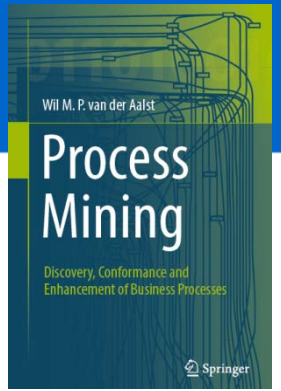
Process Mining: Data Science in Action

Learning Transition Systems

prof.dr.ir. Wil van der Aalst

www.processmining.org



Technische Universiteit Eindhoven University of Technology

Where innovation starts

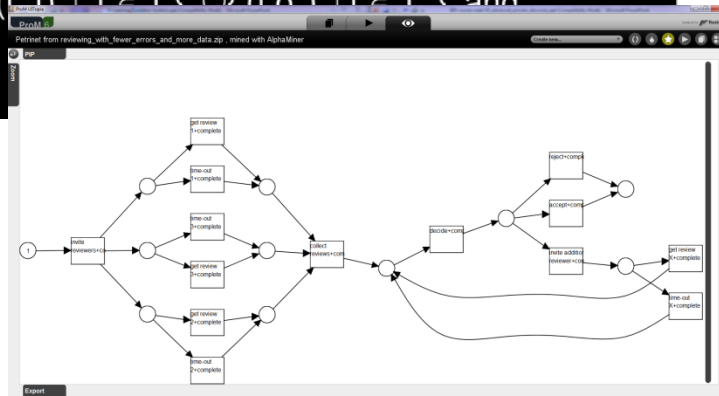
Discovery approaches seen thus far

Discovery approaches seen thus far

- **Alpha algorithm**

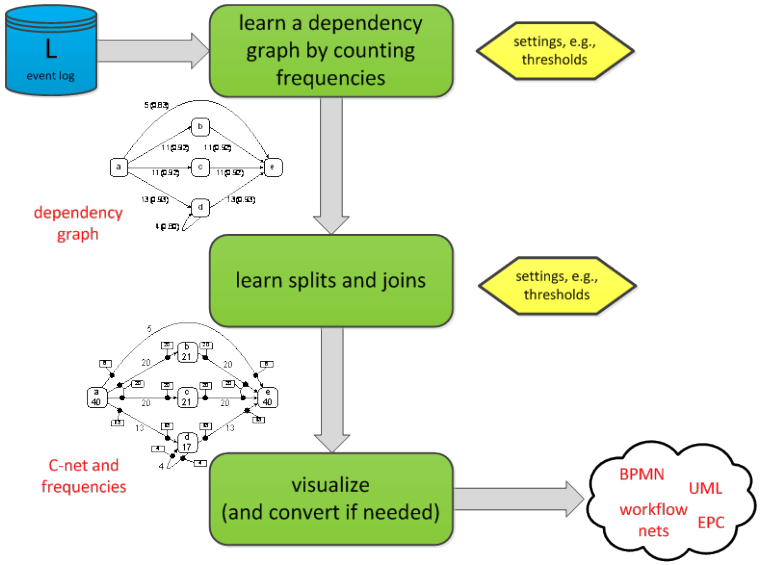
Let L be an event log over T. $\alpha(L)$ is defined as follows.

1. $T_L = \{ t \in T \mid \exists_{\sigma \in L}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in L}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in L}\, t = \mathit{last}(\sigma) \}$,
4. $X_L = \{ (A,B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_L b \wedge \forall_{a1,a2 \in A}\, a_1 \#_L a_2 \wedge \forall_{b1,b2 \in B}\, b_1 \#_L b_2 \}$,
5. $Y_L = \{ (A,B) \in X_L \mid \forall_{(A',B') \in X_L}\, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_L = \{ p_{(A,B)} \mid (A,B) \in Y_L \} \cup \{ i_L, o_L \}$,
7. $F_L = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_L \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_L \wedge b \in B \} \cup \{ (i_L, t) \mid t \in T_I \} \cup \{ (t, o_L) \mid t \in T_O \}$, and
8. $\alpha(L) = (P_L, T_L, F_L)$.



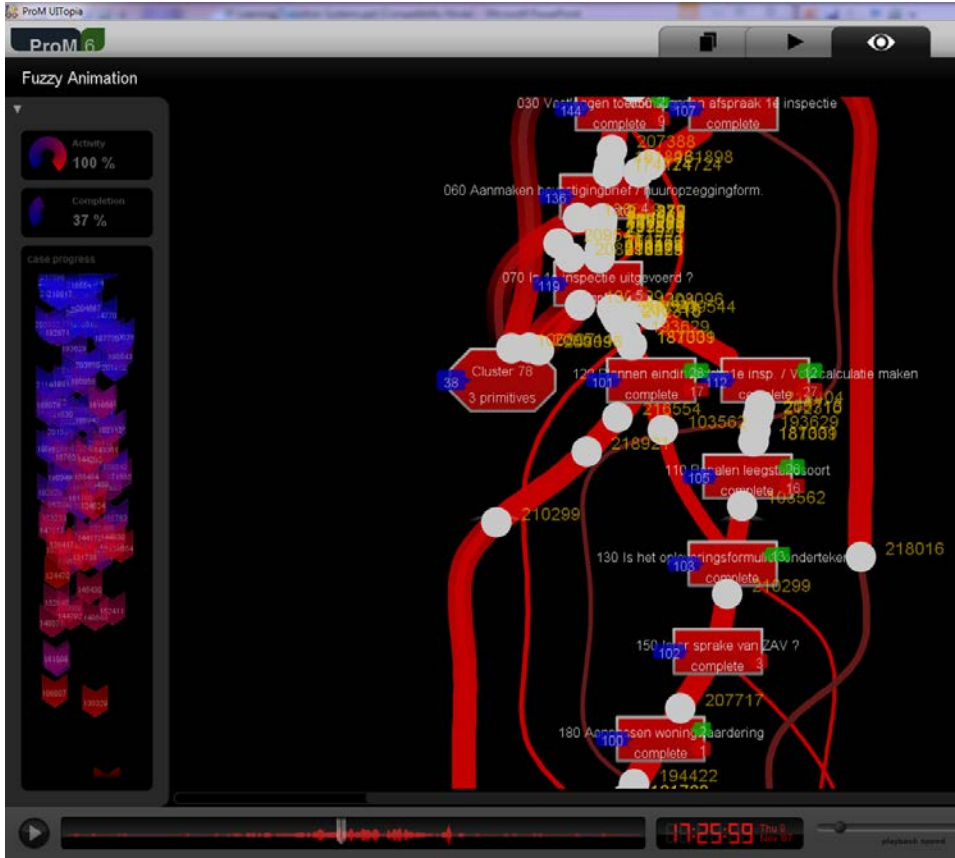
Discovery approaches seen thus far

- **Alpha algorithm**
- **Heuristic mining**
  - **dependency graph**
  - **C-net**



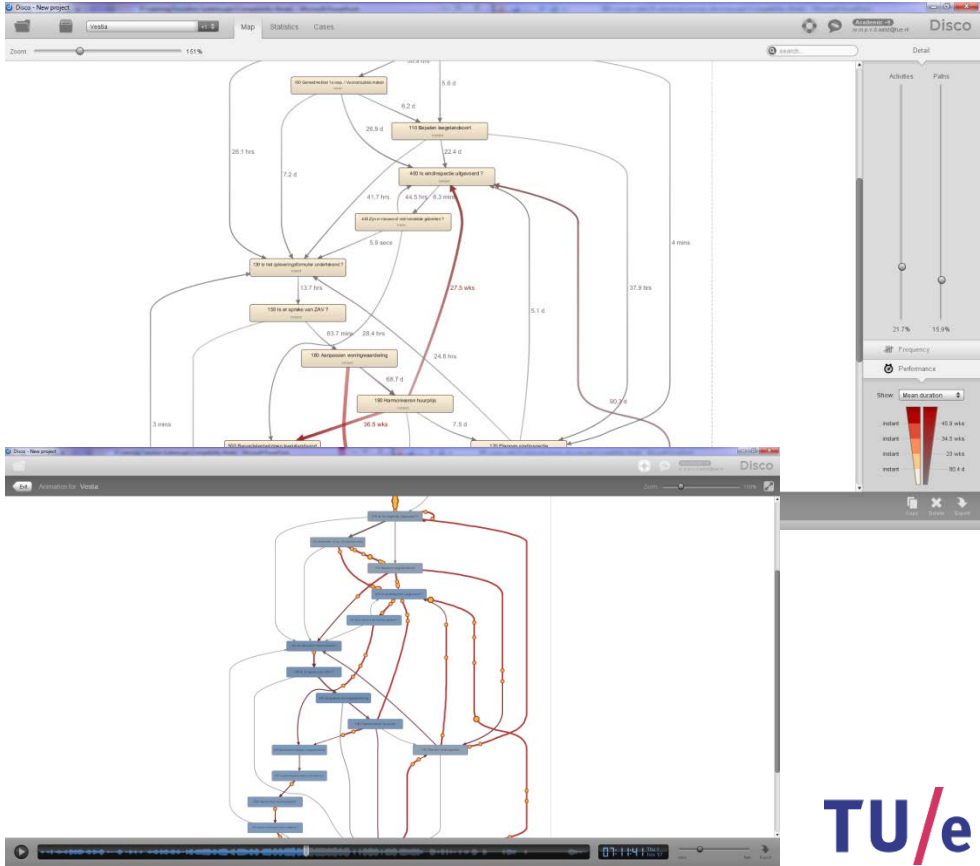
Discovery approaches seen thus far

- **Alpha algorithm**
- **Heuristic mining**
- **Fuzzy miner (tool only)**

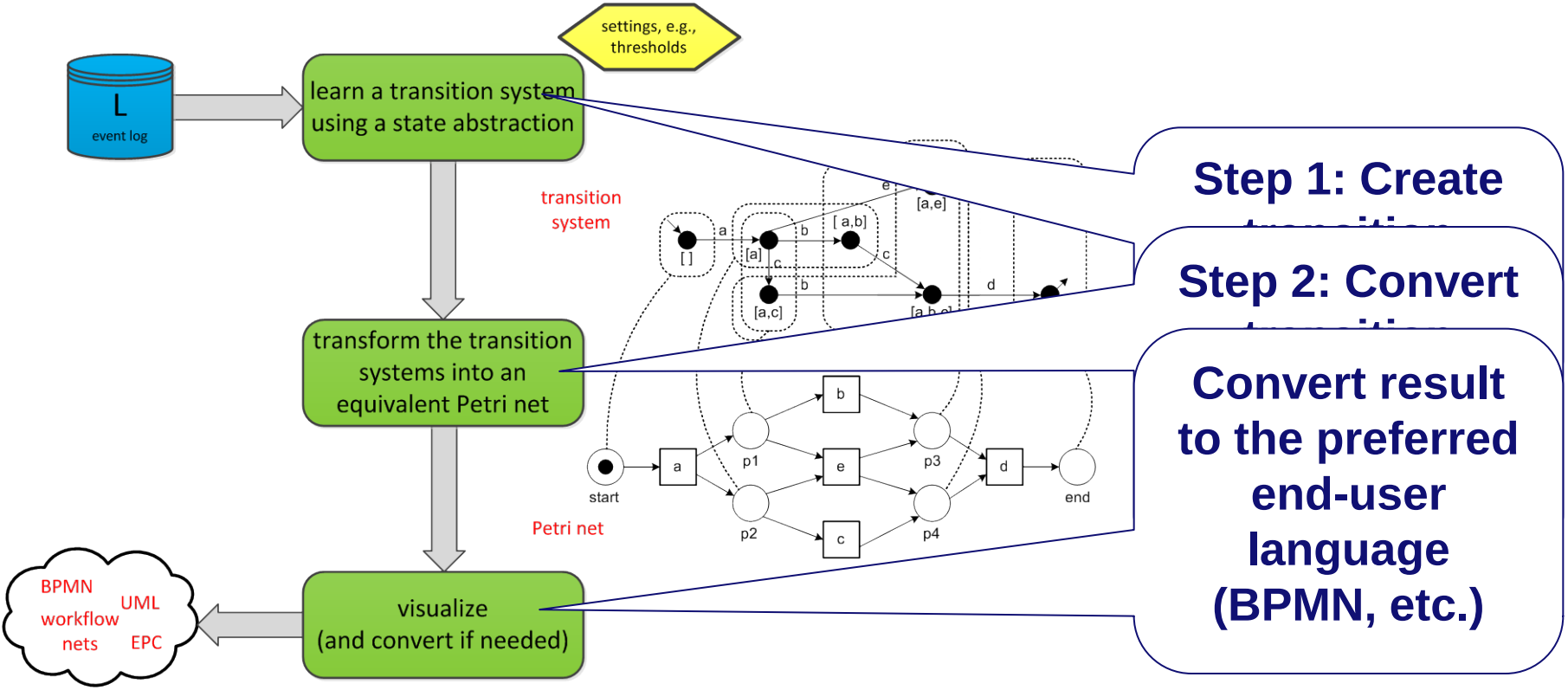


Discovery approaches seen thus far

- **Alpha algorithm**
- **Heuristic mining**
- **Fuzzy miner (tool only)**
- **Disco (tool only)**

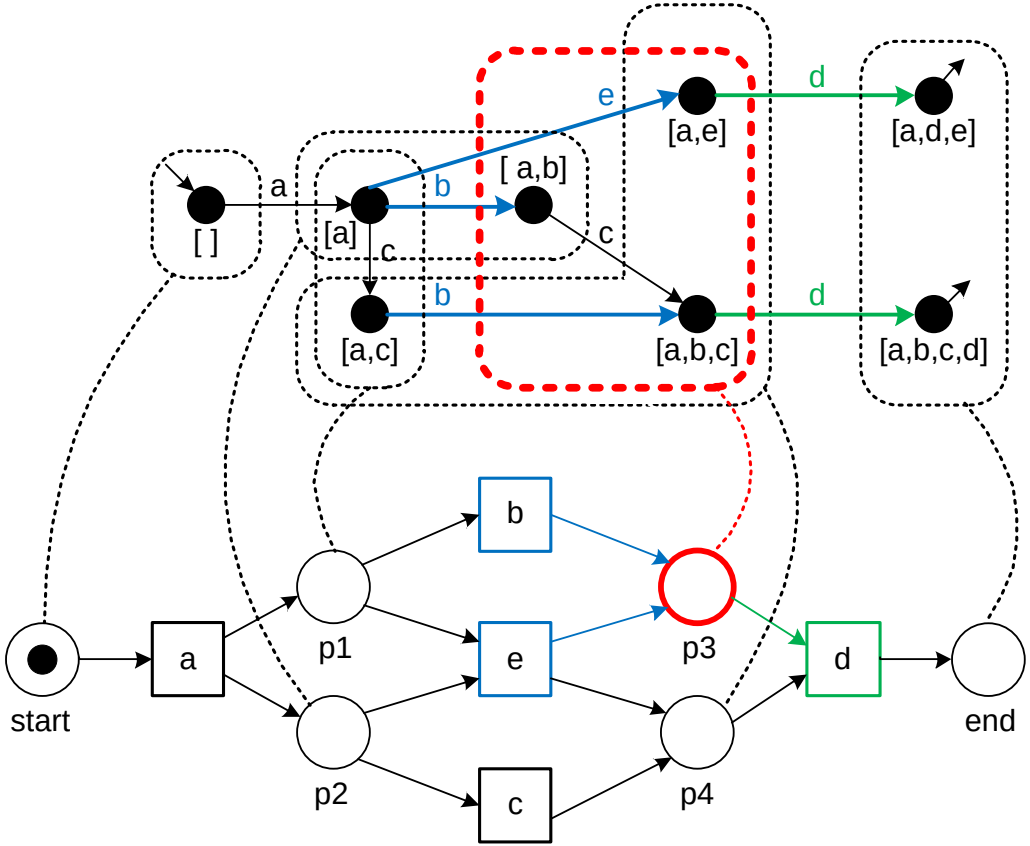


Another two-phase approach
(using state-based regions)

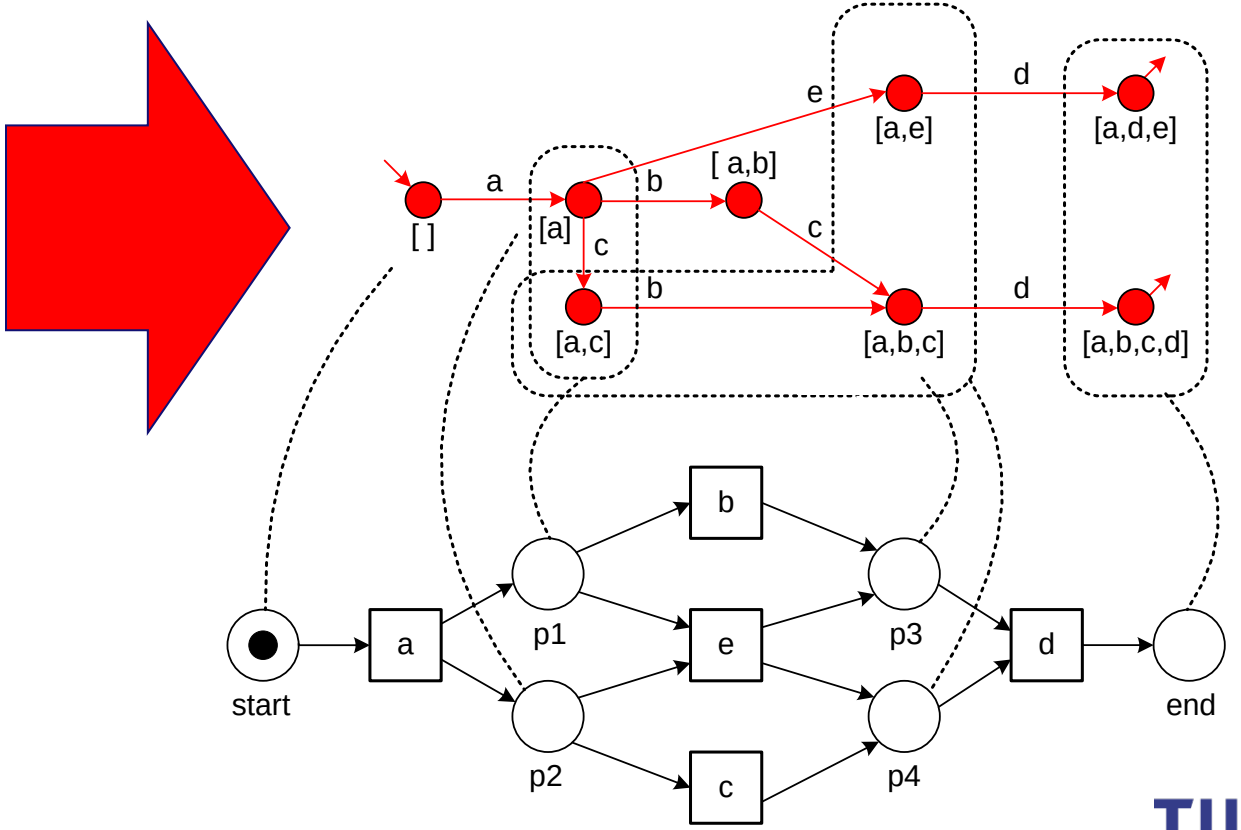


Second step: State-based regions

It is all about discovering concurrency …



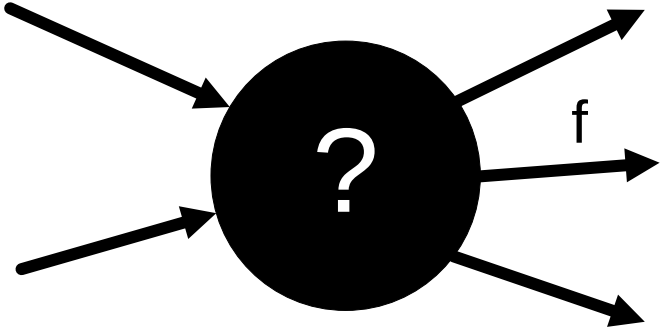
Today's focus



Learning a transition system

current state

trace: **a b c d c d c d e f a g h h h i**

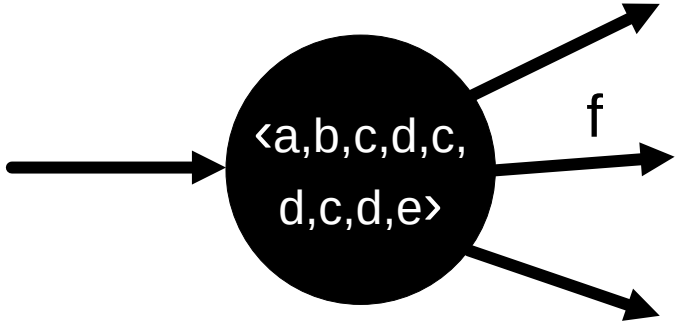


Learning a Transition System

current state

trace: **a b c d c d c d e f a g h h h i**

past

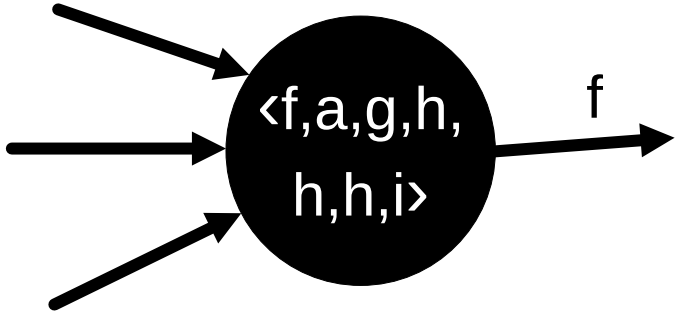


Learning a Transition System

current state

trace: **a b c d c d c d e f a g h h h i**

future

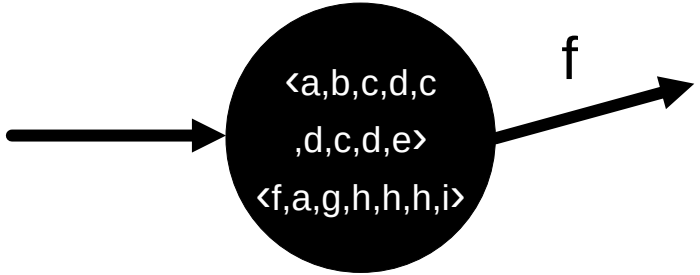


Learning a Transition System

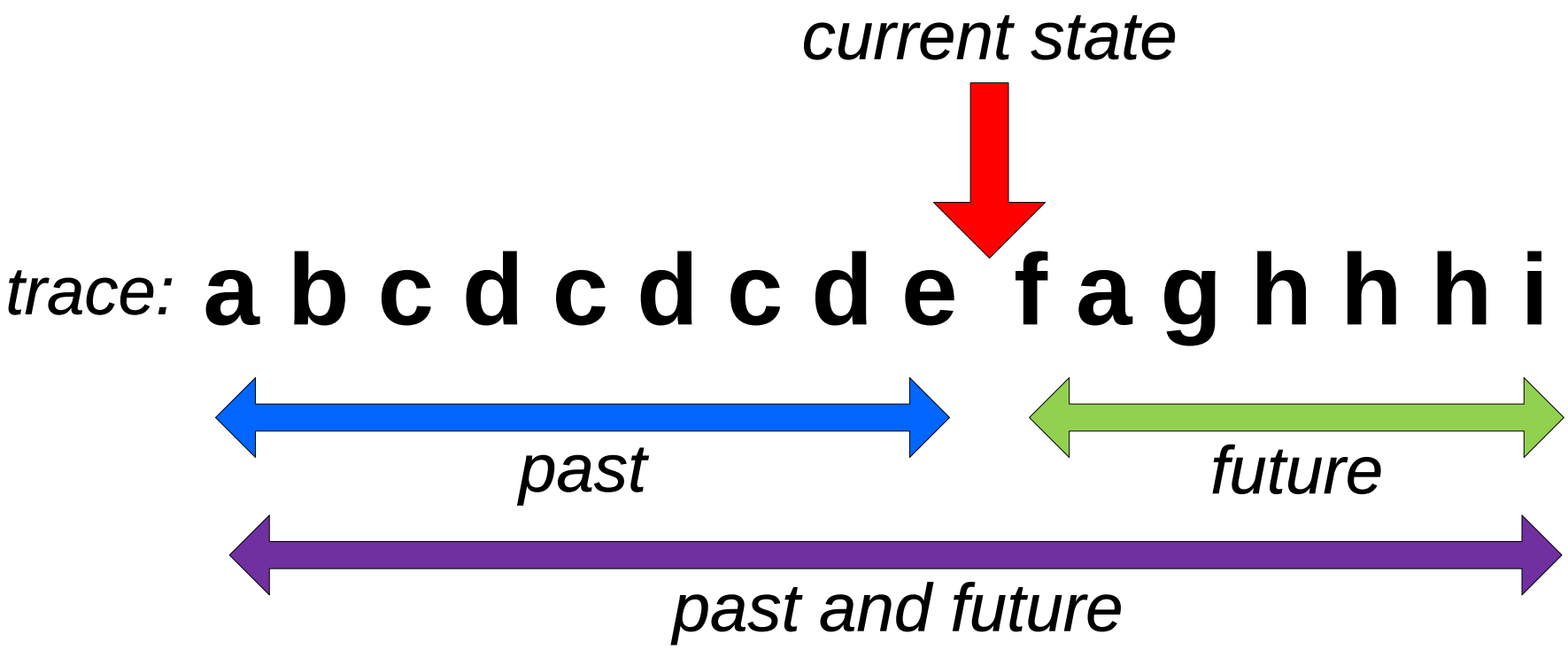
current state

trace: **a b c d c d c d e f a g h h h i**

past and future



Learning a Transition System



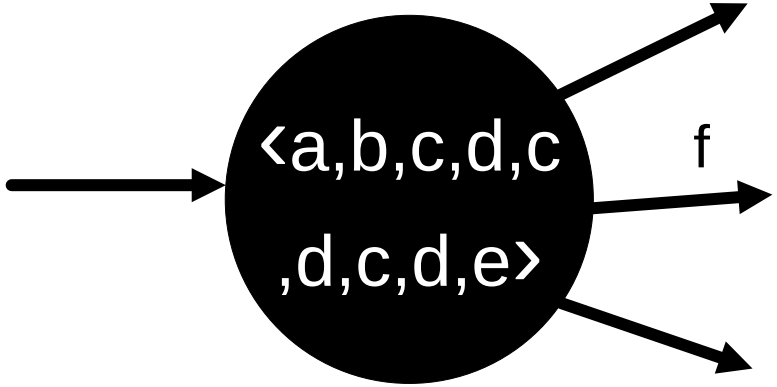
Sequence classification based on past

resulting transition system also known as the "prefix automaton"

current state

a b c d c d c d e f a g h h h i

past



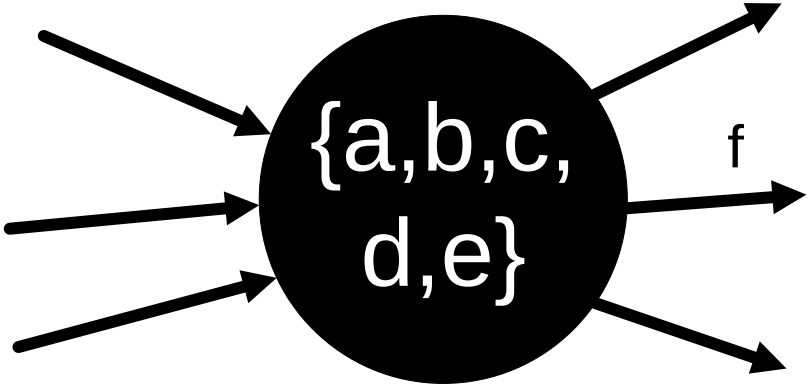
order and frequency matter

Set abstraction based on past

current state

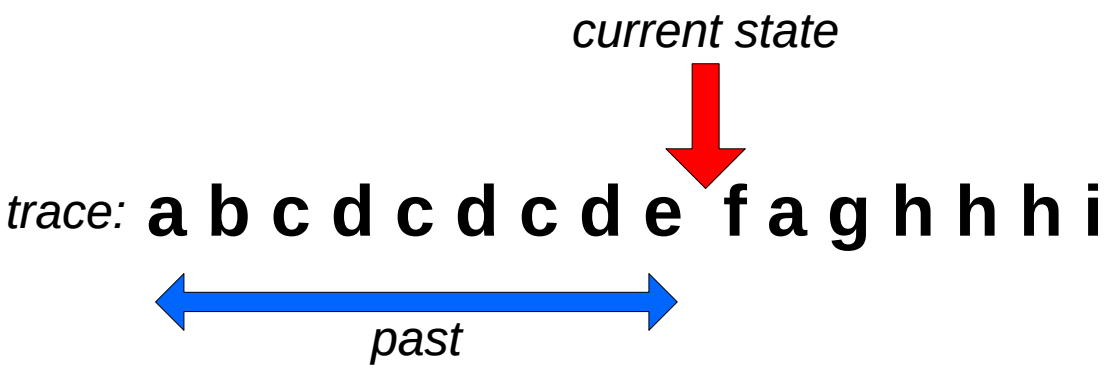
trace: **a b c d c d c d e f a g h h h i**

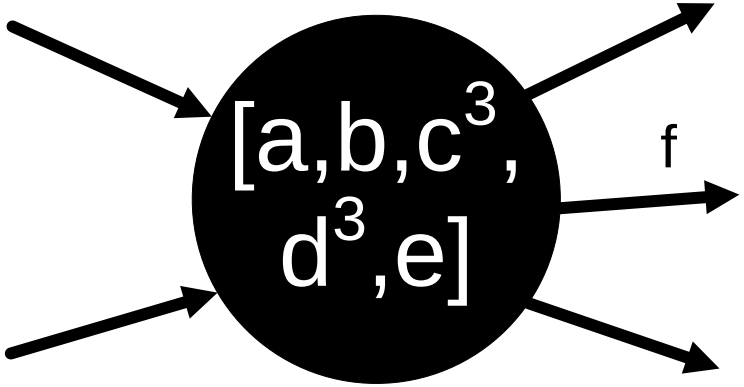
past



order and frequency don't matter

Multiset abstraction based on past





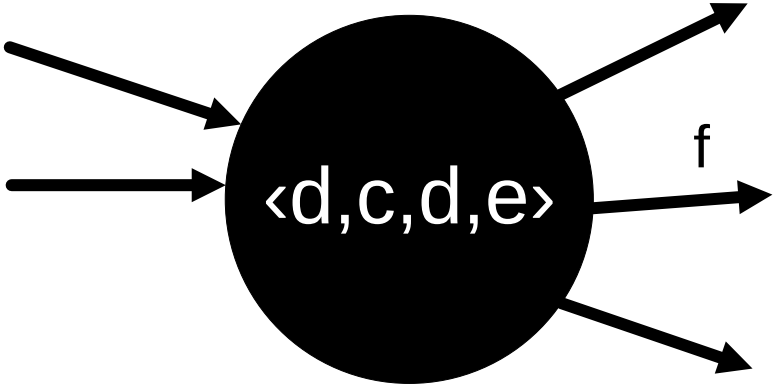
only frequency matters

Sequence abstraction based on last 4 events

current state

trace: **a b c d c d c d e f a g h h h i**

past 4 events



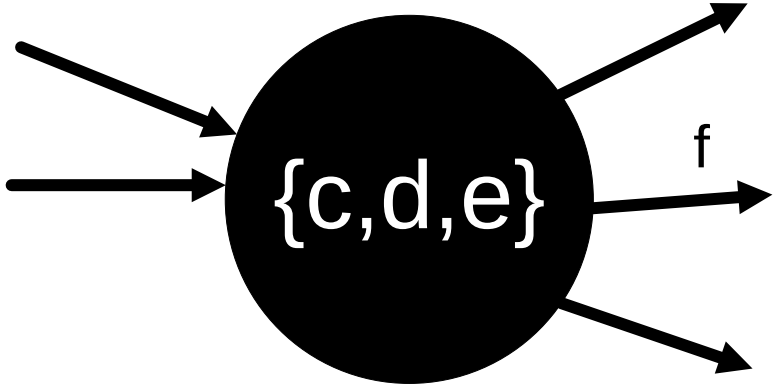
***k*-tail**

Set abstraction based on last 4 events

current state

trace: **a b c d c d c d e f a g h h h i**

past 4 events

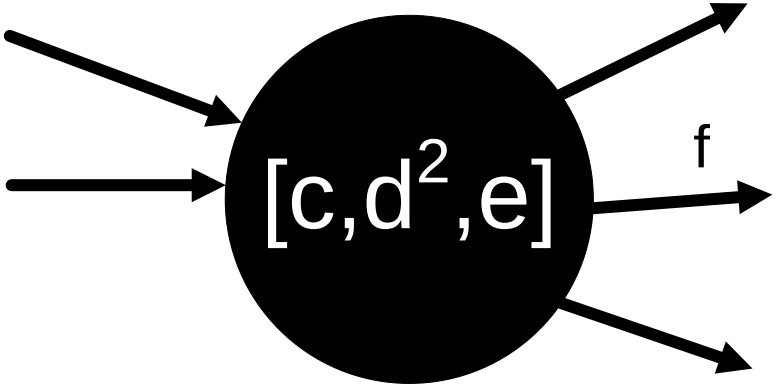


Multiset abstraction based on last 4 events

current state

trace: **a b c d c d c d e f a g h h h i**

past 4 events

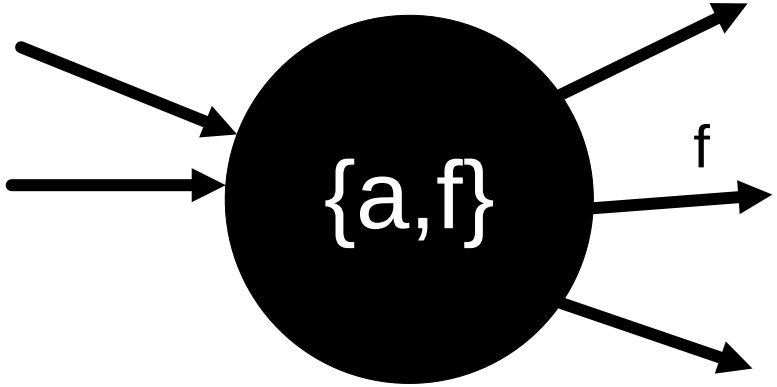


Set abstraction based on the next 2 events

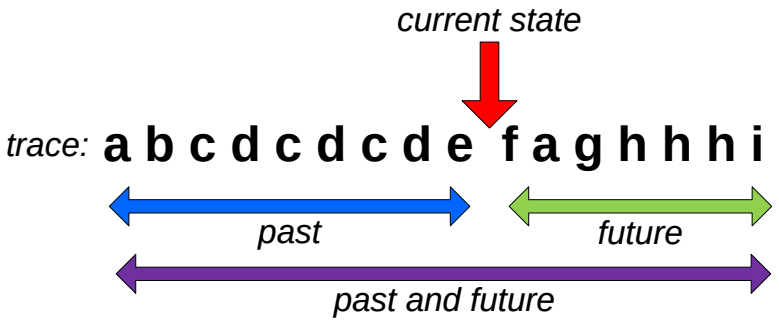
current state

trace: **a b c d c d c d e f a g h h h i**

next two events



Summary

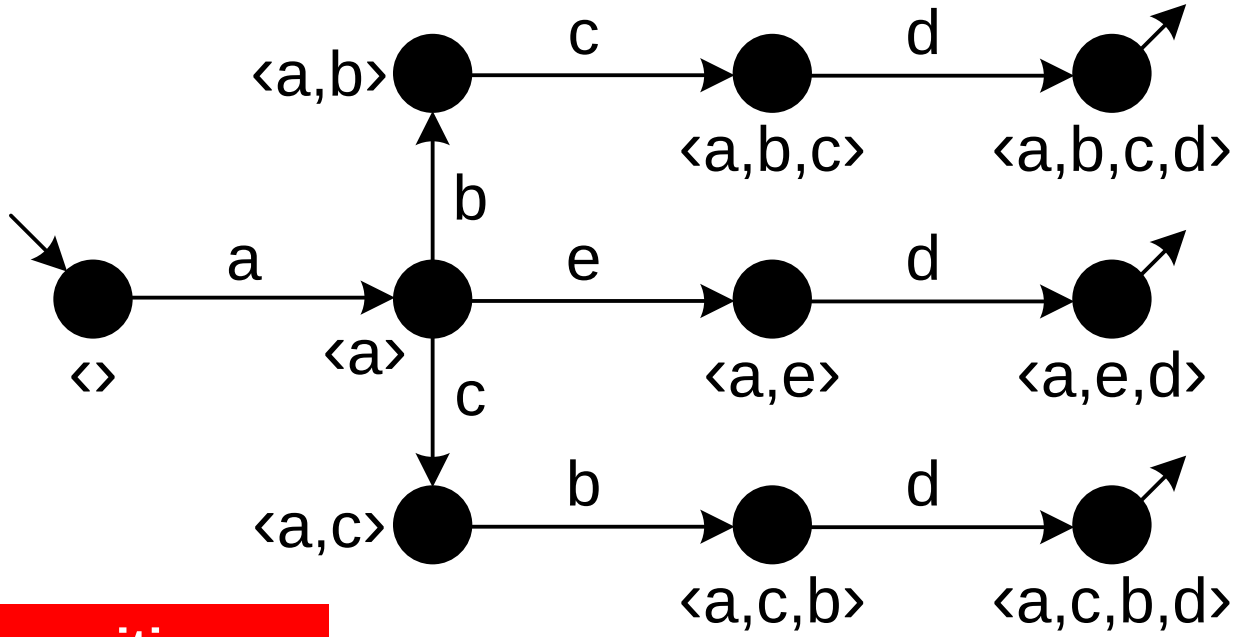


- **Position in trace determines the current state.**
- **State based on past, future, or past+future.**
- **Sequence, multiset, set abstraction.**
- **Limited horizon to abstract further (e.g., *k*-tail).**

Example

$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

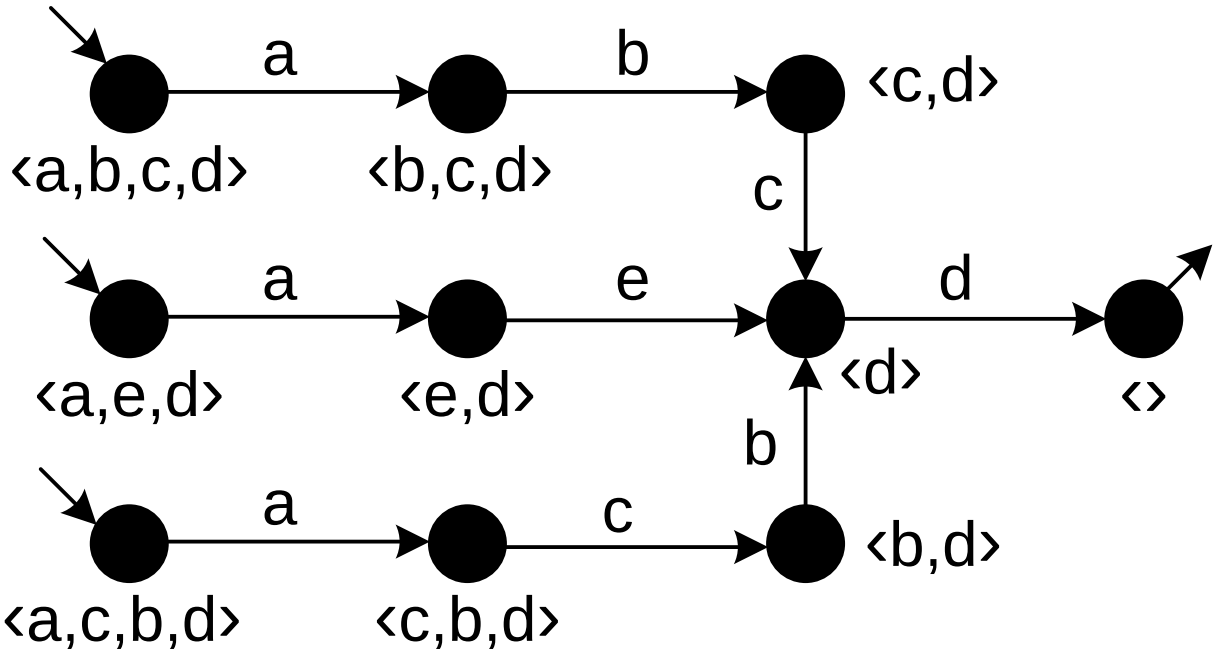
Past without abstraction



resulting transition system also known as the "prefix automaton"

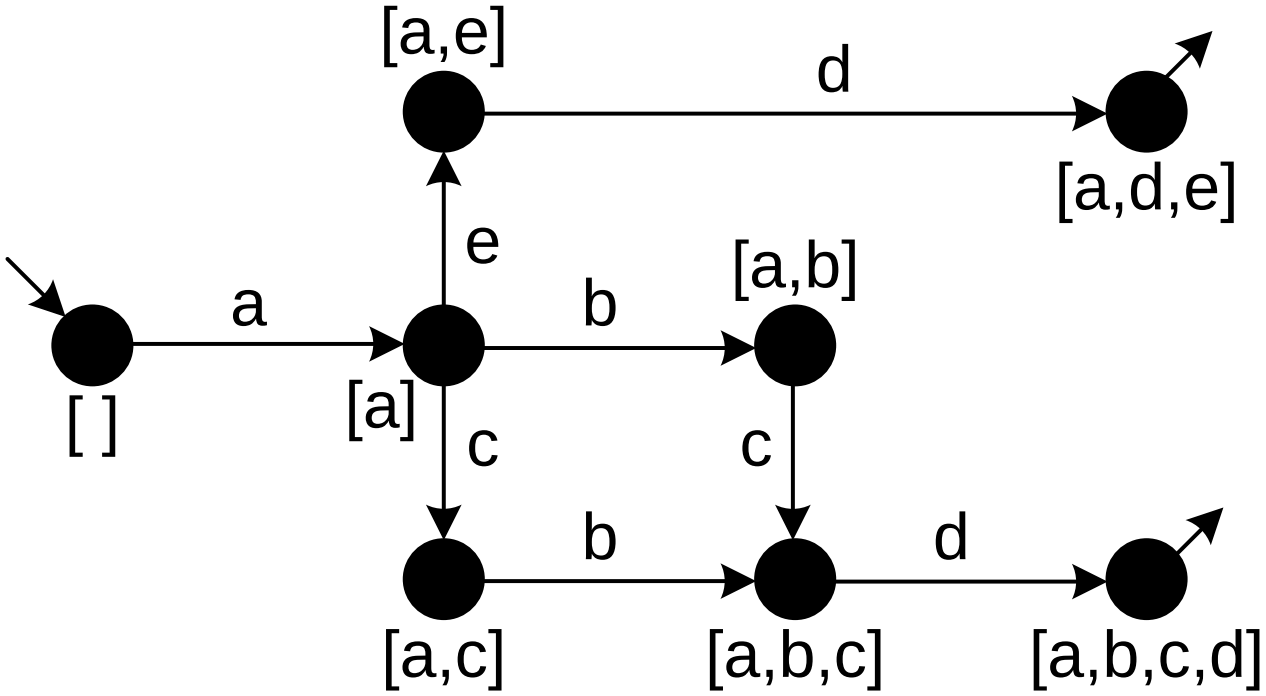
$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

Future without abstraction



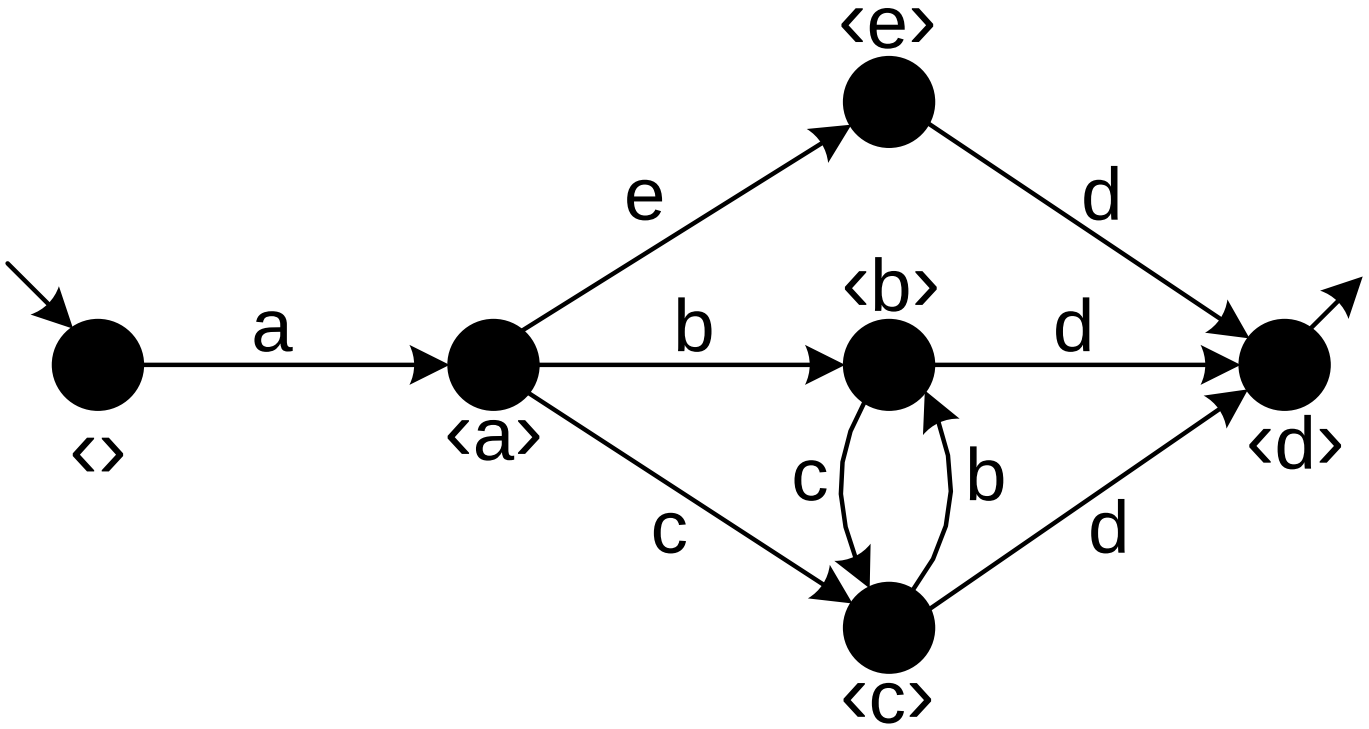
$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

Past with multiset abstraction



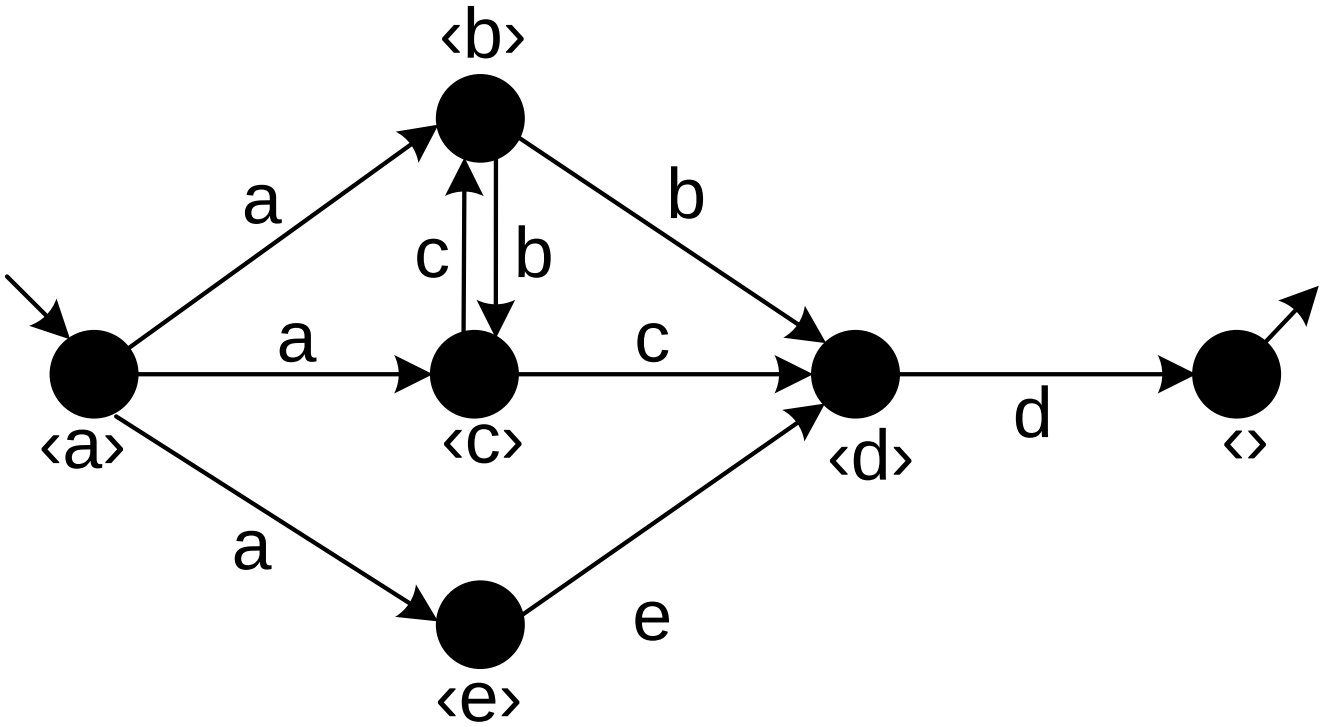
$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

Only last event matters for state



$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

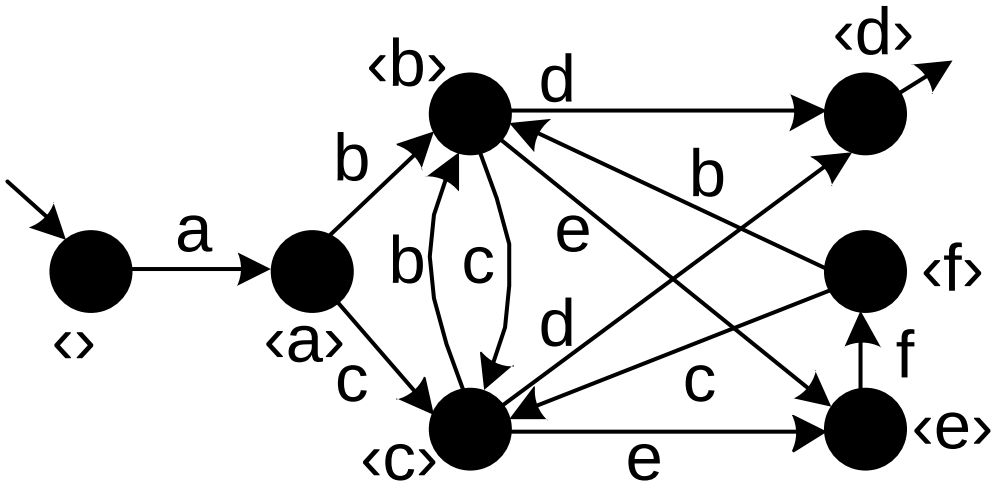
Only next event matters for state



$$L_1 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^2, \langle a,e,d\rangle]$$

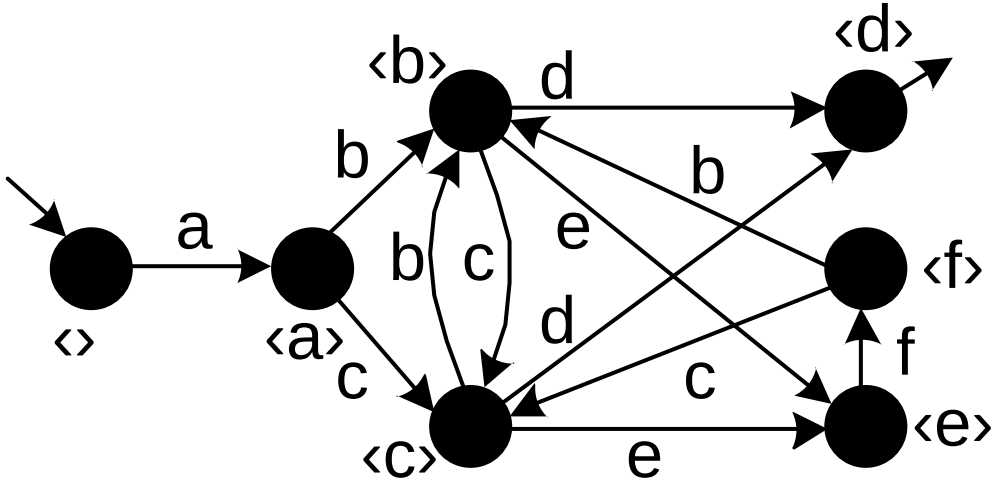
Question:
What kind of abstraction was used?

$$L_2 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^4, \langle a,b,c,e,f,b,c,d\rangle^2, \langle a,b,c,e,f,c,b,d\rangle,$$
$$\langle a,c,b,e,f,b,c,d\rangle^2, \langle a,c,b,e,f,b,c,e,f,c,b,d\rangle]$$



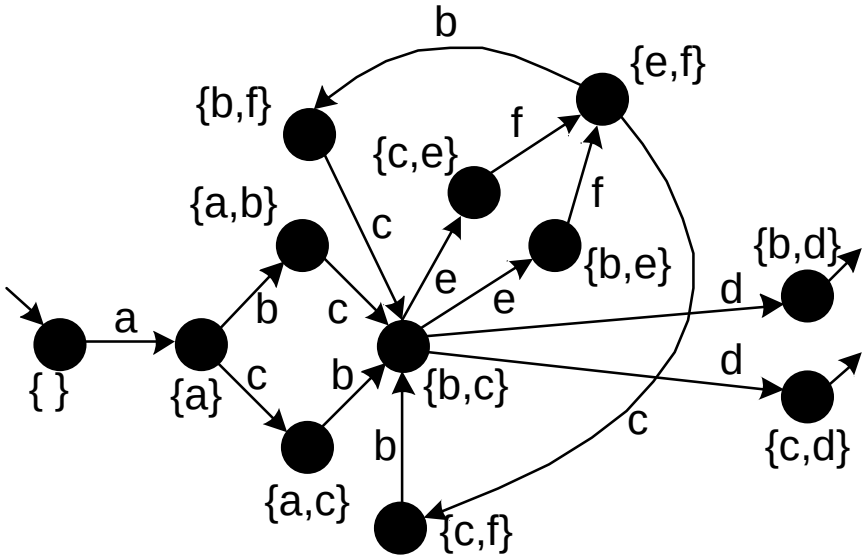
Answer:
Only last event matters

$$L_2 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^4, \langle a,b,c,e,f,b,c,d\rangle^2, \langle a,b,c,e,f,c,b,d\rangle,$$
$$\langle a,c,b,e,f,b,c,d\rangle^2, \langle a,c,b,e,f,b,c,e,f,c,b,d\rangle]$$



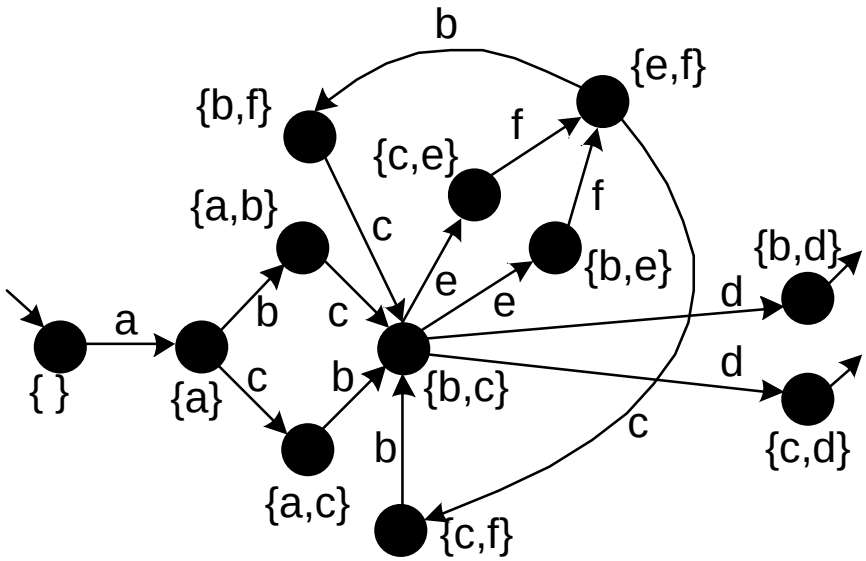
Question:
What kind of abstraction was used?

$$L_2 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^4, \langle a,b,c,e,f,b,c,d\rangle^2, \langle a,b,c,e,f,c,b,d\rangle,$$
$$\langle a,c,b,e,f,b,c,d\rangle^2, \langle a,c,b,e,f,b,c,e,f,c,b,d\rangle]$$



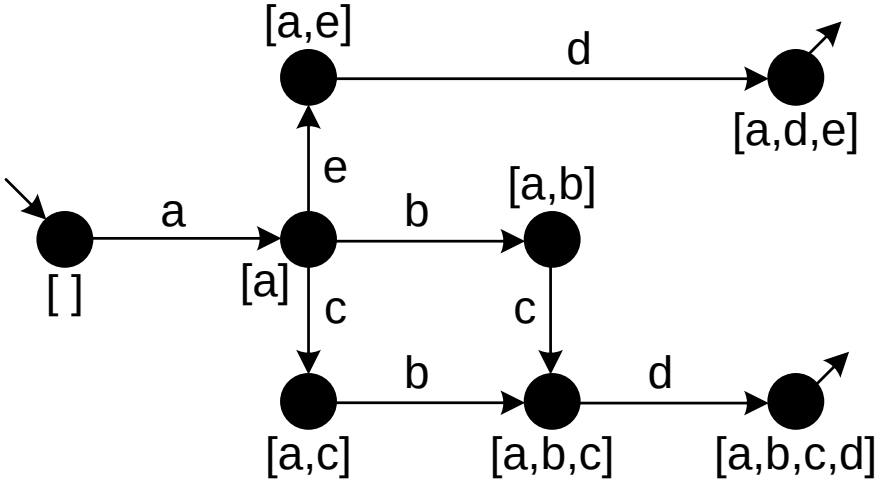
Answer:
Only last two events matters (set abstraction)

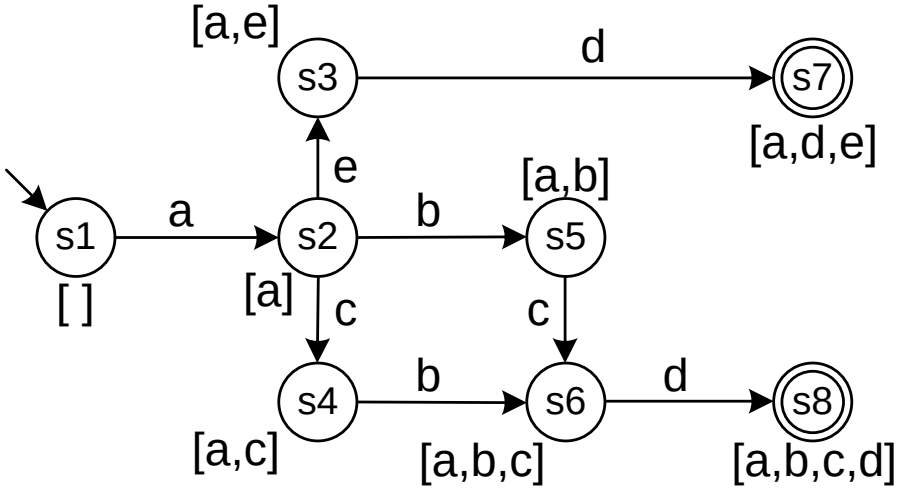
$$L_2 = [\langle a,b,c,d\rangle^3, \langle a,c,b,d\rangle^4, \langle a,b,c,e,f,b,c,d\rangle^2, \langle a,b,c,e,f,c,b,d\rangle,$$
$$\langle a,c,b,e,f,b,c,d\rangle^2, \langle a,c,b,e,f,b,c,e,f,c,b,d\rangle]$$



Last two events are considered without looking at the order.

Remark about notation

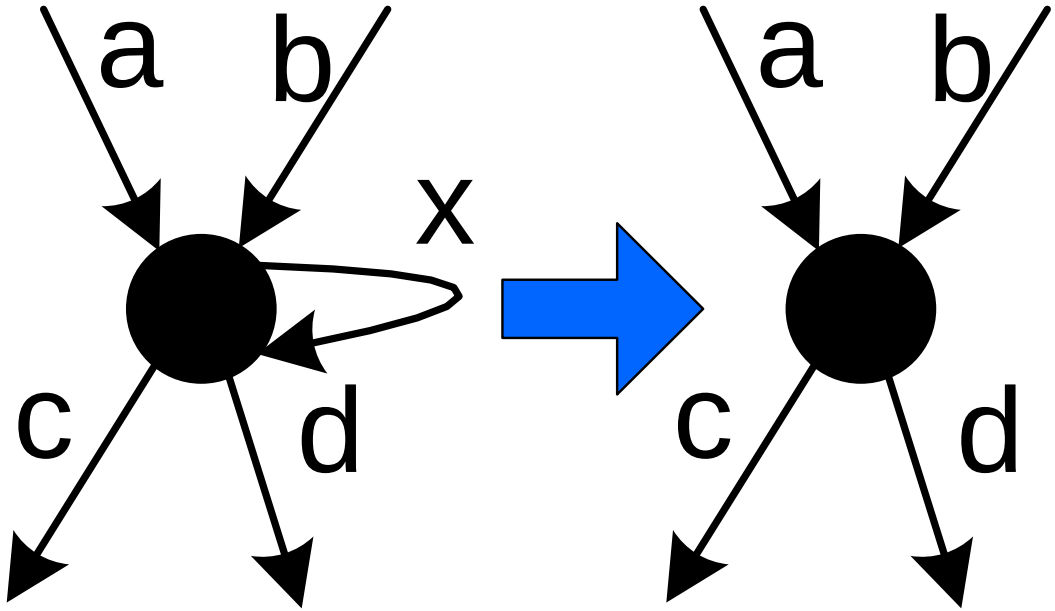




Extensions and variations

- Events may have other attributes that can be used:
  - States based on resources or data elements rather than activities.
  - Transition names based on resources.
- Filtering of infrequent paths.
- Filtering of infrequent activities.

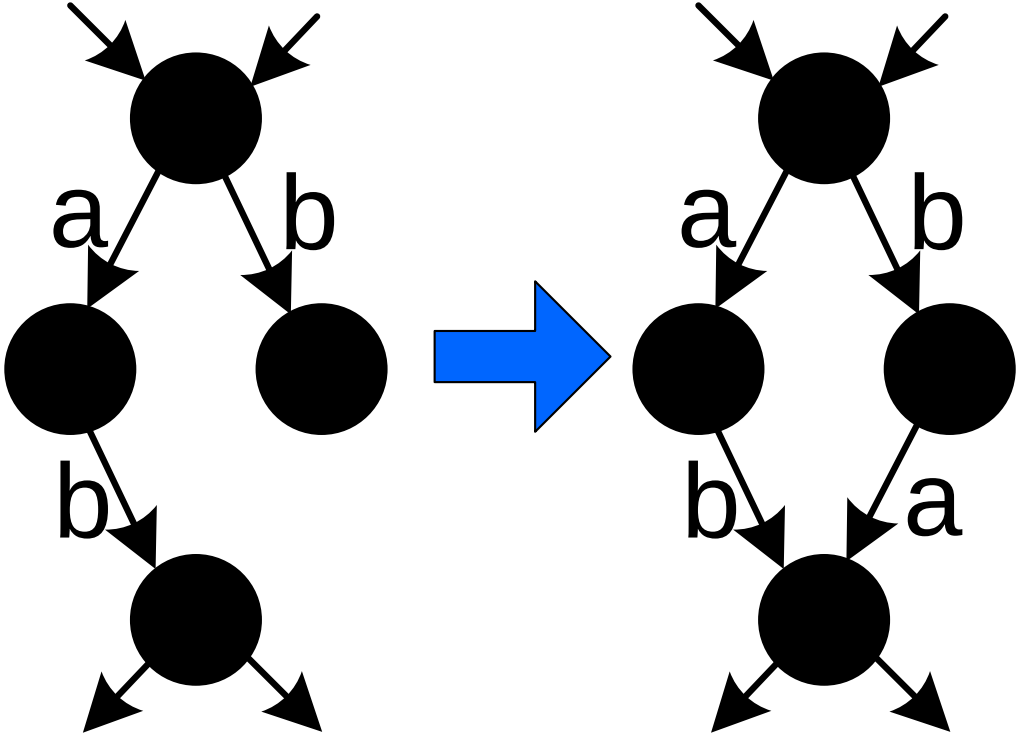
Postprocessing of transition system



removing self loops

(will jeopardize fitness)

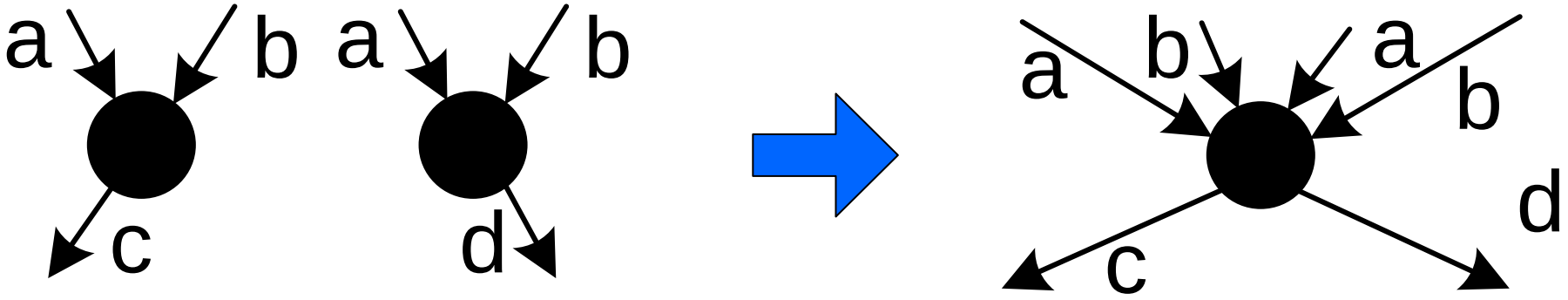
Postprocessing of transition system



improve diamond structure

(for missing interleavings)

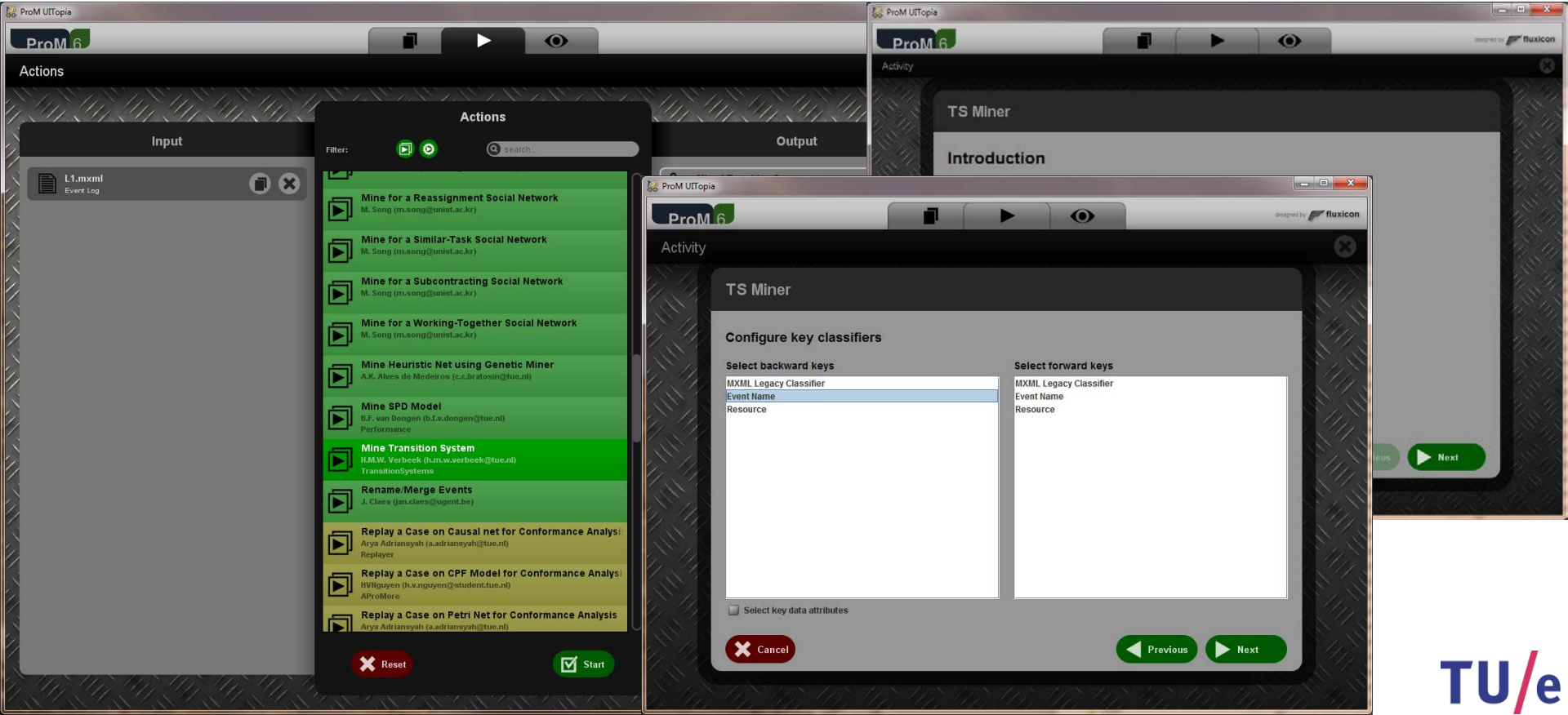
Postprocessing of transition system



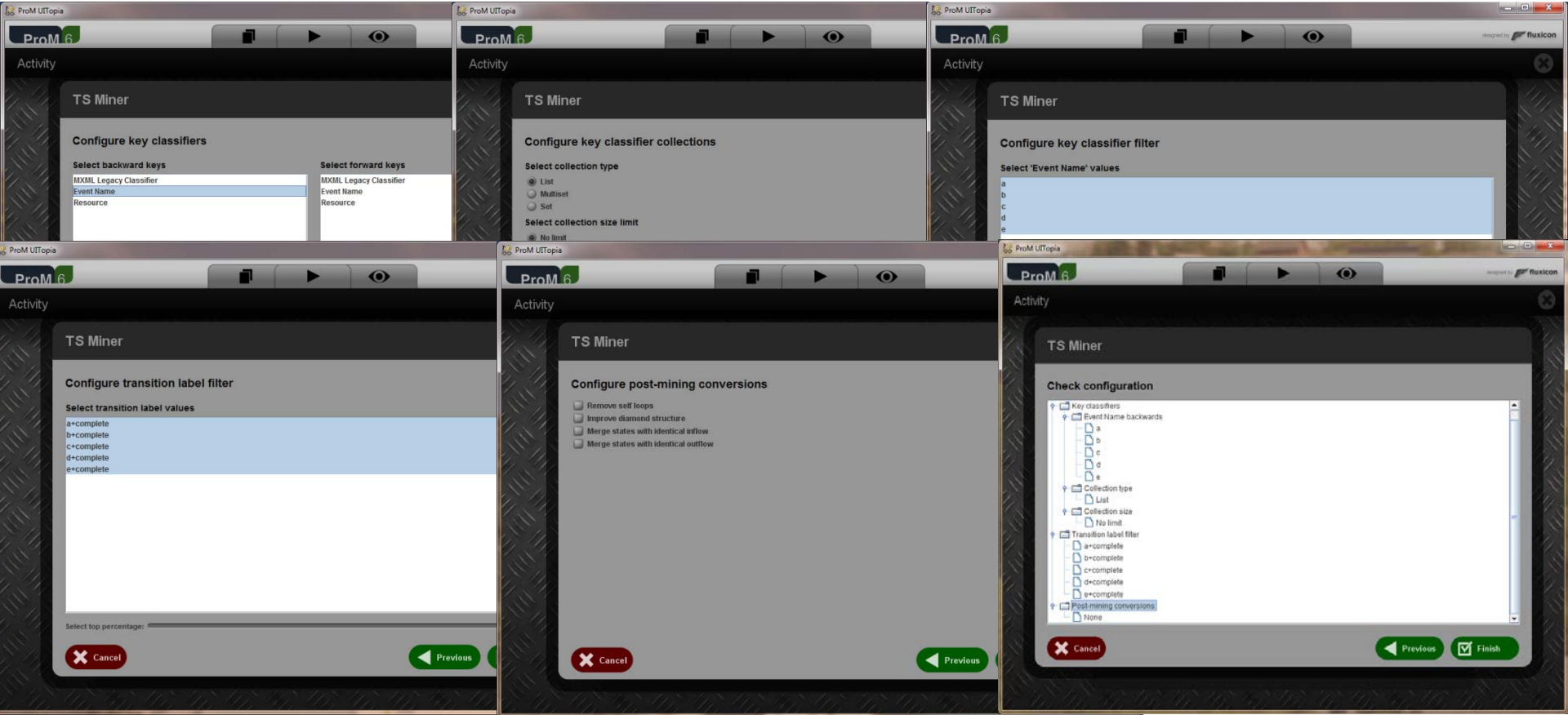
merge similar states

(e.g., based on inputs)

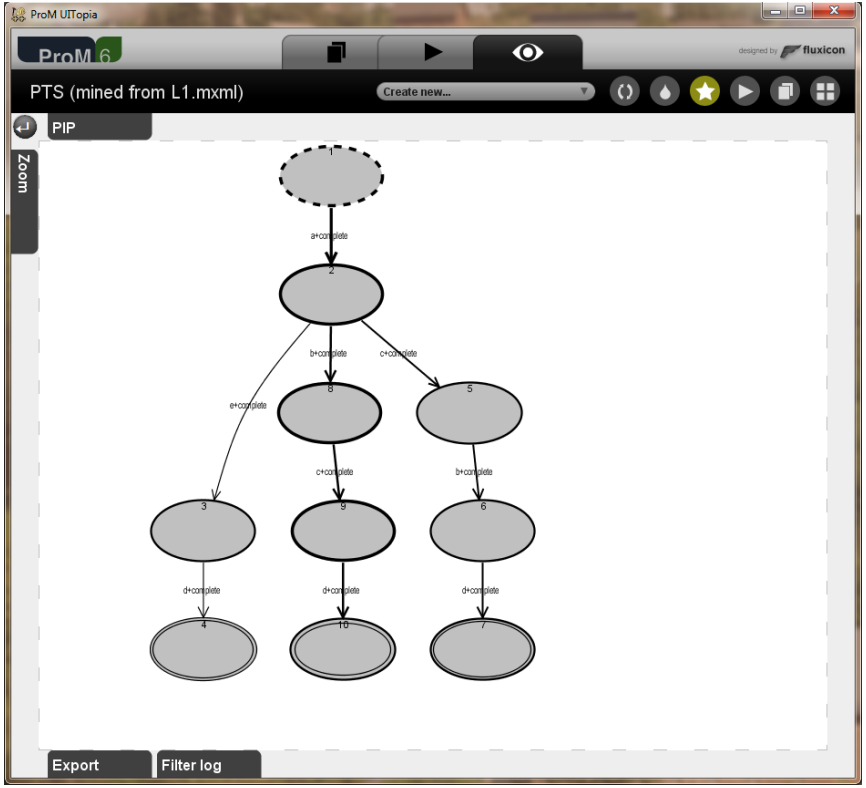
Mine Transition System plug-in of ProM

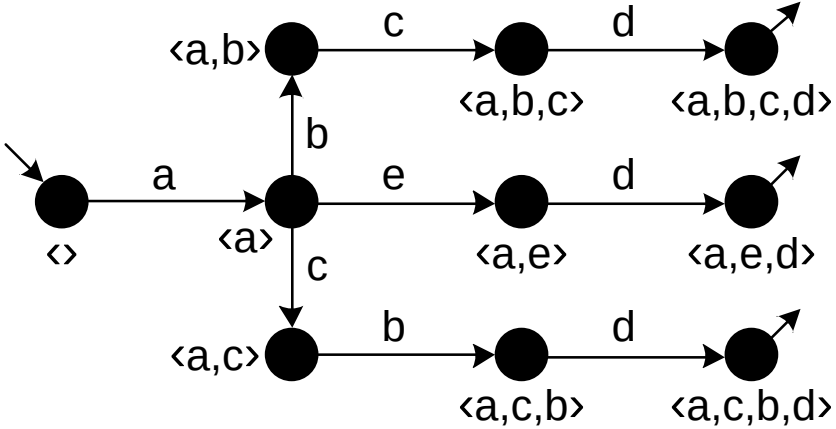


Highly configurable

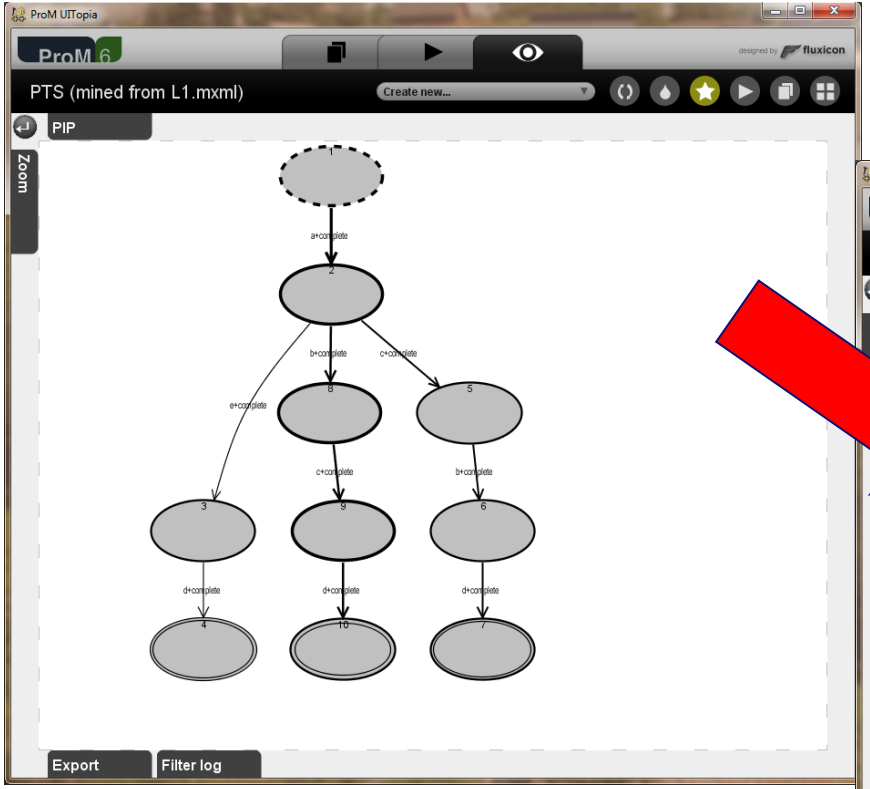


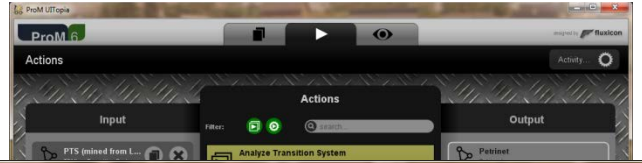
Output

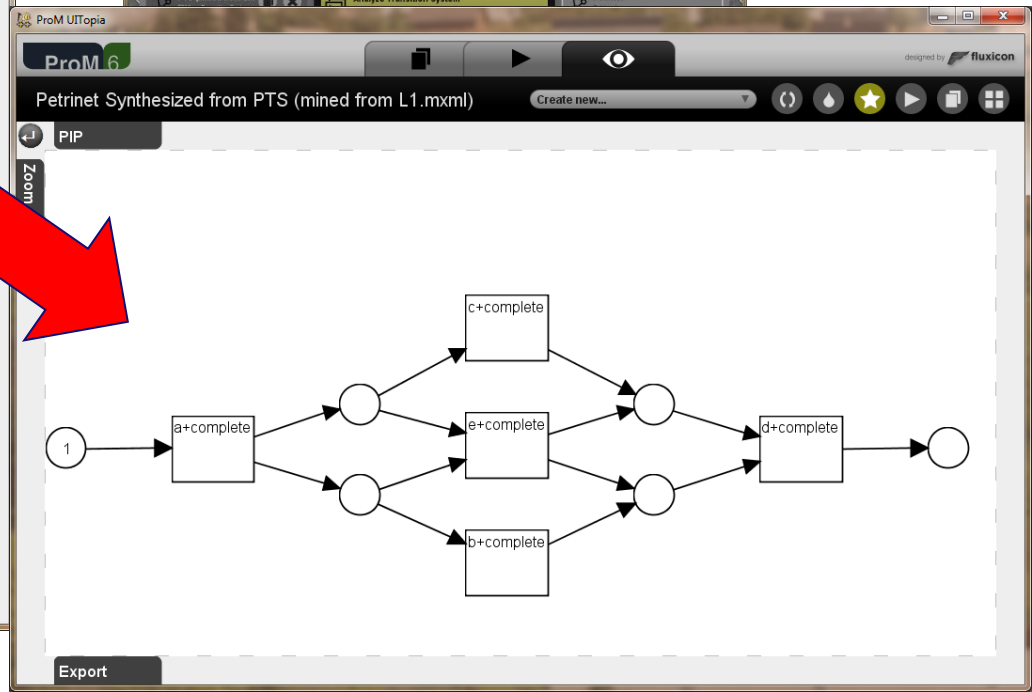




Next: state-based regions







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