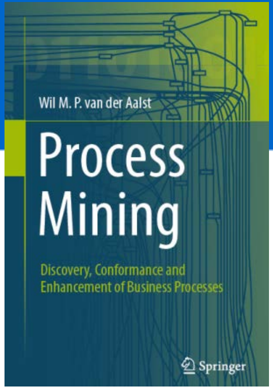
Process Mining: Data Science in Action

Two-Phase Process Discovery And Its Limitations

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


Wil M. P. van der Aalst
Process Mining
Discovery, Conformance and Enhancement of Business Processes
Springer

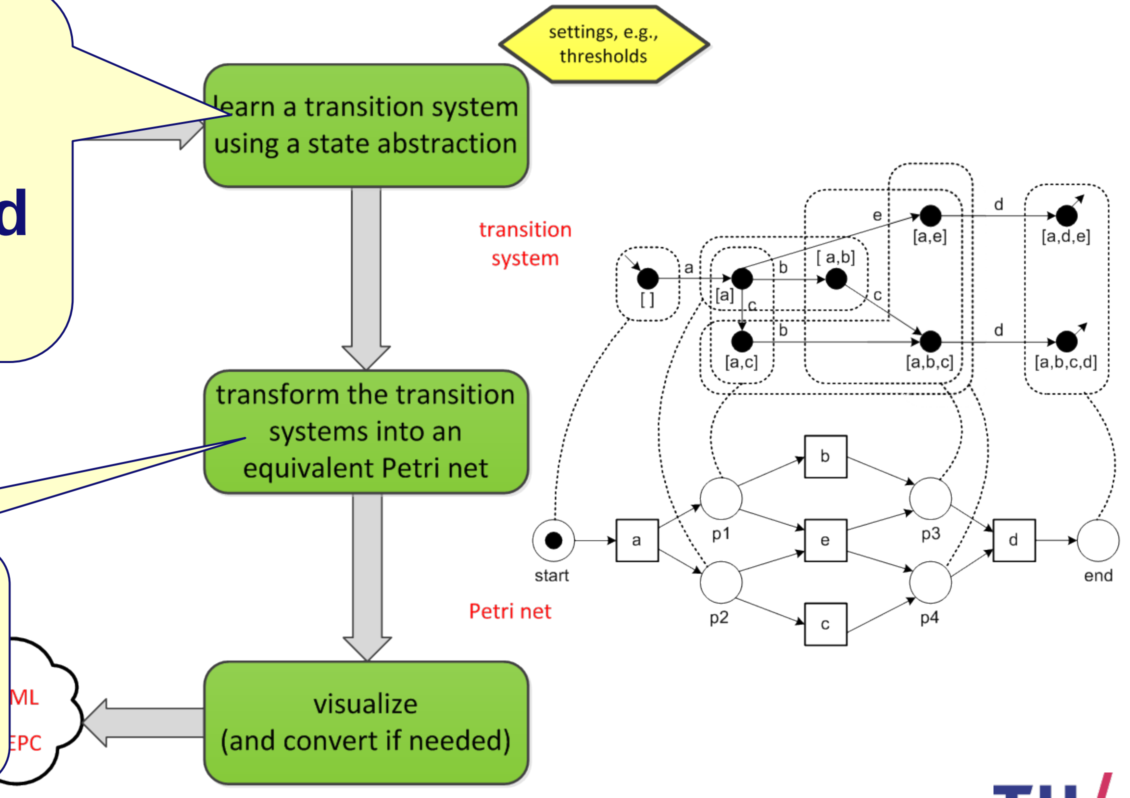
Technische Universiteit Eindhoven University of Technology

Where innovation starts

From event logs to transition systems to process models

different abstractions possible to mediate between overfitting and underfitting

discovery of concurrency


settings, e.g., thresholds
learn a transition system using a state abstraction
transform the transition systems into an equivalent Petri net
visualize (and convert if needed)
ML
EPC
transition system
Petri net
[ ]
a
[a]
b
[a,b]
c
[a,c]
e
[a,e]
d
[a,d,e]
[a,b,c]
[a,b,c,d]
start
a
p1
p2
b
e
c
p3
p4
d
end

Algorithm discussed before

1. For each transition label in the transition system, a transition is added to the Petri net.
2. The minimal non-trivial regions are computed.
3. For each minimal non-trivial region in the transition system, a place is added.
4. The corresponding arcs are generated.
5. A token is added to each place that corresponds to a region containing the initial state.

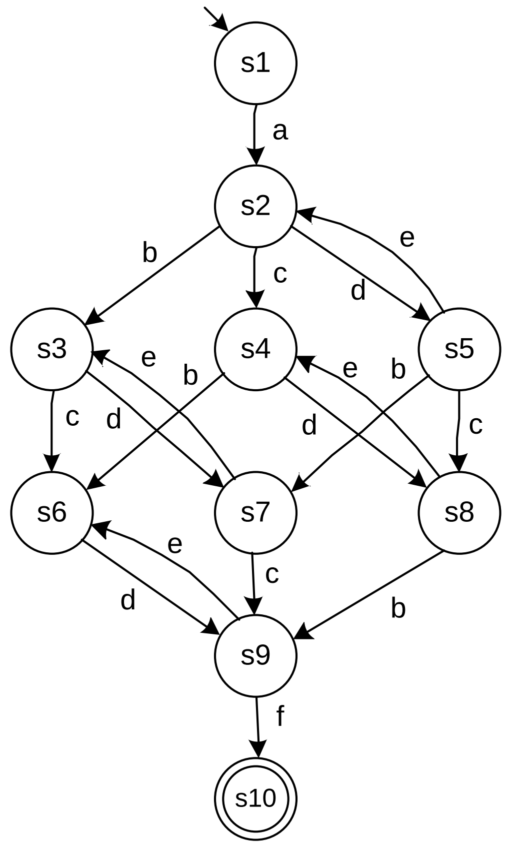
Example: event log to transition system

$\langle a,b,c,d,f\rangle^{45}$

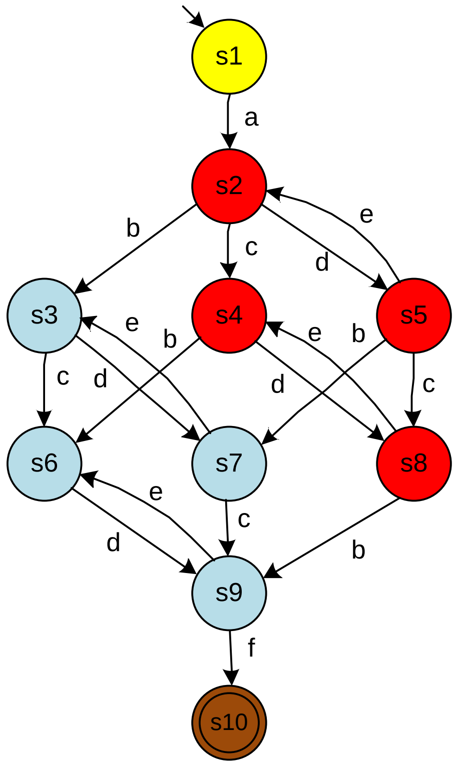
$\langle a,d,c,b,f\rangle^{42}$

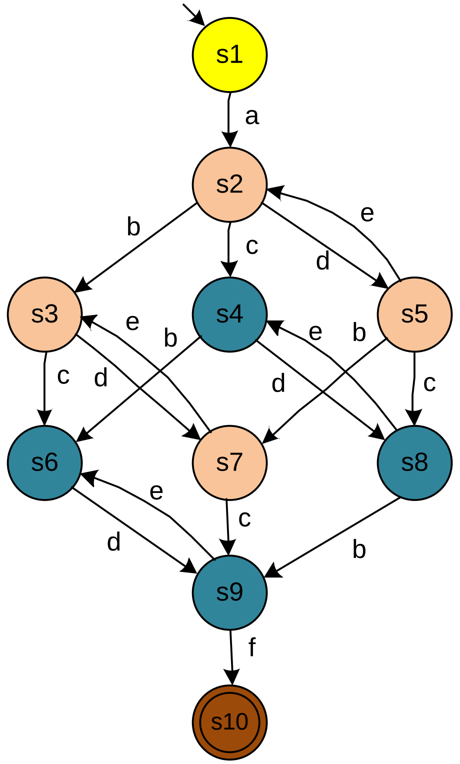
$\langle a,c,d,e,d,b,f\rangle^{38}$

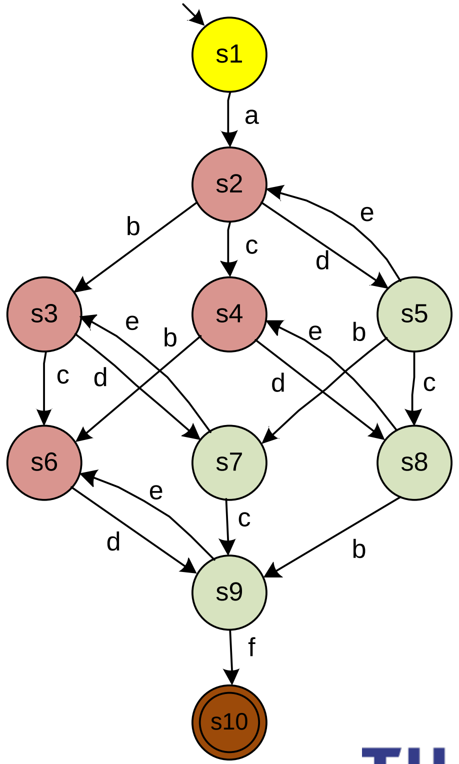
...



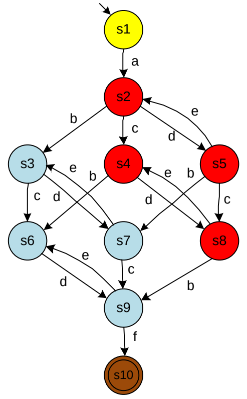
Example: Minimal non-trivial regions

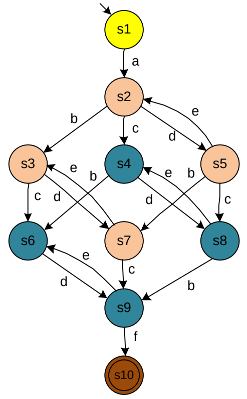


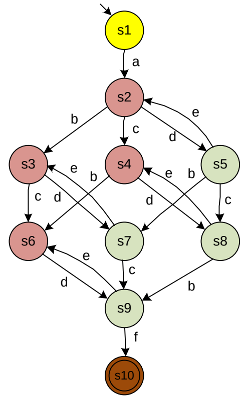




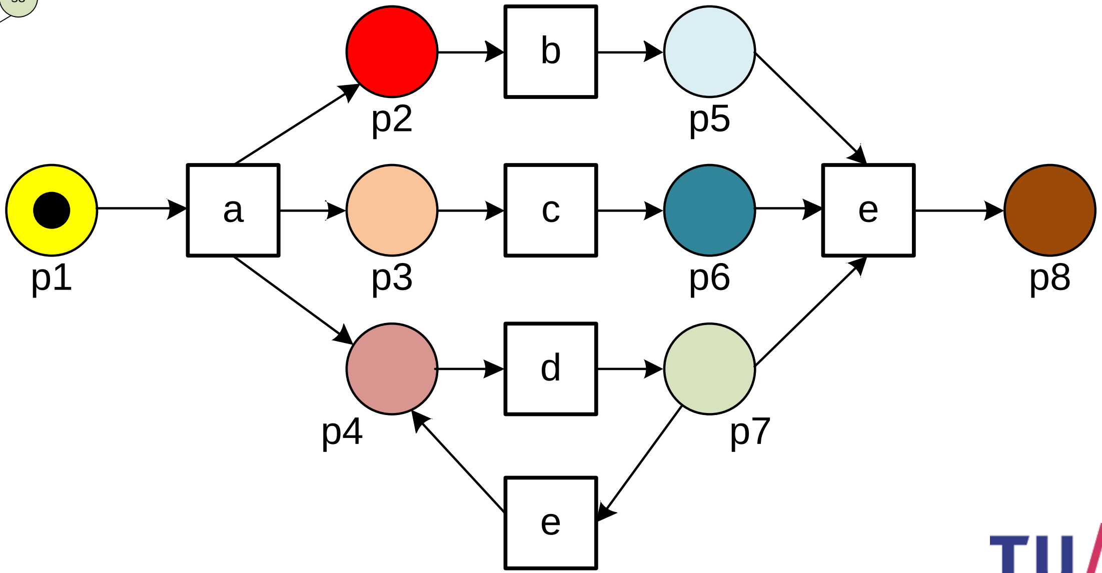
Example: Minimal regions define places







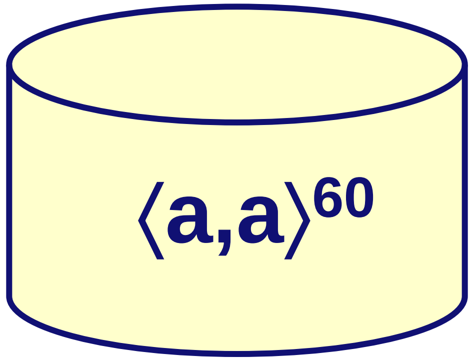
Transition system and Petri net are bisimilar.

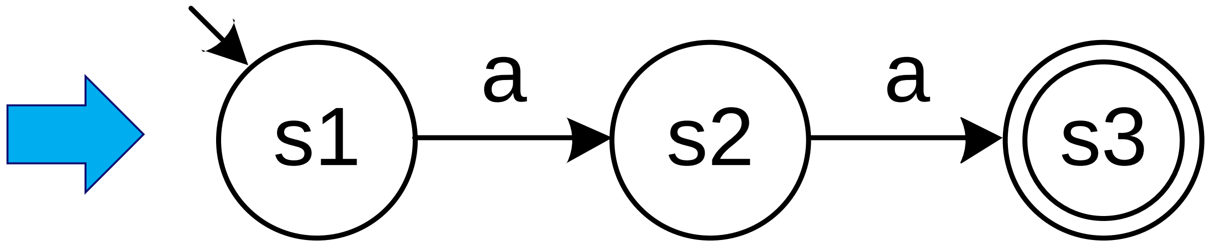


Weaknesses of approach based on state-based regions

- Inability to discover particular process constructs (can be handled through extensions of the basic algorithm).
- Inability to balance the four forces (fitness, precision, generalization, and simplicity) well.

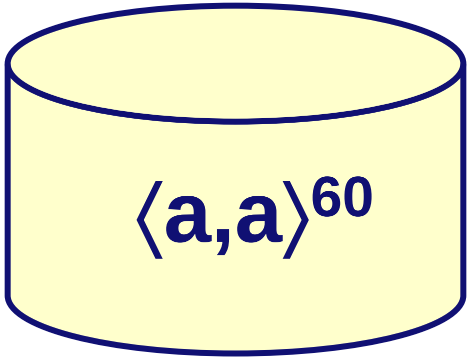
Question: Construct Petri net using regions

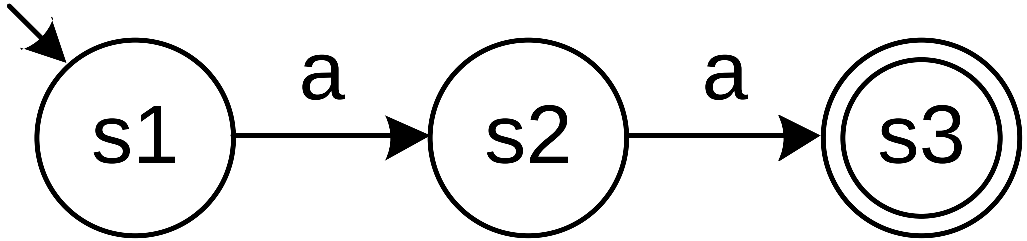




What is the Petri net constructed based on state-based regions?

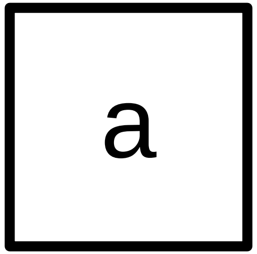
Answer: Petri net without places





Only trivial regions: ∅ and {s1,s2,s3}

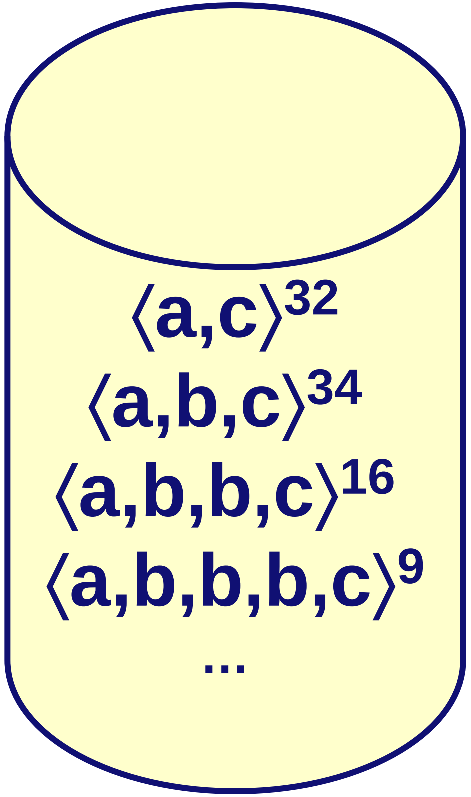
Petri net has no places:

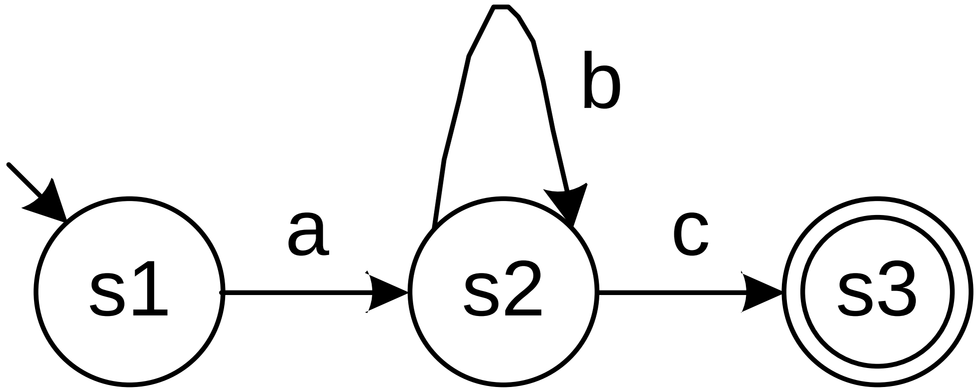


Also allows for:

- ⟨a⟩
- ⟨a,a,a⟩
- ⟨a,a,a,a,a,a,a⟩

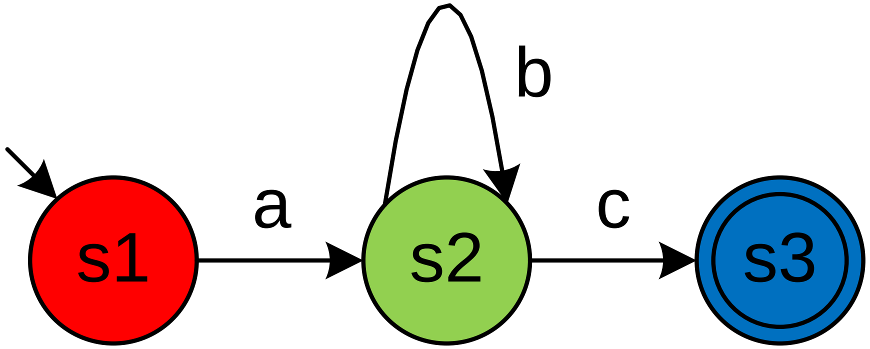
Question: Construct Petri net using regions

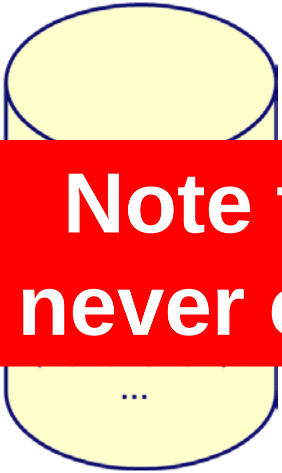




What is the Petri net constructed based on state-based regions?

Answer (1/2)





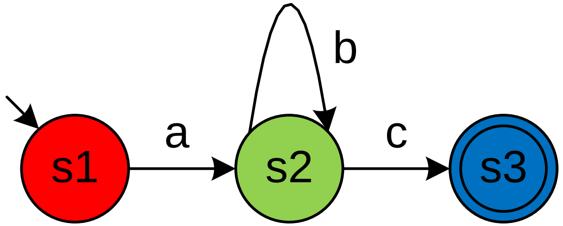
Note that b is never crossing.

Regions:

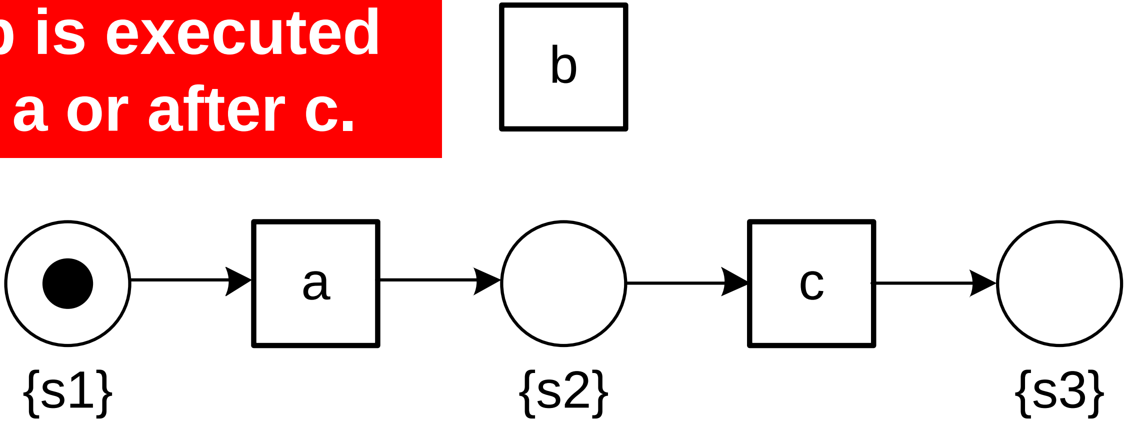
- **{s1} (a exits, b and c do not cross)**
- **{s2} (a enters, b does not cross, c exits)**
- **{s3} (a and b does not cross, c enters)**

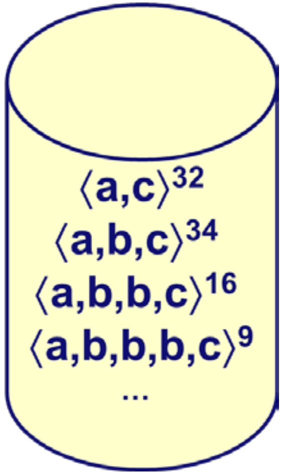
Answer (2/2)

Three regions: {s1}, {s2} , and {s3}.



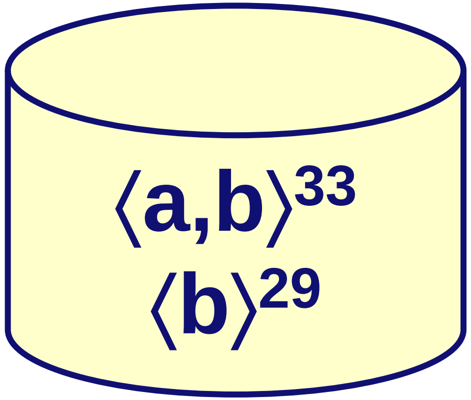
Resulting Petri net allows for traces where b is executed before a or after c.





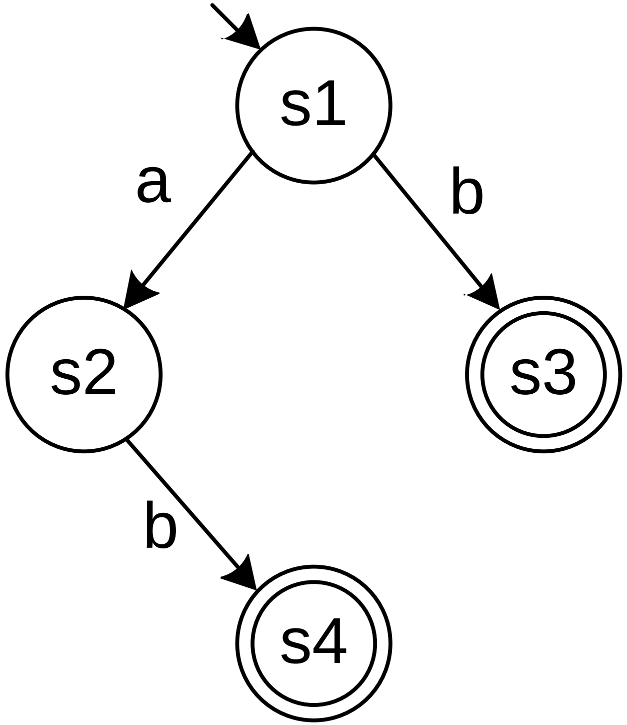
$\langle a,c\rangle^{32}$
$\langle a,b,c\rangle^{34}$
$\langle a,b,b,c\rangle^{16}$
$\langle a,b,b,b,c\rangle^{9}$
...

Question: Construct Petri net using regions



$\langle a,b \rangle^{33}$

$\langle b \rangle^{29}$

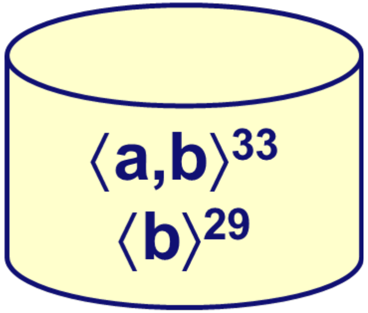


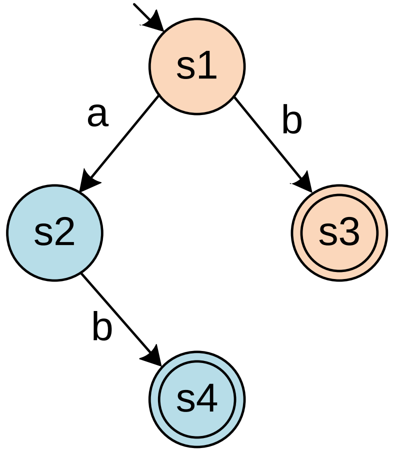
What is the Petri net constructed based on state-based regions?

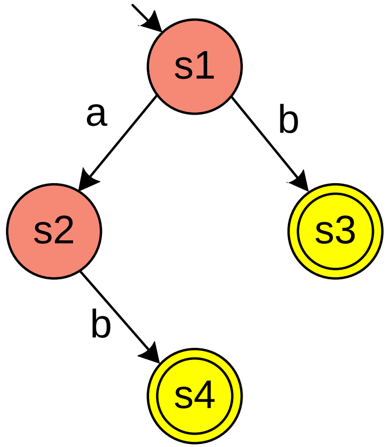
Answer (1/2)

Regions:

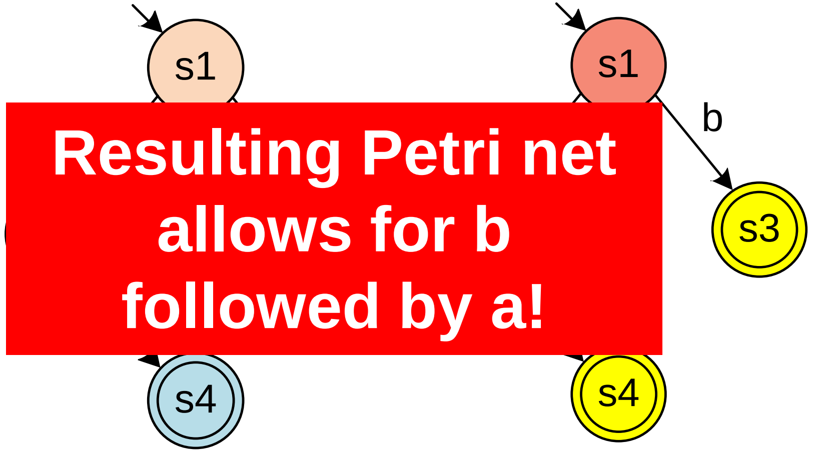
- **{s1,s2} (a does not cross, b exits)**
- **{s3,s4} (a does not cross, b enters)**
- **{s1,s3} (a exits and b does not cross)**
- **{s2,s4} (a enters and b does not cross)**



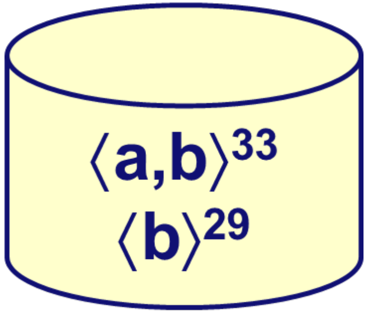


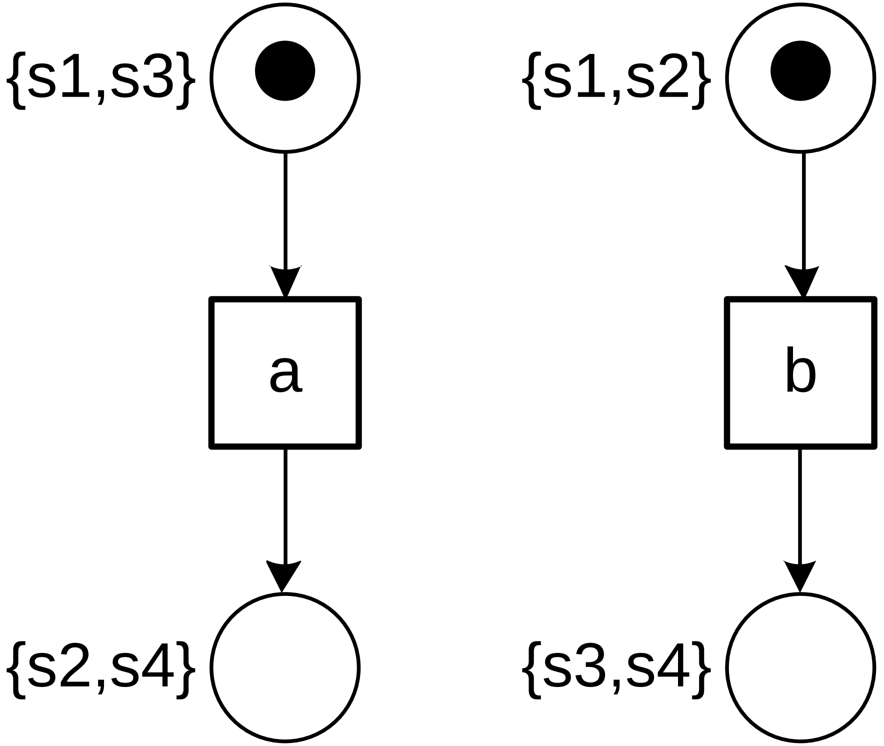


Answer (2/2)

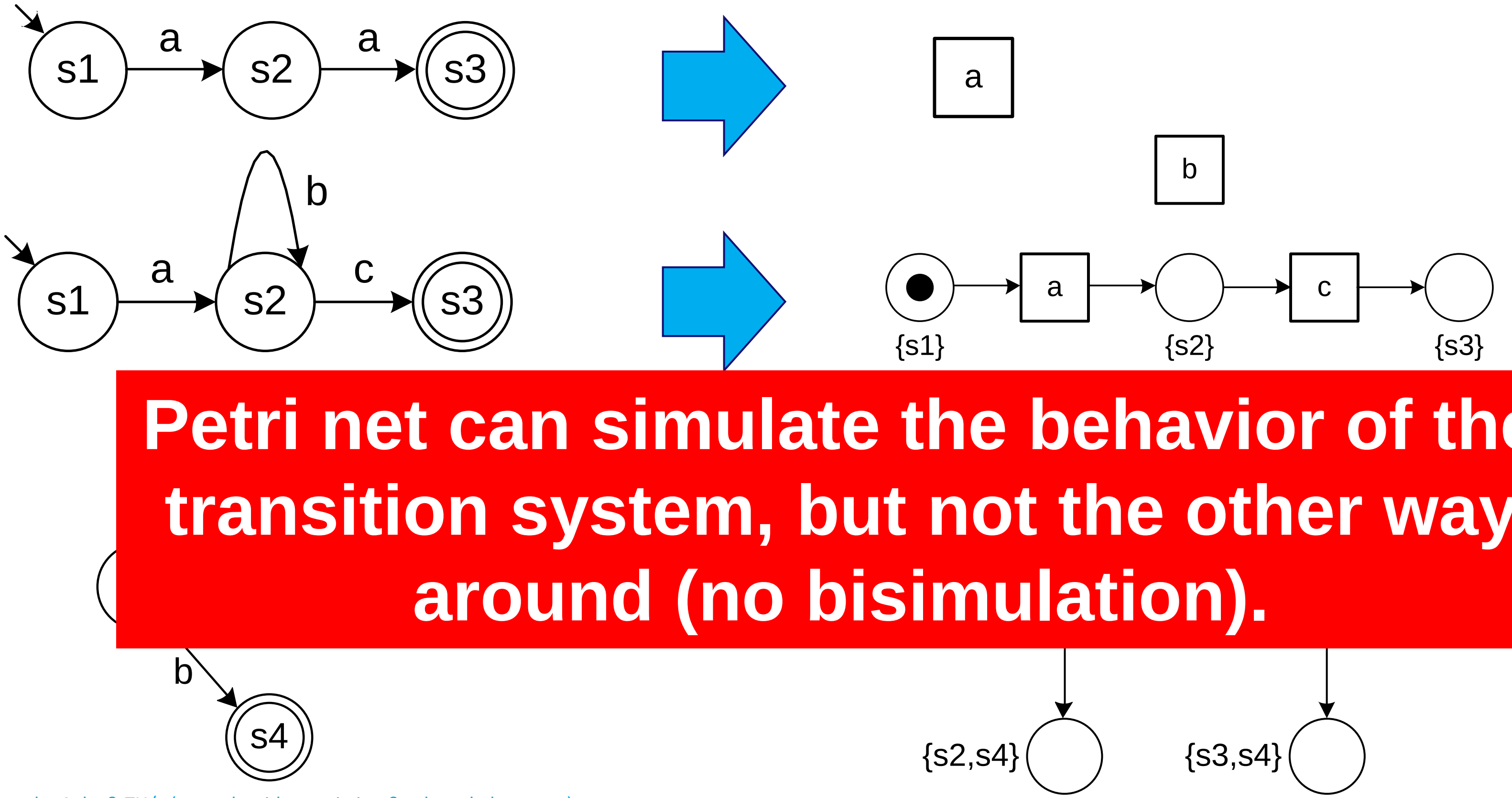


Resulting Petri net allows for b followed by a!



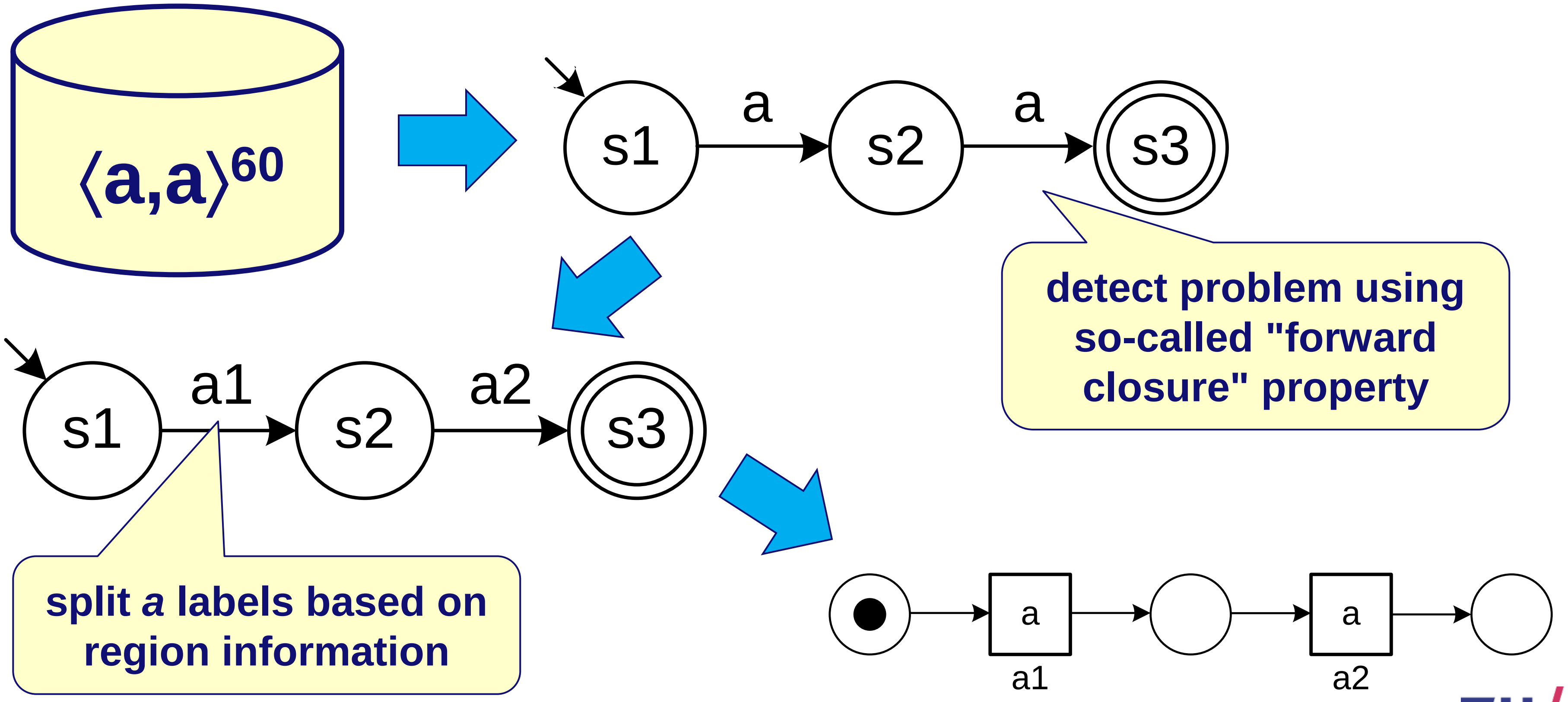


All underfitting …



Petri net can simulate the behavior of the transition system, but not the other way around (no bisimulation).

Refinement ensuring bisimulation



detect problem using so-called "forward closure" property

split *a* labels based on region information

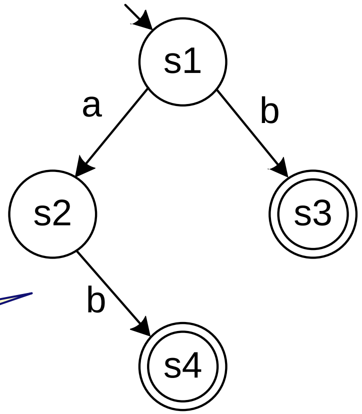
TU/e

Refinement ensuring bisimu

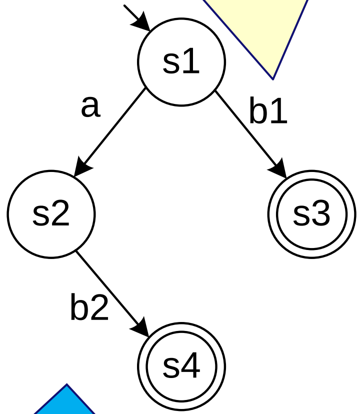
split *b* labels based on region information

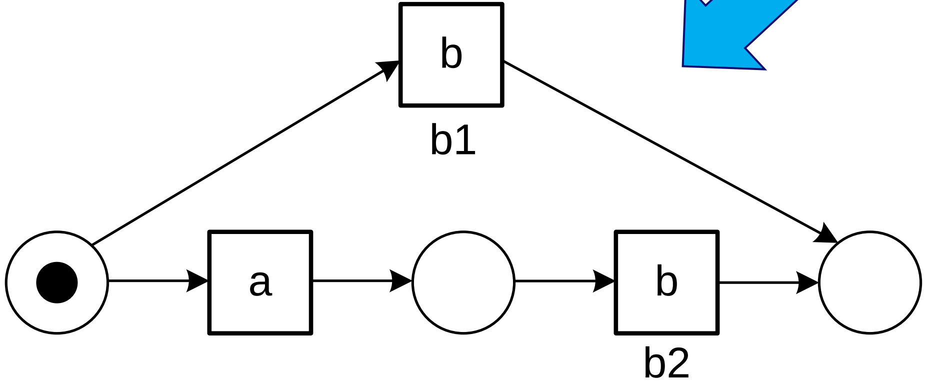
$\langle a,b\rangle^{33}$

$\langle b\rangle^{29}$

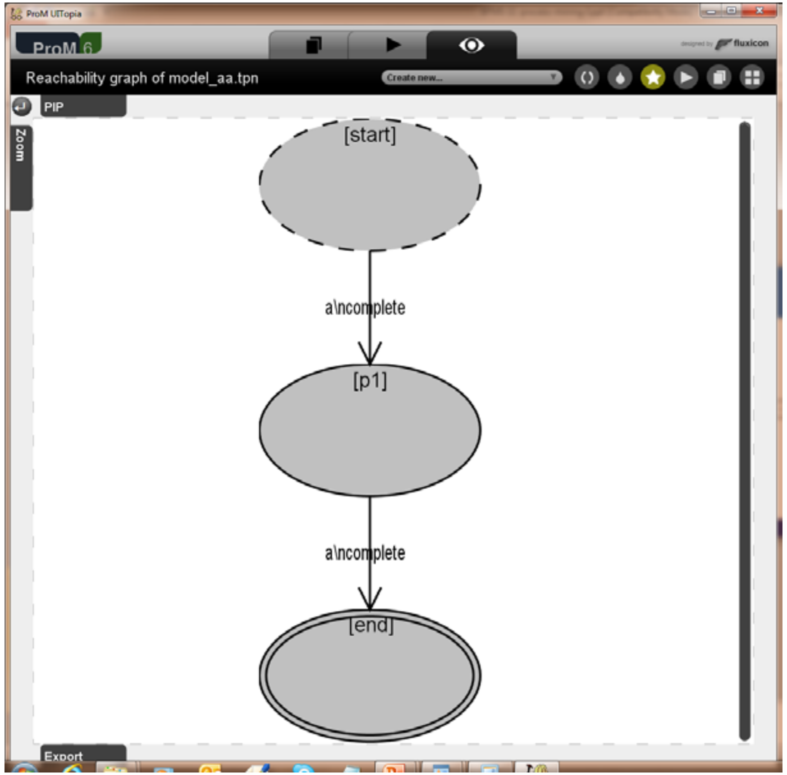


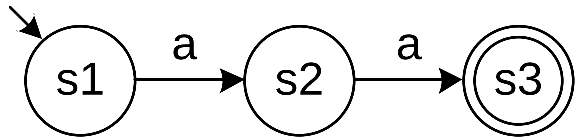
detect problem using so-called "forward closure" property

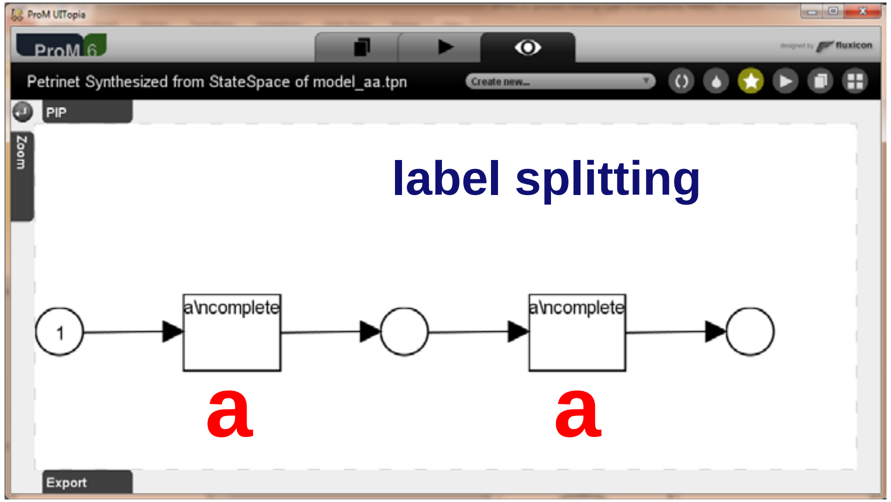




ProM plug-in ensures bisimulation





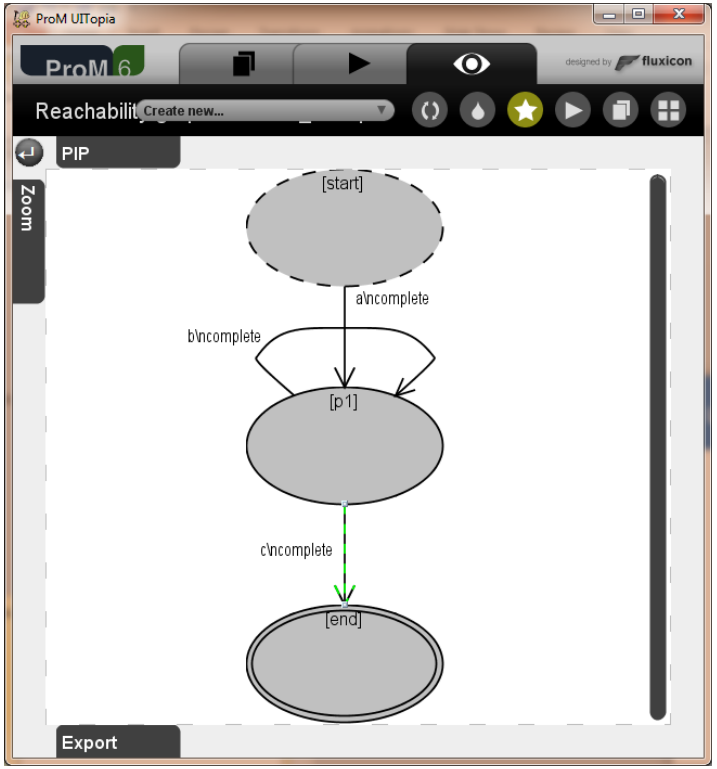


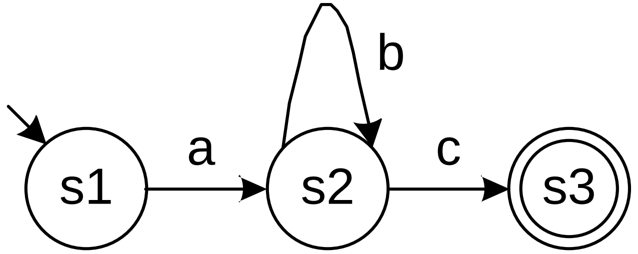
label splitting

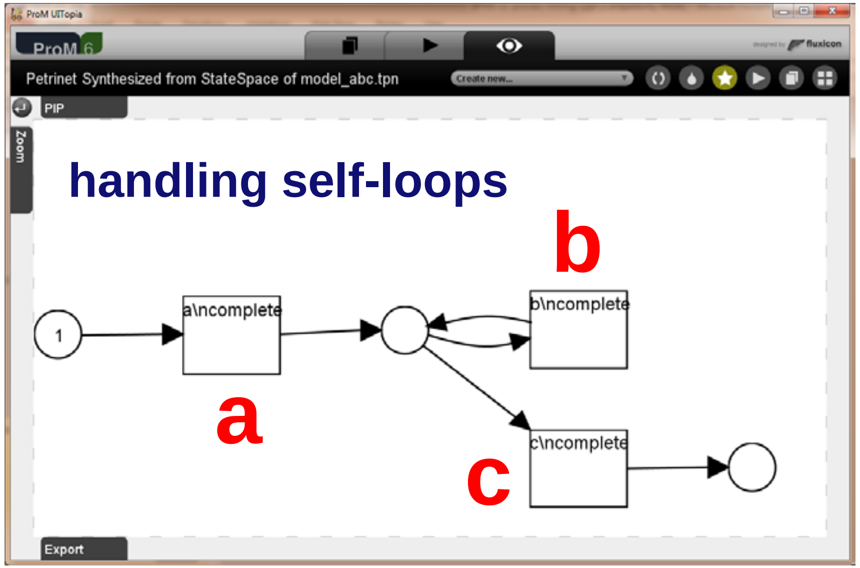
a

a

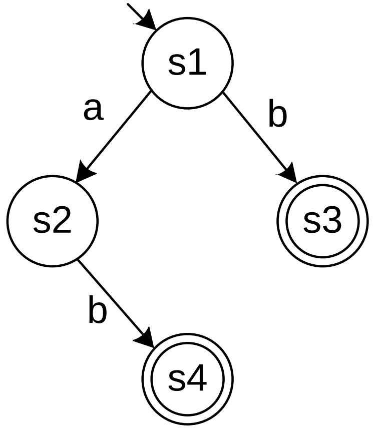
ProM plug-in ensures bisimulation

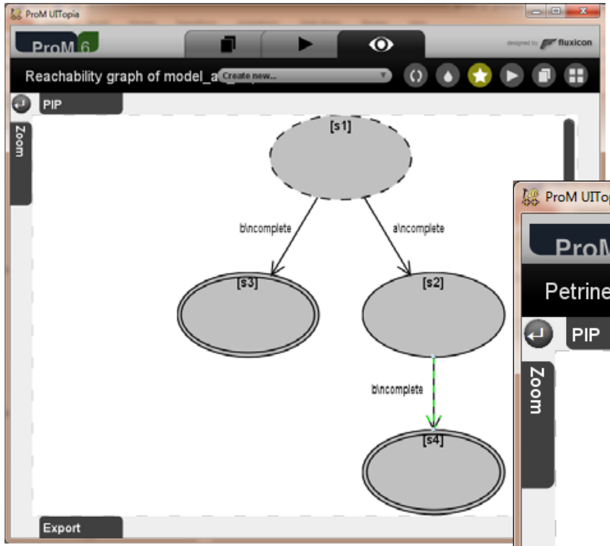


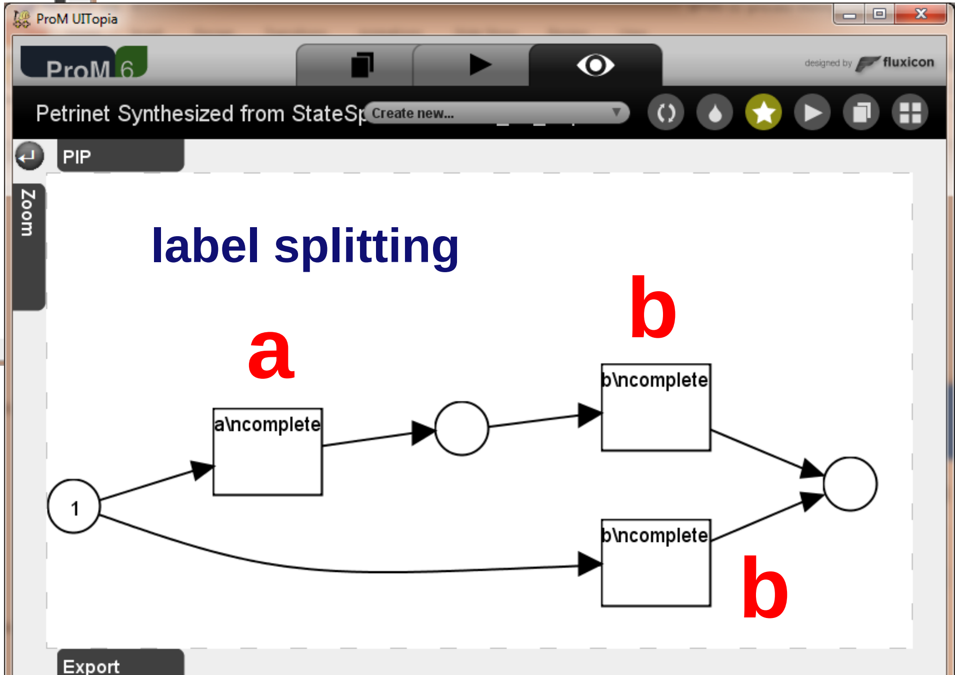




ProM plug-in ensures bisimulation

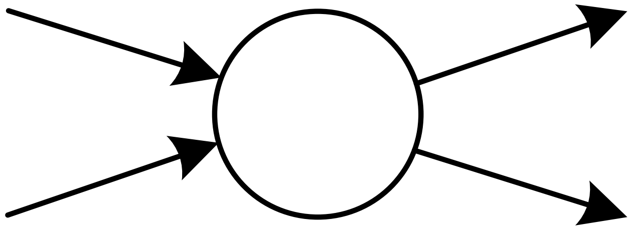






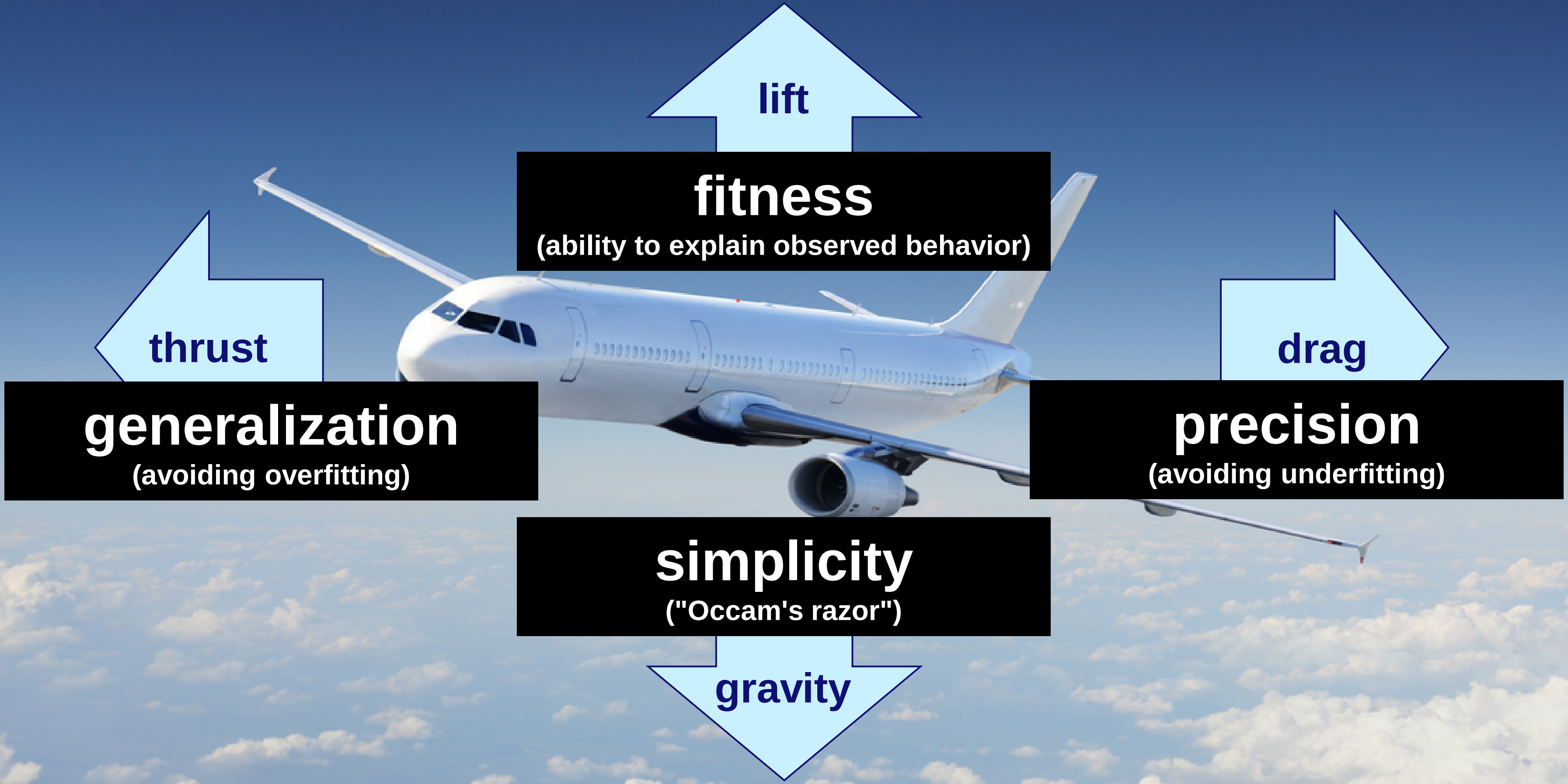
Many refinements are possible

- One can impose various constraints on places.
- Typical examples:
  - At most *k* input or *k* output arcs.
  - Only pure split or join places.
  - Only free-choice places (separate choice and synchronization).
  - ...



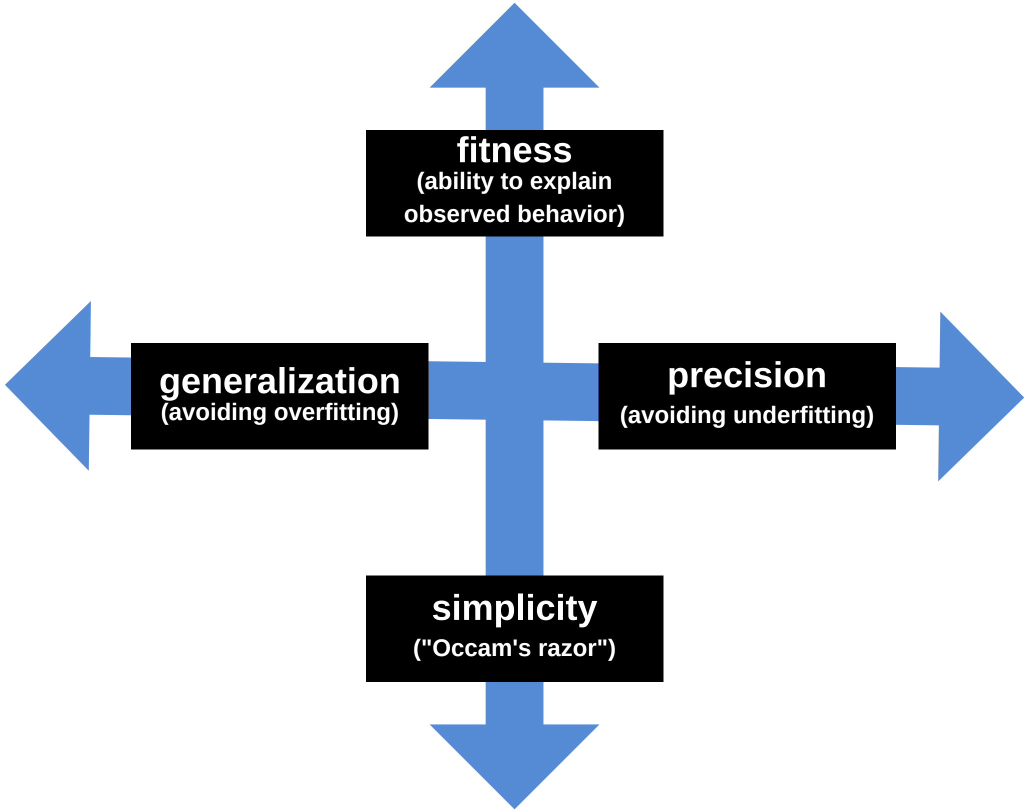
Weaknesses of approach based on state-based regions

- Inability to discover particular process constructs (can be handled through extensions of the basic algorithm).
- Inability to balance the four forces (fitness, precision, generalization, and simplicity) well.

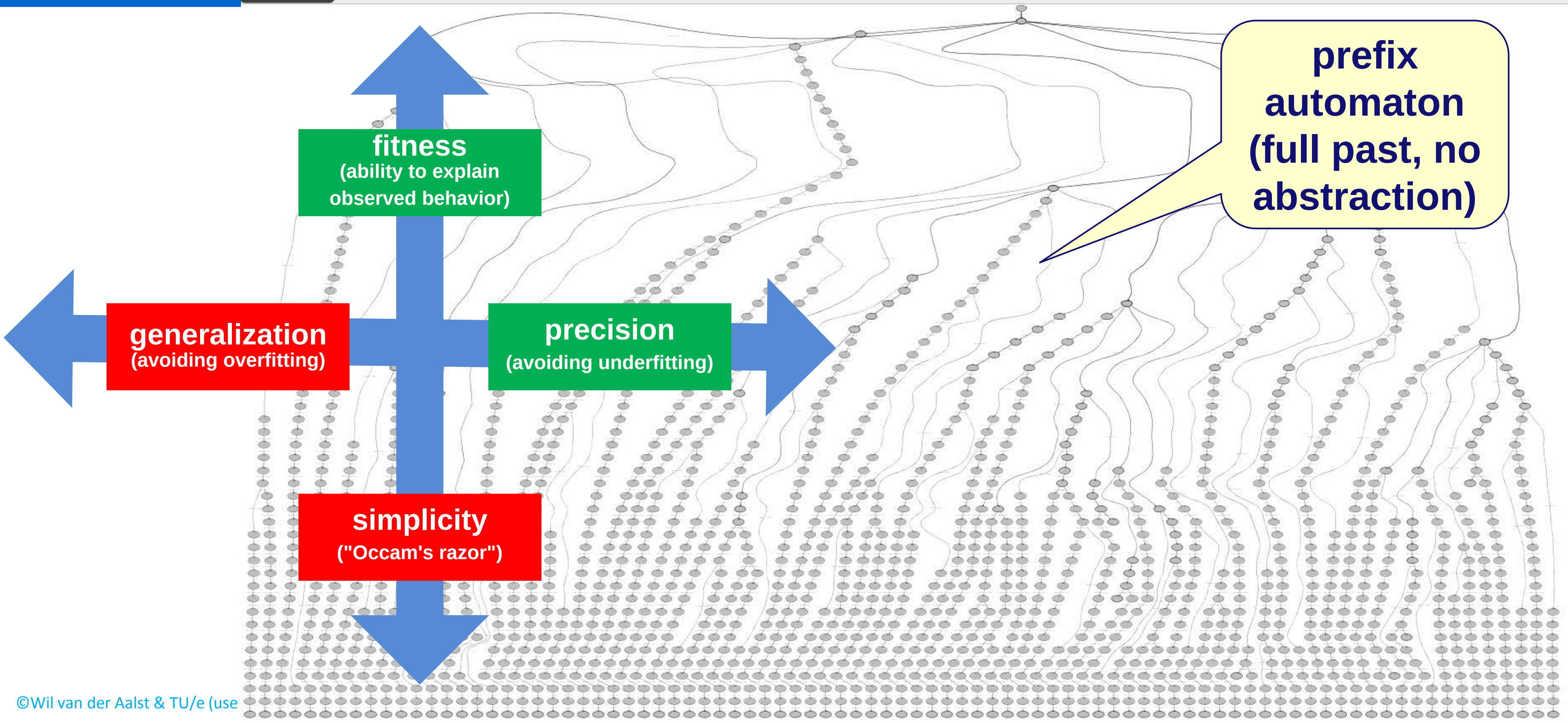

lift
fitness
(ability to explain observed behavior)
thrust
generalization
(avoiding overfitting)
drag
precision
(avoiding underfitting)
simplicity
("Occam's razor")
gravity

Four Forces

Step 1: Learning a transition system

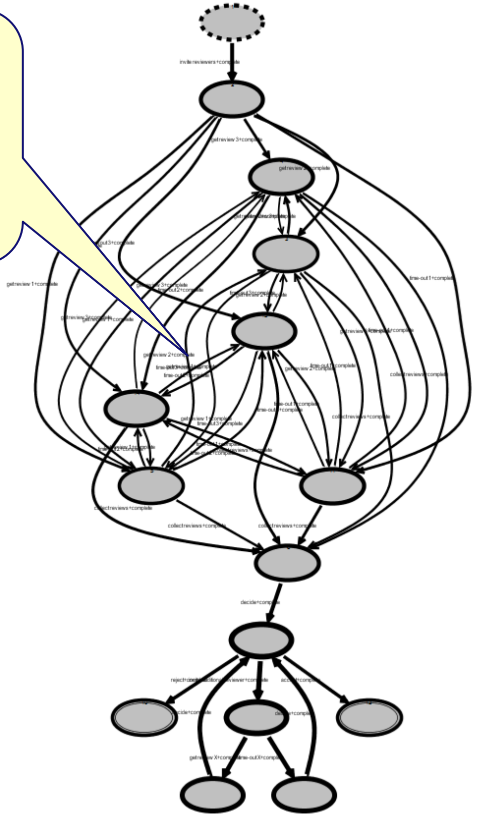


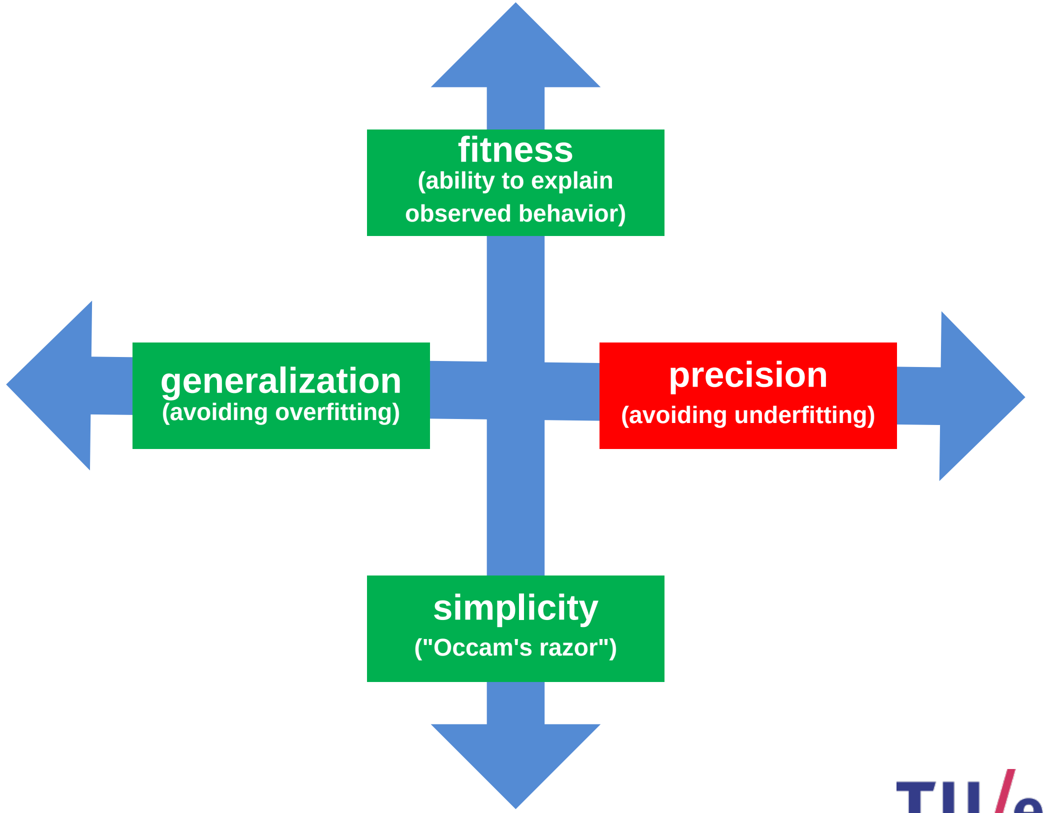
Step 1: Learning a transition system



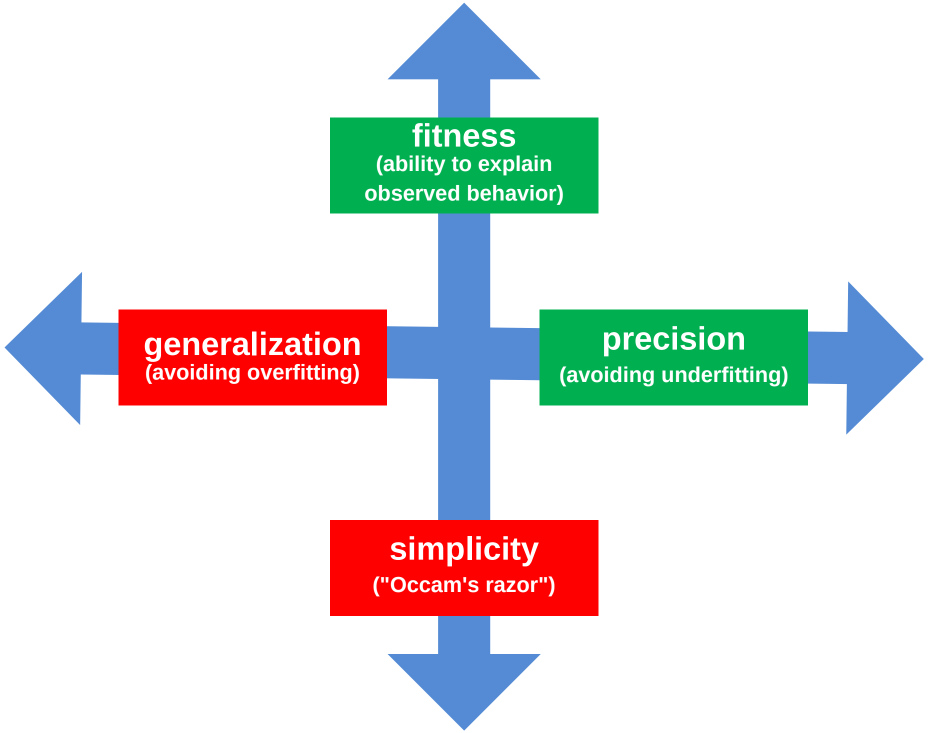
Step 1: Learning a transition system

state is based on last event only





Step 2: Discovering concurrency



- Classical state-based regions (including label splitting) ensure behavioral equivalence (bisimulation).
- No special attention for generalization or simplicity.

Summary

- Region-based techniques can be used to discover complex process patterns.
- Provide insights into the essence of process discovery.
- But:
  - Overfitting may be a problem (make sure the initial transition system is general enough).
  - Inability to leave out infrequent behavior (but can be done in the transition system).
  - Noise and incompleteness cannot be handled well.

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 P ess Discove niques

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

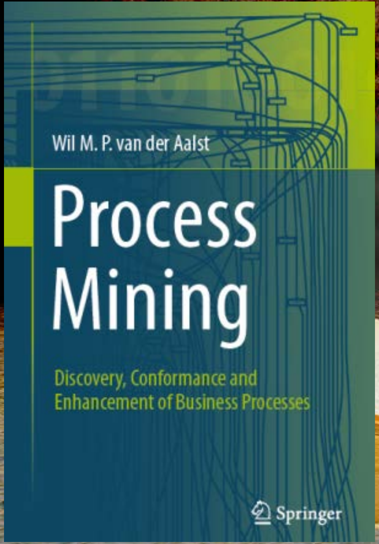
Chapter 11 Analyzing "Lasagna Processes"

Chapter 12 Analyzing "Spaghetti Processes"

Part V: Reflection

Chapter 13 Cartography and Navigation

Chapter 14 Epilogue



Wil M. P. van der Aalst

Process Mining

Discovery, Conformance and Enhancement of Business Processes

Springer