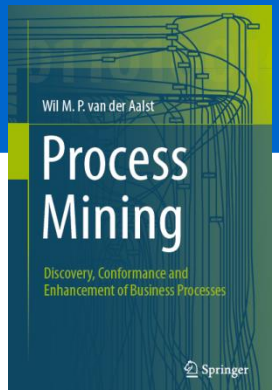
Process Mining: Data Science in Action

Petri Nets (2/2)

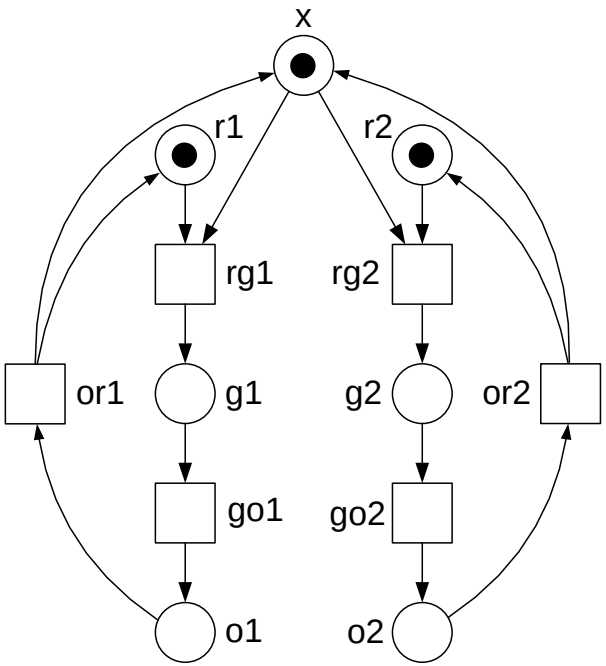


prof.dr.ir. Wil van der Aalst
www.processmining.org

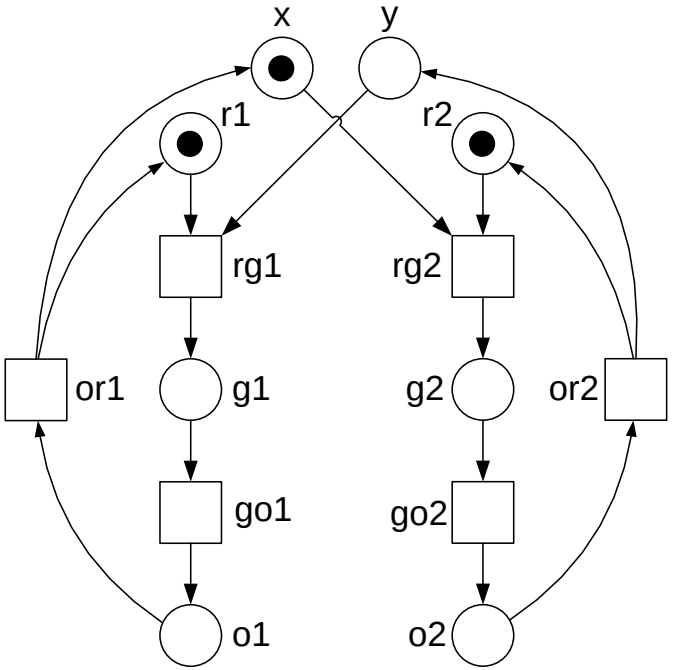
TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

Safe traffic lights



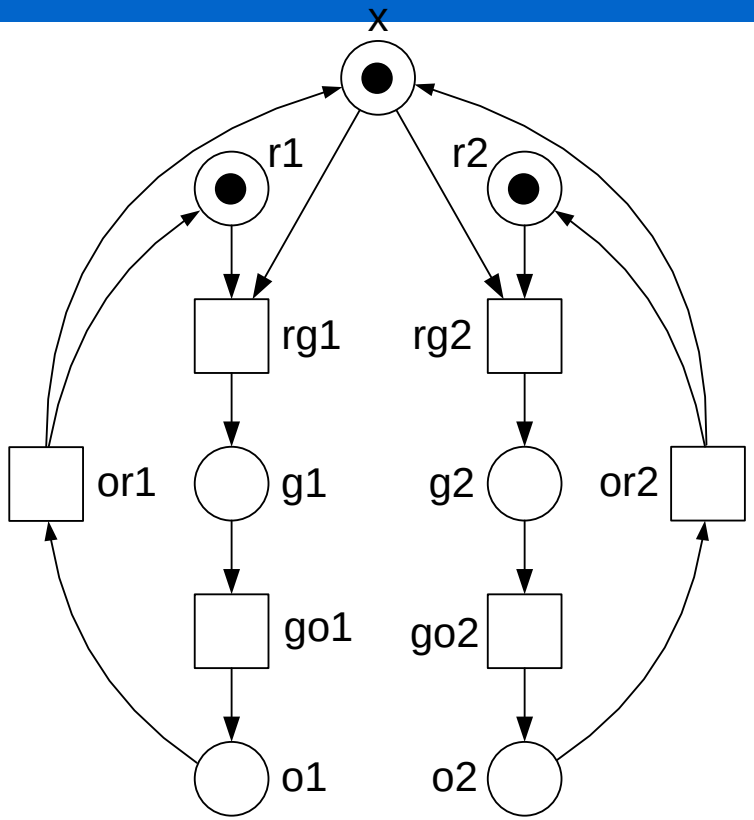
non-deterministic



alternating

TU/e

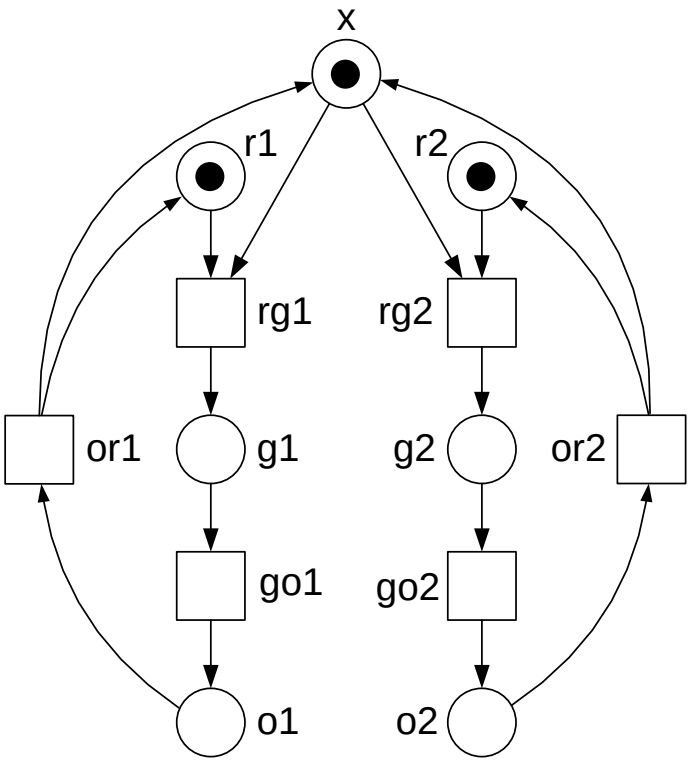
Non-deteministic traffic lights



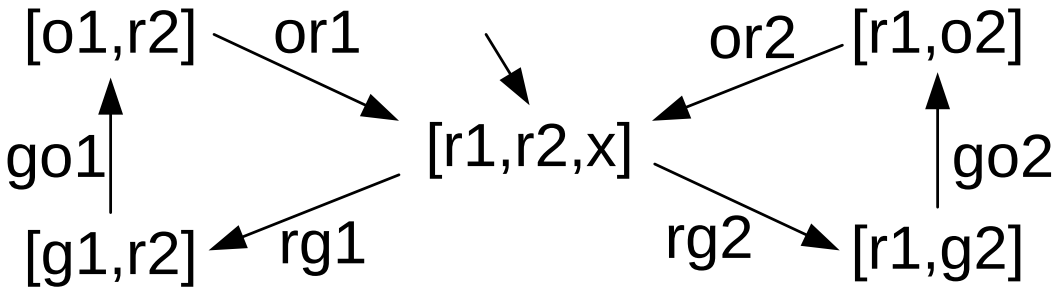
- **Initial marking:**
  [r1,r2,x]
- **Set of reachable markings:**
  { [r1,r2,x], [g1,r2], [r1,g2], [o1,r2], [r1,o2] }

TU/e

Reachability graph

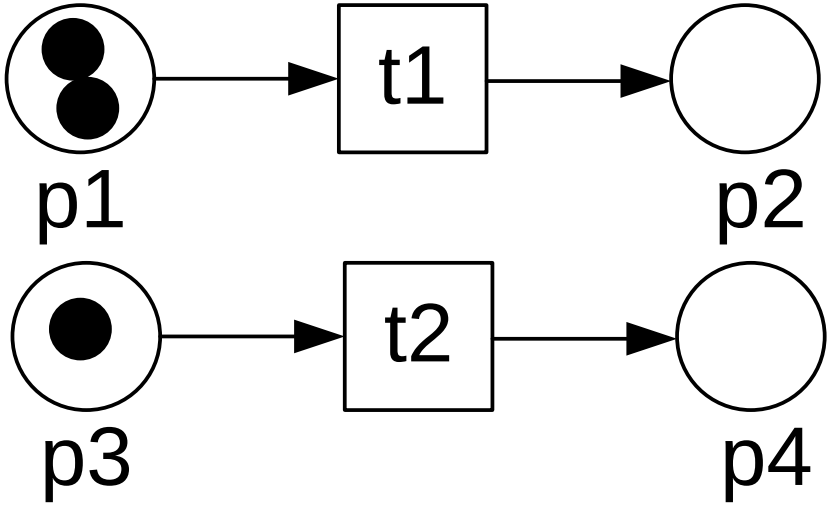


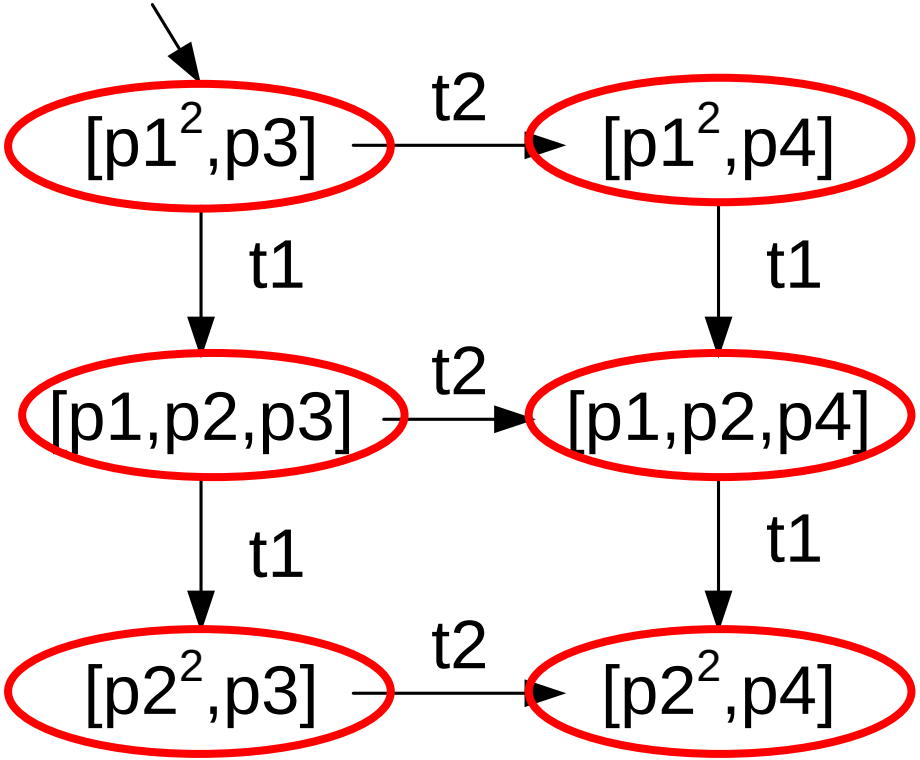
The reachability graph of a Petri net is a transition system with one initial state (initial marking) and no explicit final marking.



TU/e

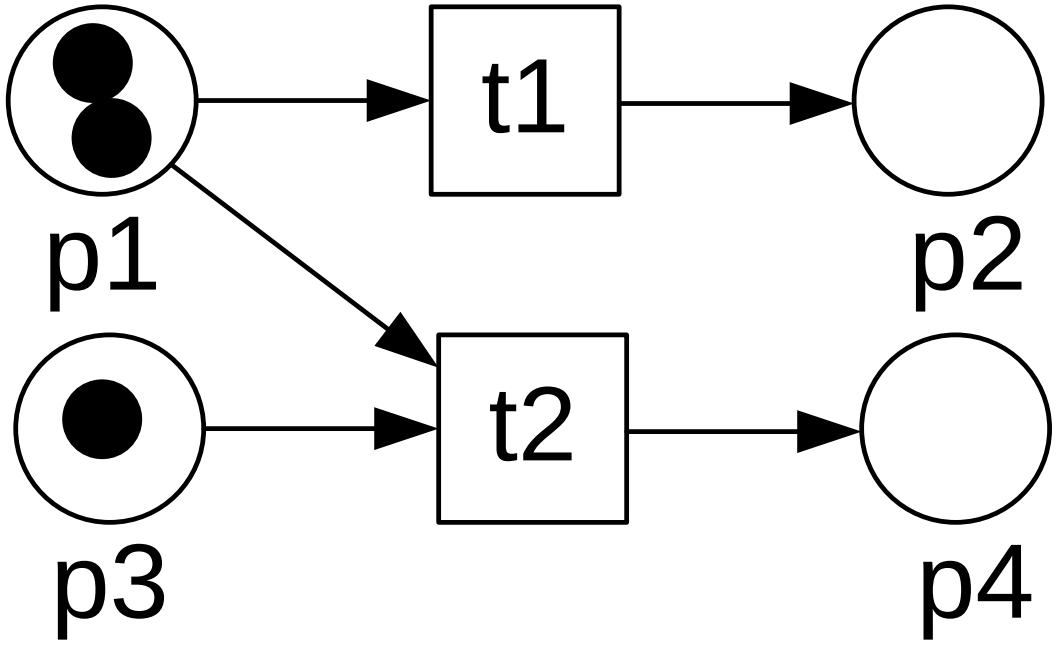
Reachability graph





TU/e

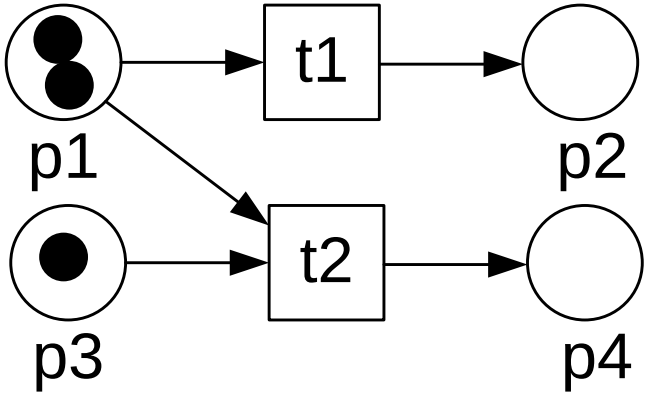
Question

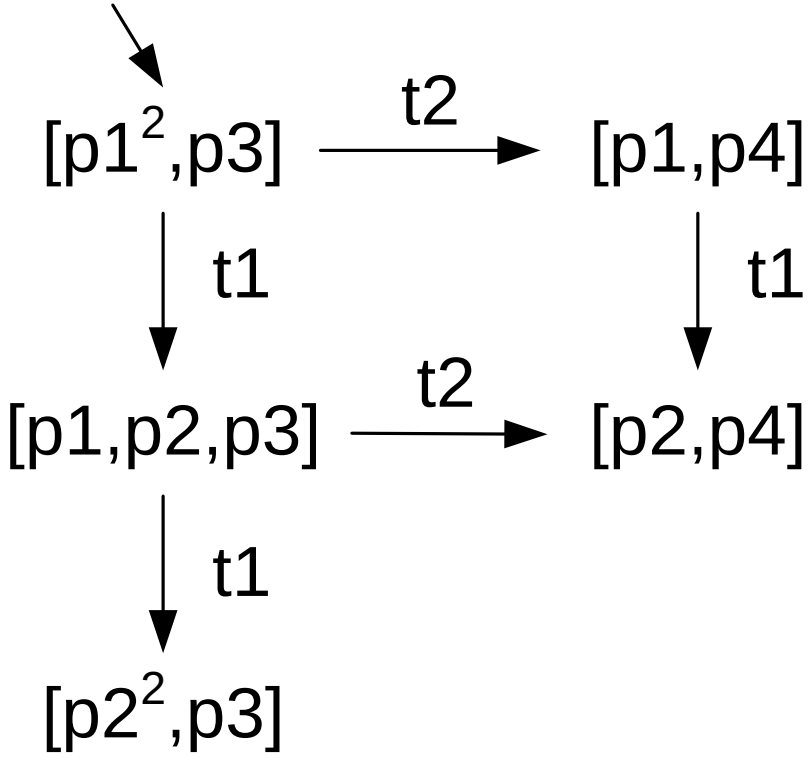


Construct the reachability graph

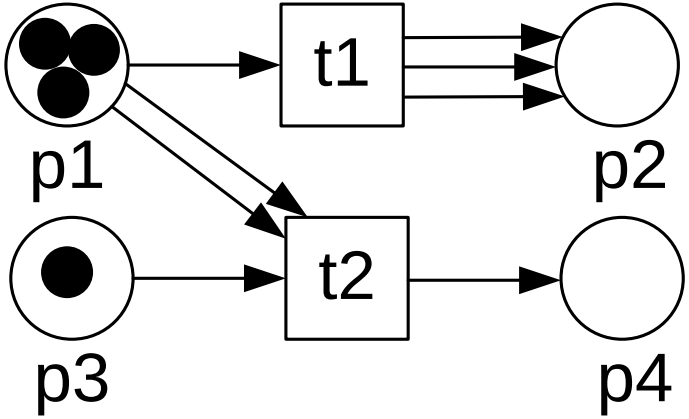
TU/e

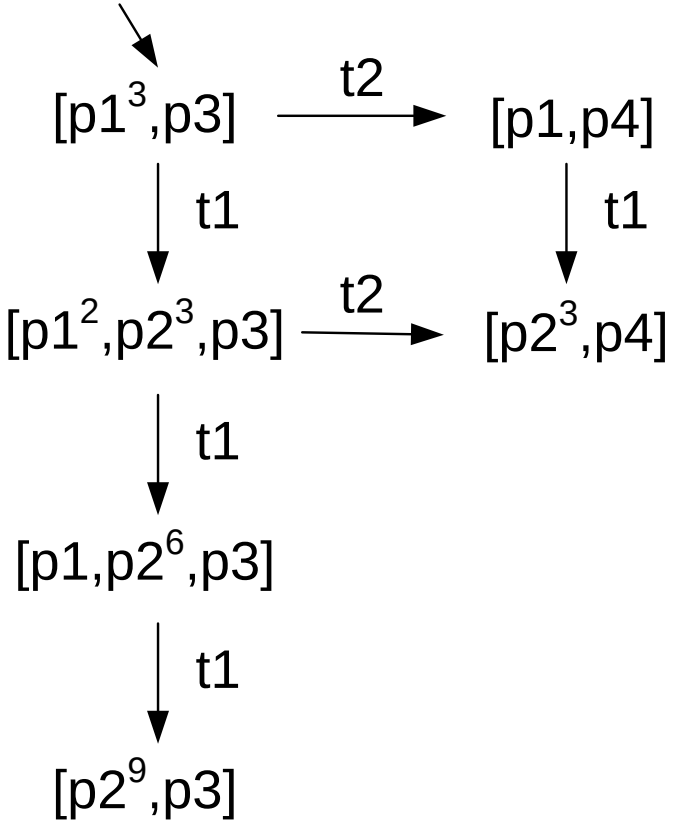
Answer



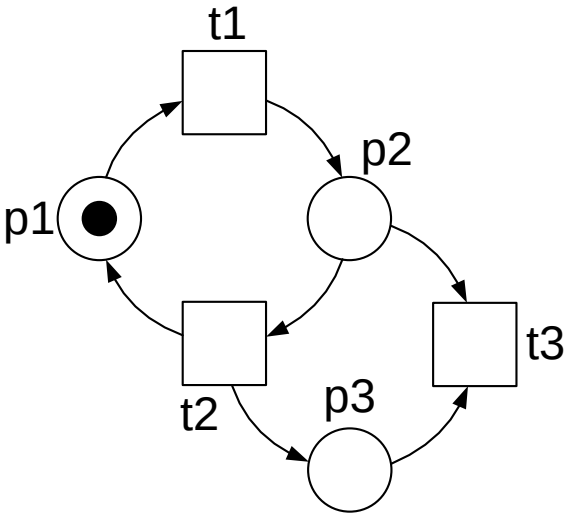


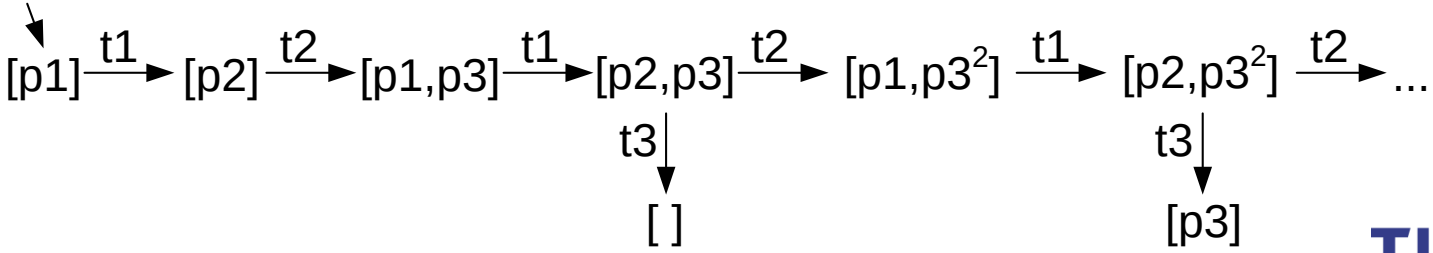
Multiple arcs connecting a place and a transition





Reachability graph does not need to be finite





Modeling can be very challenging …

practice, practice, practice, …

Question (not easy, takes 10 minutes!)

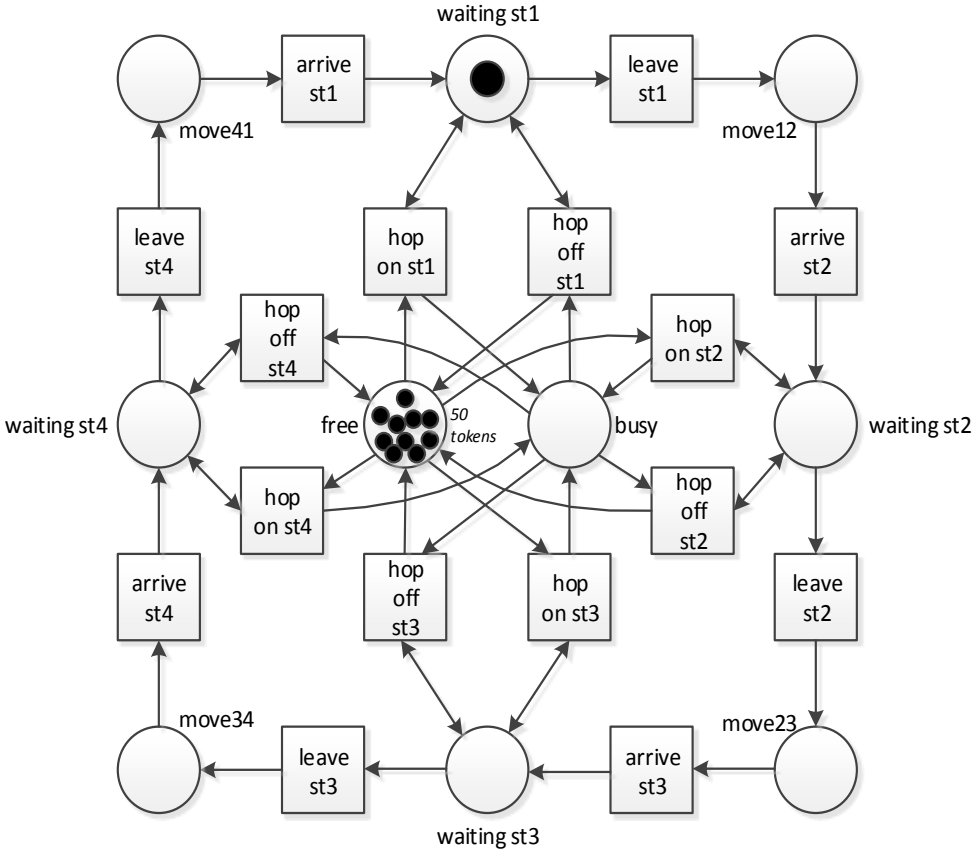
- Model a circular railway system with four stations (st1, st2, st3, and st4) and one train.
- At each station passengers may "hop on" or "hop off". This is impossible if the train is moving.
- The train has a capacity of 50 persons; if the train is full, no new passengers may hop on.
- Model the above process in terms of a Petri net.
- What is the number of reachable states?

TU/e

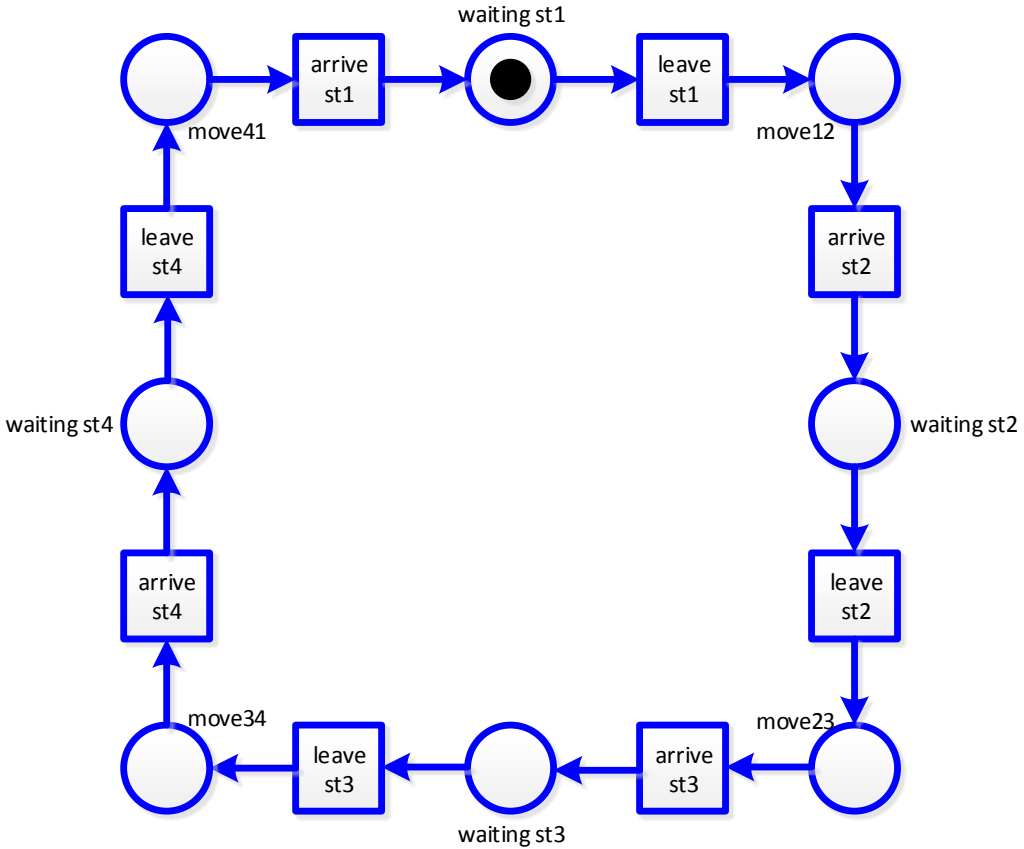
Hints

- How to describe the state of the train in terms of its location (e.g., moving from st1 to st2) and number of passengers (e.g., 36)?
- What are possible actions?
- When are they possible?

Answer (1/4)



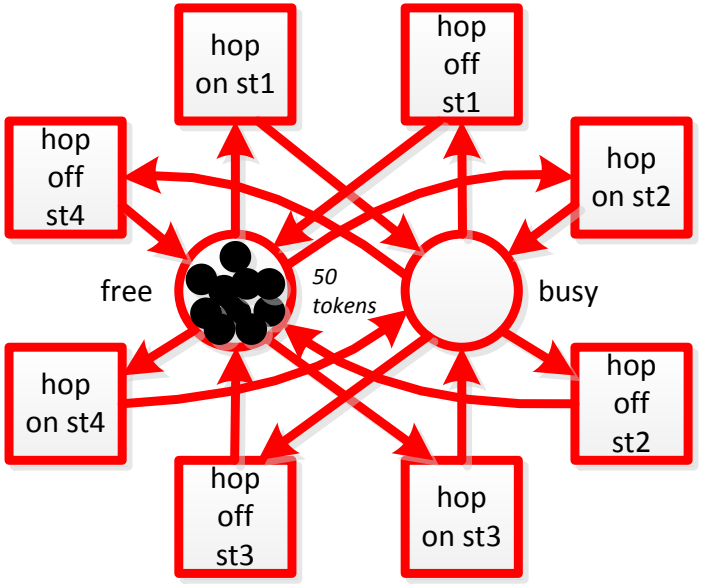
Answer (2/4)



state of the train: location

TU/e

Answer (3/4)



state of the train: passengers

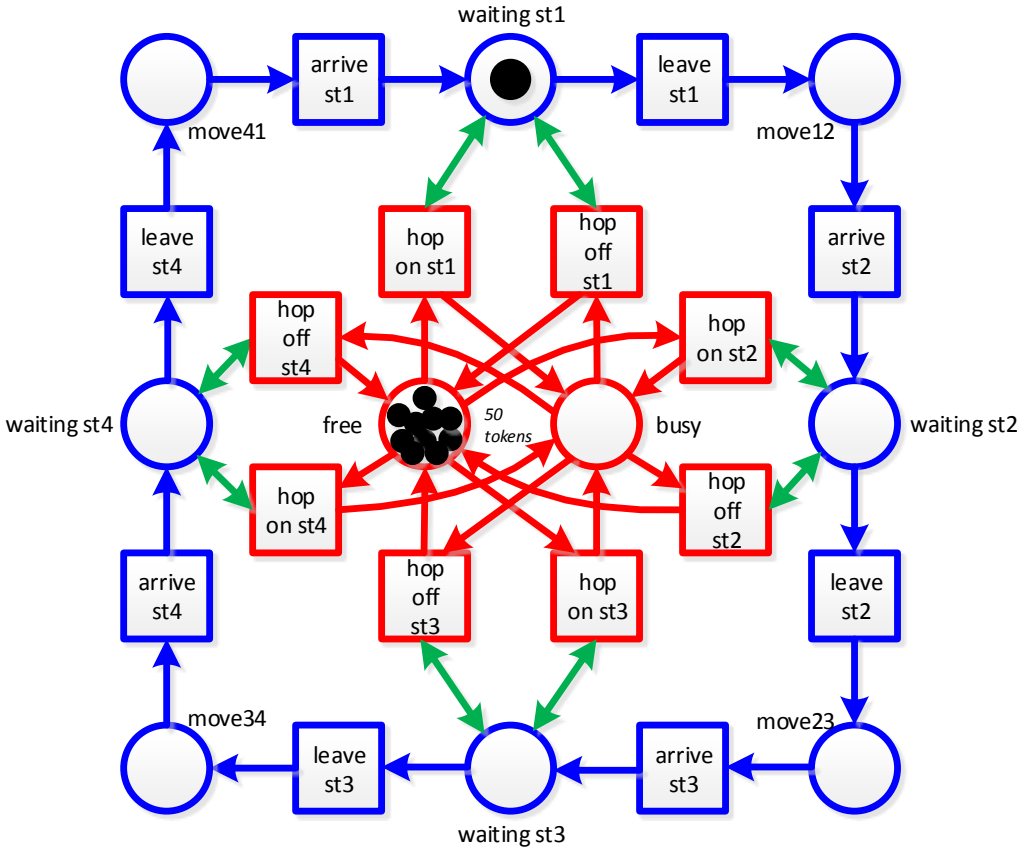
TU/e

Answer (4/4)

51 x 8 = 408 reachable markings

Reachability graph is already too large to draw manually!

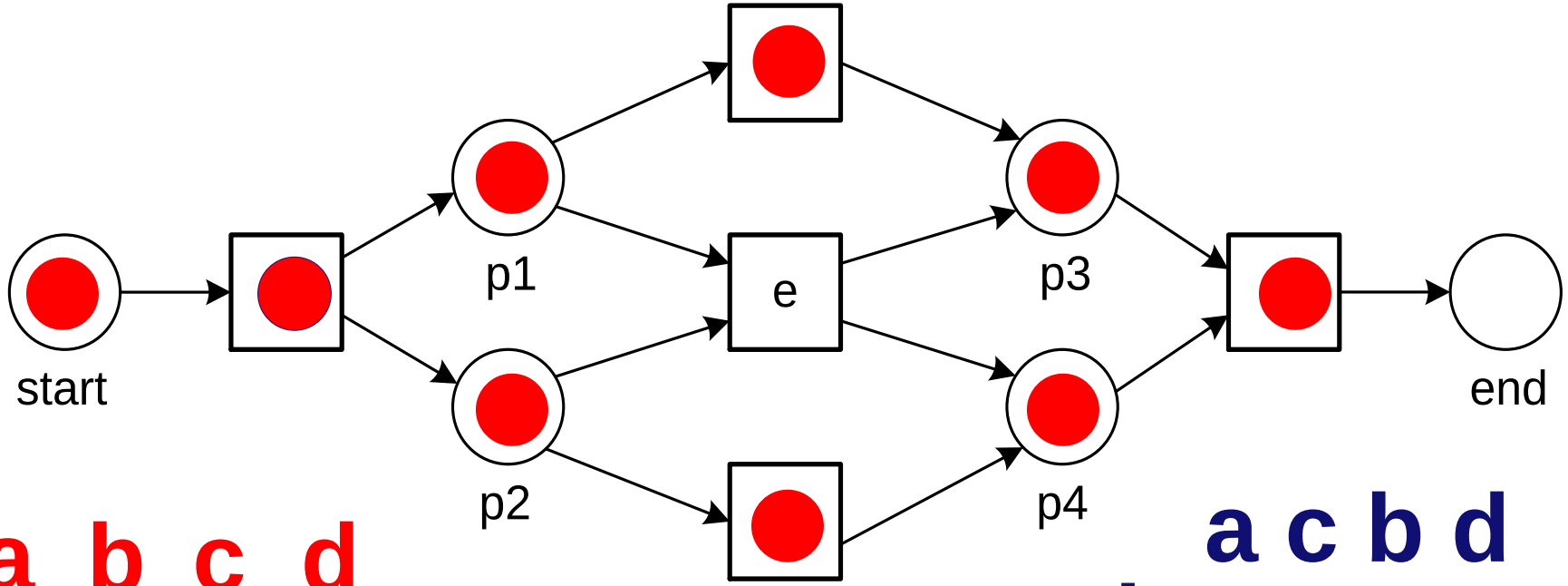
merged model



TU/e

"Token game" defines play-out

Play-Out (Classical use of models)



a b c d

a e d

a c b d

a c b d

a e d

a b c d

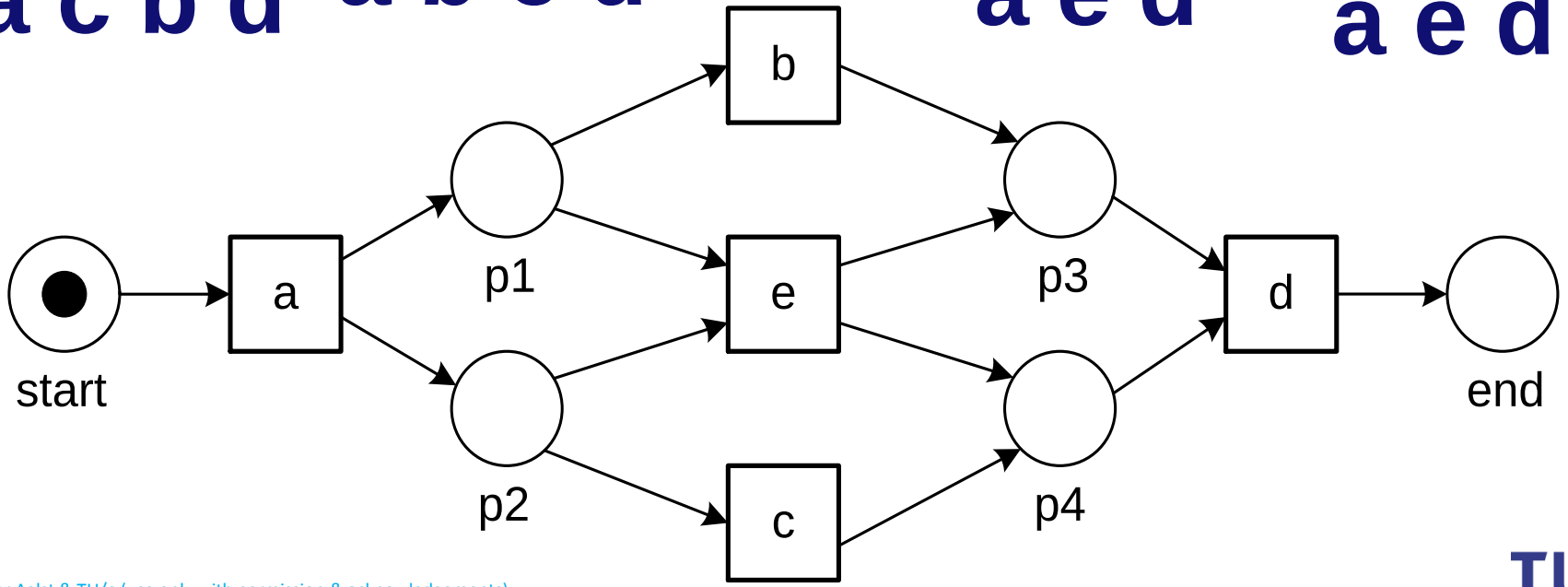
a e d

a c b d

TU/e

Play-In

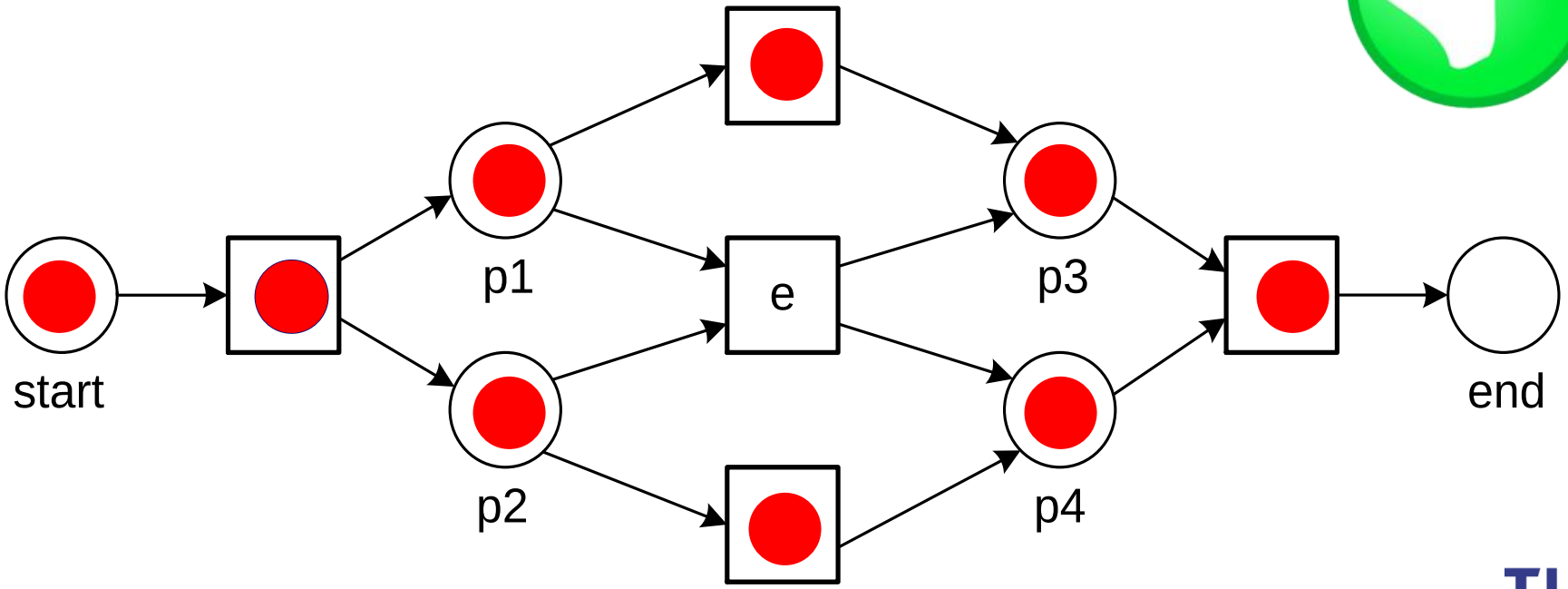
a b c d a e d a c b d a c b d a b c d a e d a e d



TU/e

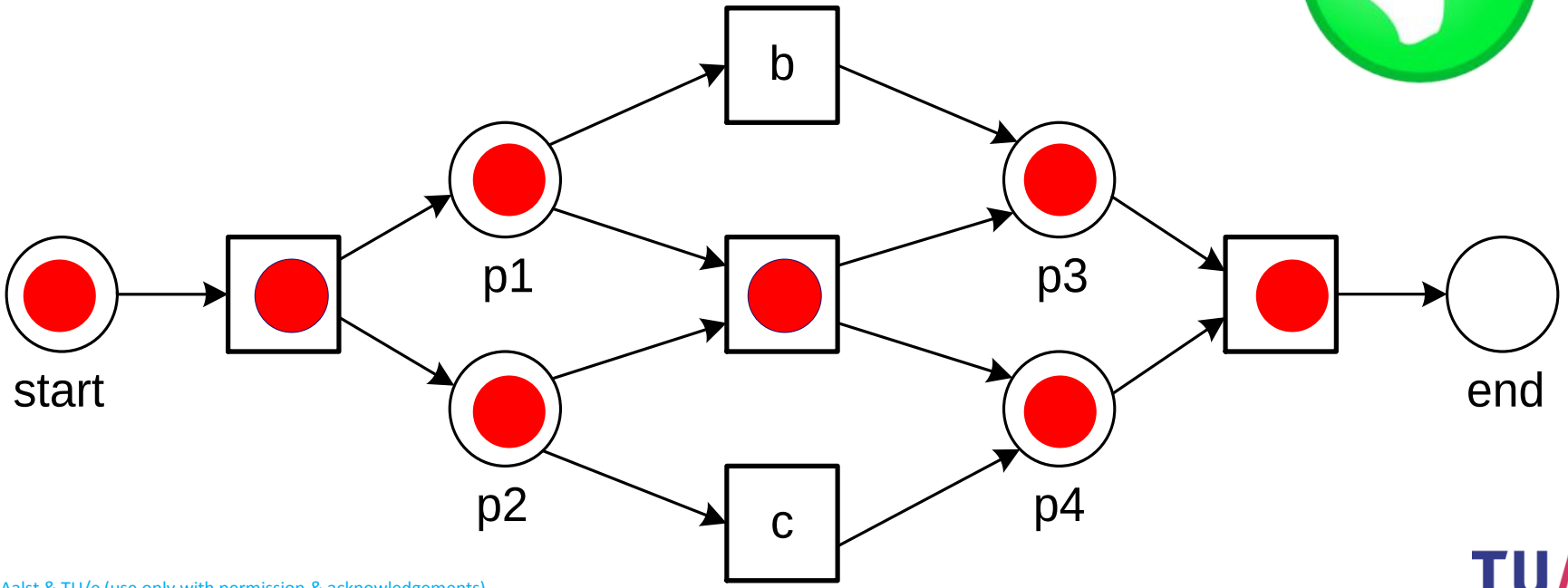
Replay

a b c d



Replay

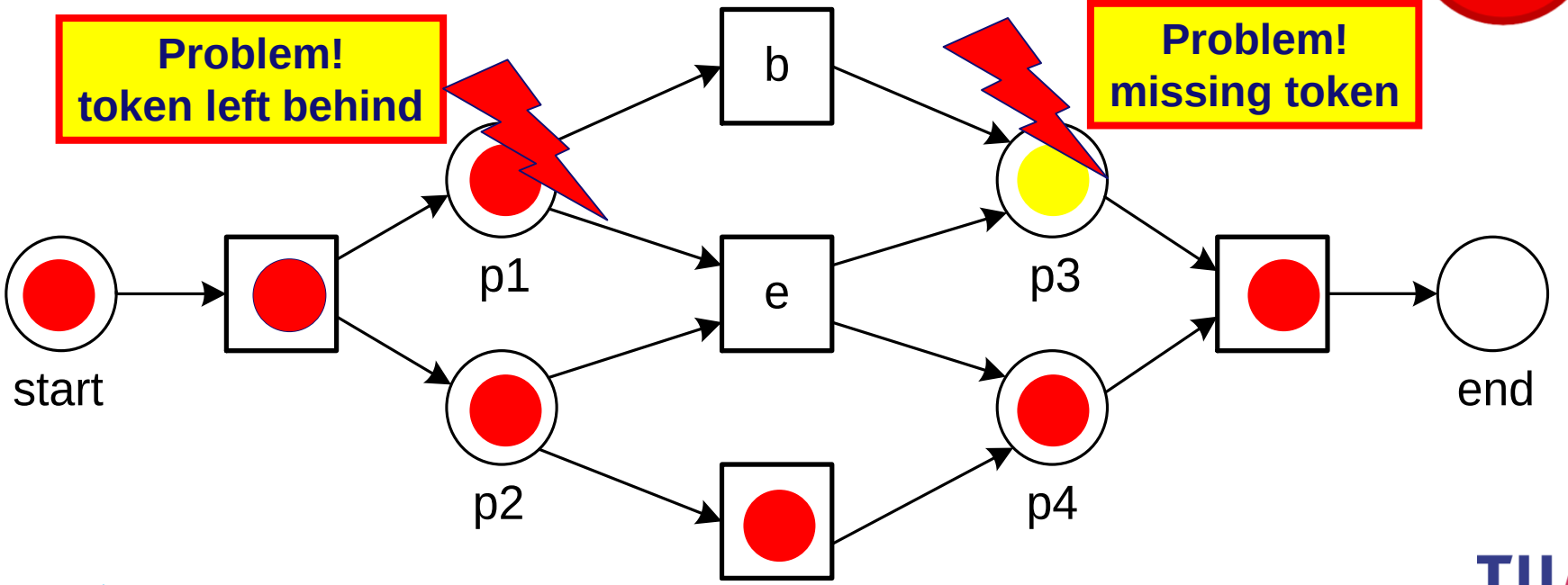
a e d



TU/e

Replay can detect problems

a c d



TU/e

Part I: Preliminaries

- Chapter 1: Introduction
- Chapter 2: Process Modeling and Analysis
- Chapter 3: Data Mining

Part II: From Event Logs to Process Models

- Chapter 4: Getting the Data
- Chapter 5: Process Discovery: An Introduction
- Chapter 6: Advanced Process Discovery Techniques

Part III: Beyond Process Discovery

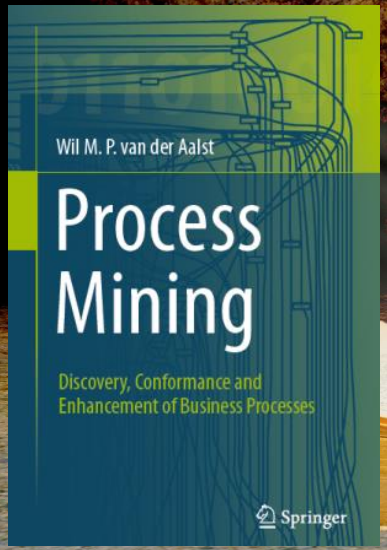
- Chapter 7: Conformance Checking
- Chapter 8: Mining Additional Perspectives
- Chapter 9: Operational Support

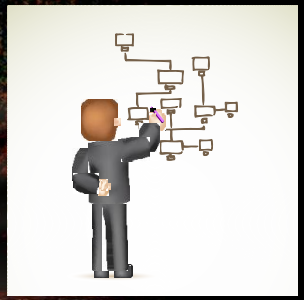
Part IV: Putting Process Mining to Work

- 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