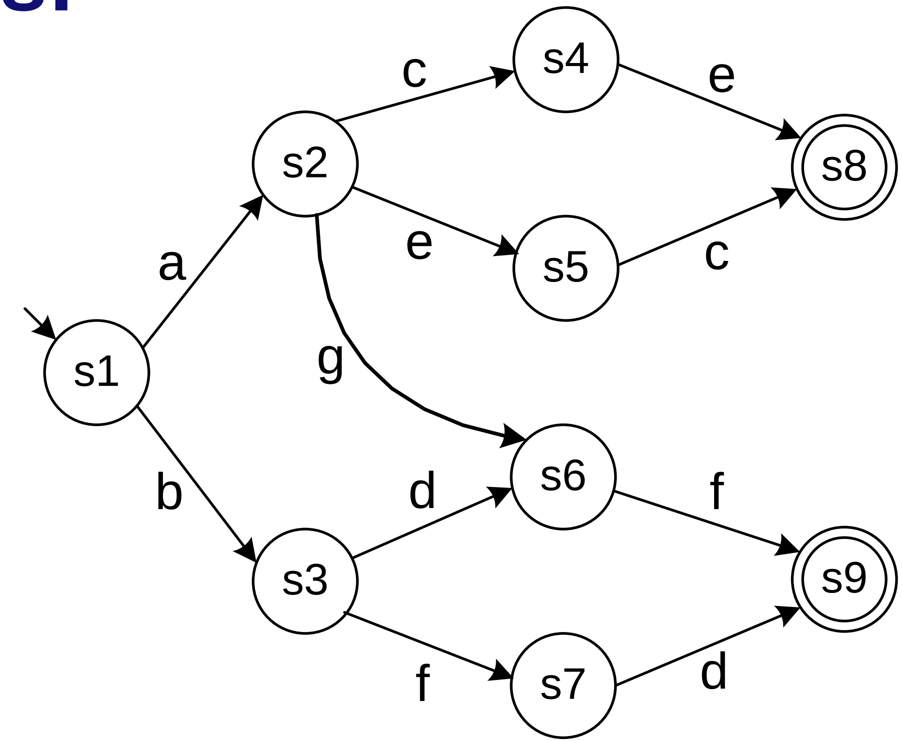


How many complete traces?

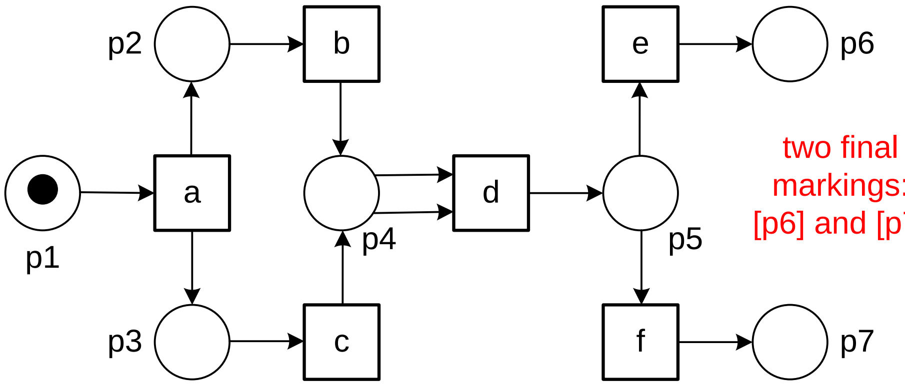
Answer

Five complete traces:

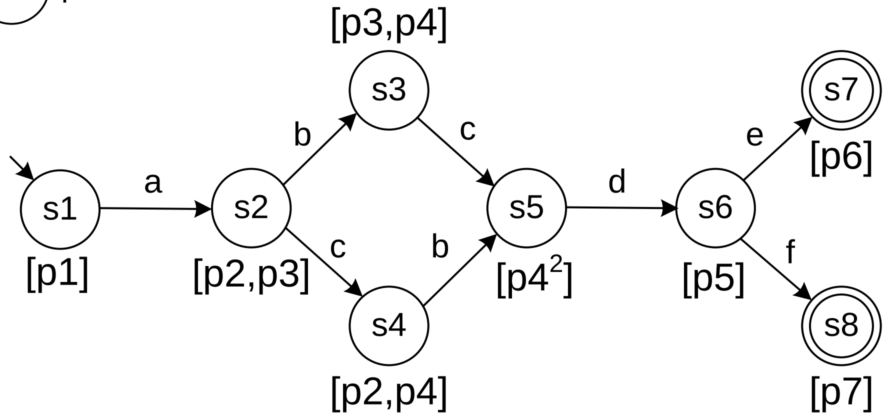
- ⟨a,c,e⟩
- ⟨a,e,c⟩
- ⟨a,g,f⟩
- ⟨b,d,f⟩
- ⟨b,f,d⟩



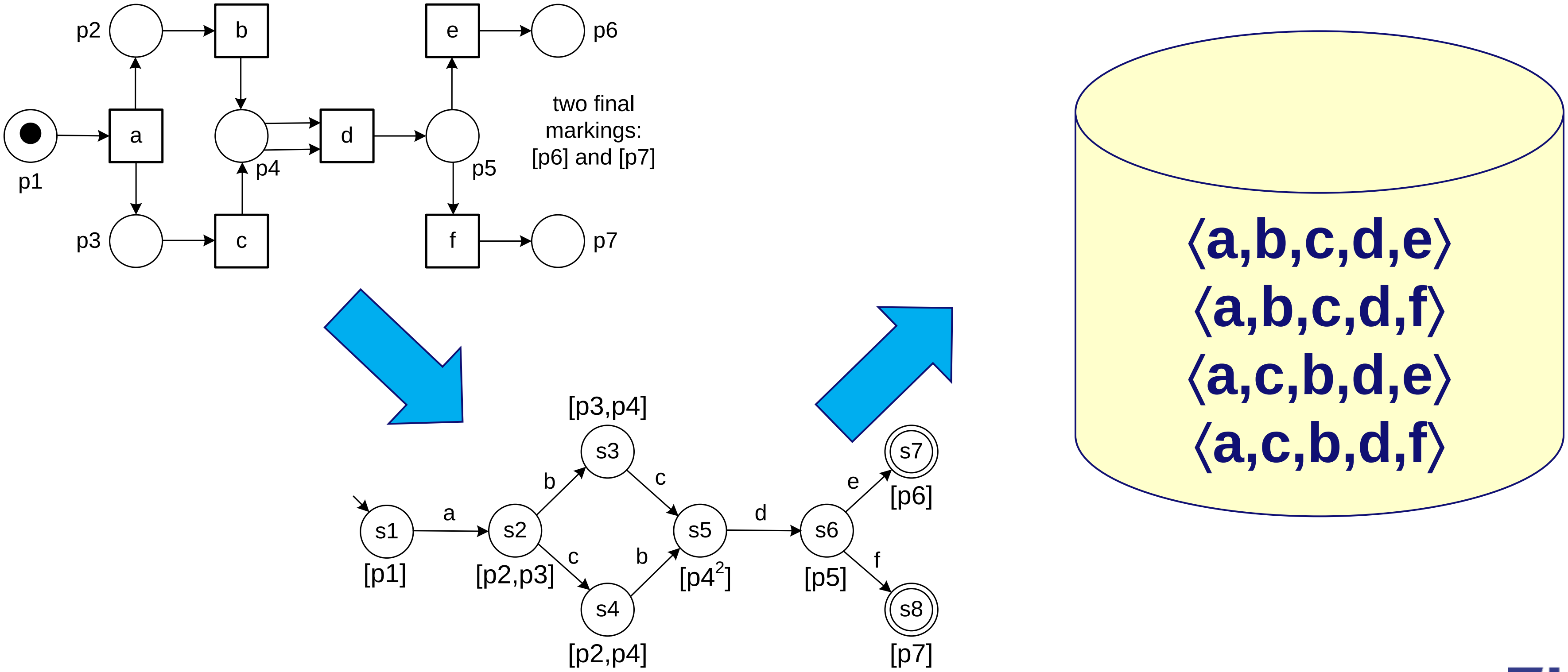
A Petri net may also have a designated set of final markings



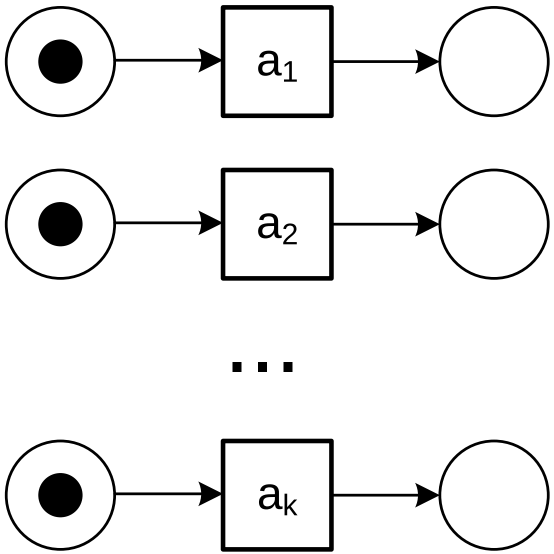
two final markings: [p6] and [p7]



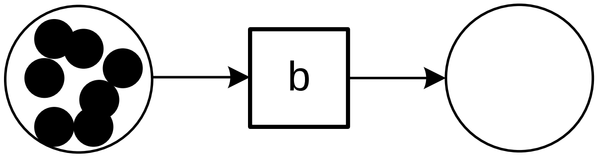
Play-out


p1
p2
p3
p4
p5
p6
p7
a
b
c
d
e
f
two final markings: [p6] and [p7]
s1
s2
s3
s4
s5
s6
s7
s8
[p1]
[p2,p3]
[p3,p4]
[p2,p4]
[p4^2]
[p5]
[p6]
[p7]
⟨a,b,c,d,e⟩
⟨a,b,c,d,f⟩
⟨a,c,b,d,e⟩
⟨a,c,b,d,f⟩

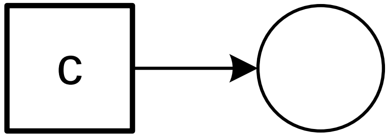
Transition system may be exponential in size of Petri net (or even infinite)



2^k states



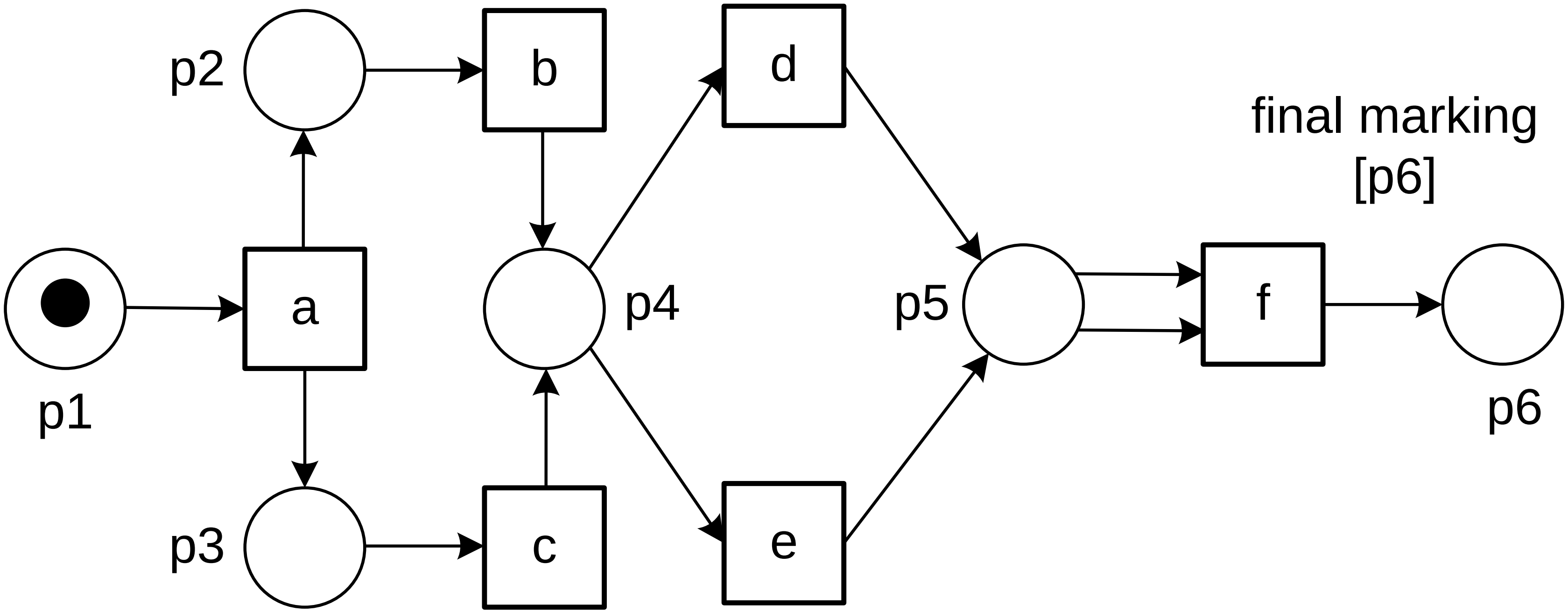
(k+1) states



infinitely many states

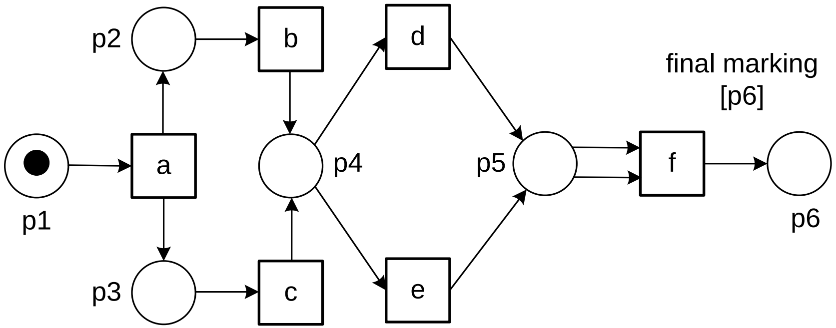
Play-out model:

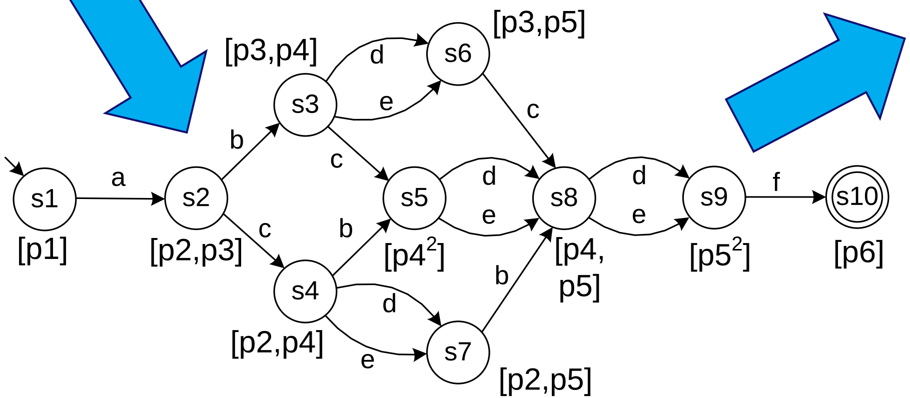
Give the transition system and all complete traces that are possible



TU/e

Resulting transition system and set of complete traces

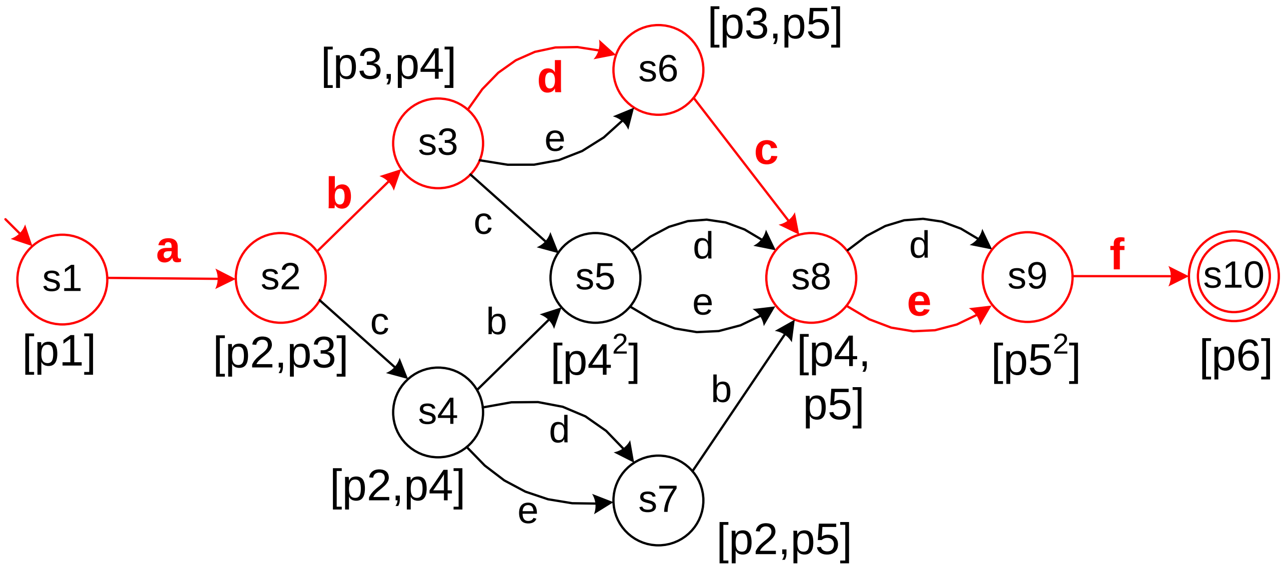




16 possible traces

⟨a,b,c,d,d,f⟩
⟨a,b,c,d,e,f⟩
⟨a,b,c,e,d,f⟩
⟨a,b,c,e,e,f⟩
⟨a,c,b,d,d,f⟩
⟨a,c,b,d,e,f⟩
⟨a,c,b,e,d,f⟩
⟨a,c,b,e,e,f⟩
⟨a,b,d,c,d,f⟩
⟨a,b,d,c,e,f⟩
⟨a,b,e,c,d,f⟩
⟨a,b,e,c,e,f⟩
⟨a,c,d,b,d,f⟩
⟨a,c,d,b,e,f⟩
⟨a,c,e,b,d,f⟩
⟨a,c,e,b,e,f⟩

Example path



⟨a,b,c,d,d,f⟩
⟨a,b,c,d,e,f⟩
⟨a,b,c,e,d,f⟩
⟨a,b,c,e,e,f⟩
⟨a,c,b,d,d,f⟩
⟨a,c,b,d,e,f⟩
⟨a,c,b,e,d,f⟩
⟨a,c,b,e,e,f⟩
⟨a,b,d,c,d,f⟩
⟨a,b,d,c,e,f⟩
⟨a,b,e,c,d,f⟩
⟨a,b,e,c,e,f⟩
⟨a,c,d,b,d,f⟩
⟨a,c,d,b,e,f⟩
⟨a,c,e,b,d,f⟩
⟨a,c,e,b,e,f⟩

Part I: Preliminaries

Chapter 1
Introduction

Chapter 2
Process Modeling and Analysis

Chapter 3
Data Mining

Part II: From Event Logs to Pr Models

Chapter 4
Getting the Data

Chapter 5
Process Discovery: An Introduction

Chapter 6
Advanced Process Discovery Techniques

Part III: Beyond Process Discovery

Chapter 7
Conformance Checking

Chapter 8
Mining Additional Perspectives

Chapter 9
Operational Support

Part IV: Putting Process Mining to Work

Chapter 10
Tool Support

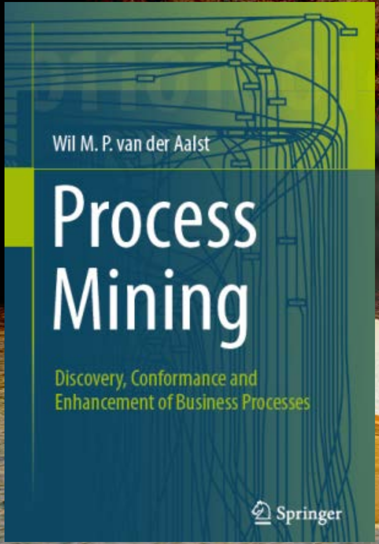
Chapter 11
Analyzing “Lasagna Processes”

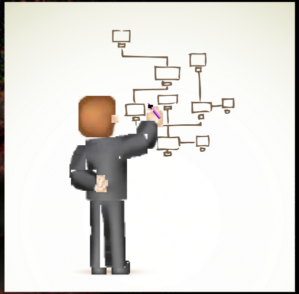
Chapter 12
Analyzing “Spaghetti Processes”

Part V: Reflection

Chapter 13
Cartography and Navigation

Chapter 14
Epilogue





TU/e