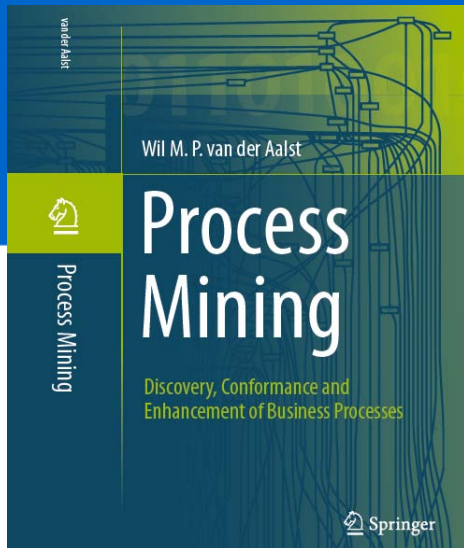


Chapter 5

Process Discovery: An Introduction

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



TU/e Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Overview

Chapter 1
Introduction

Part I: Preliminaries

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”

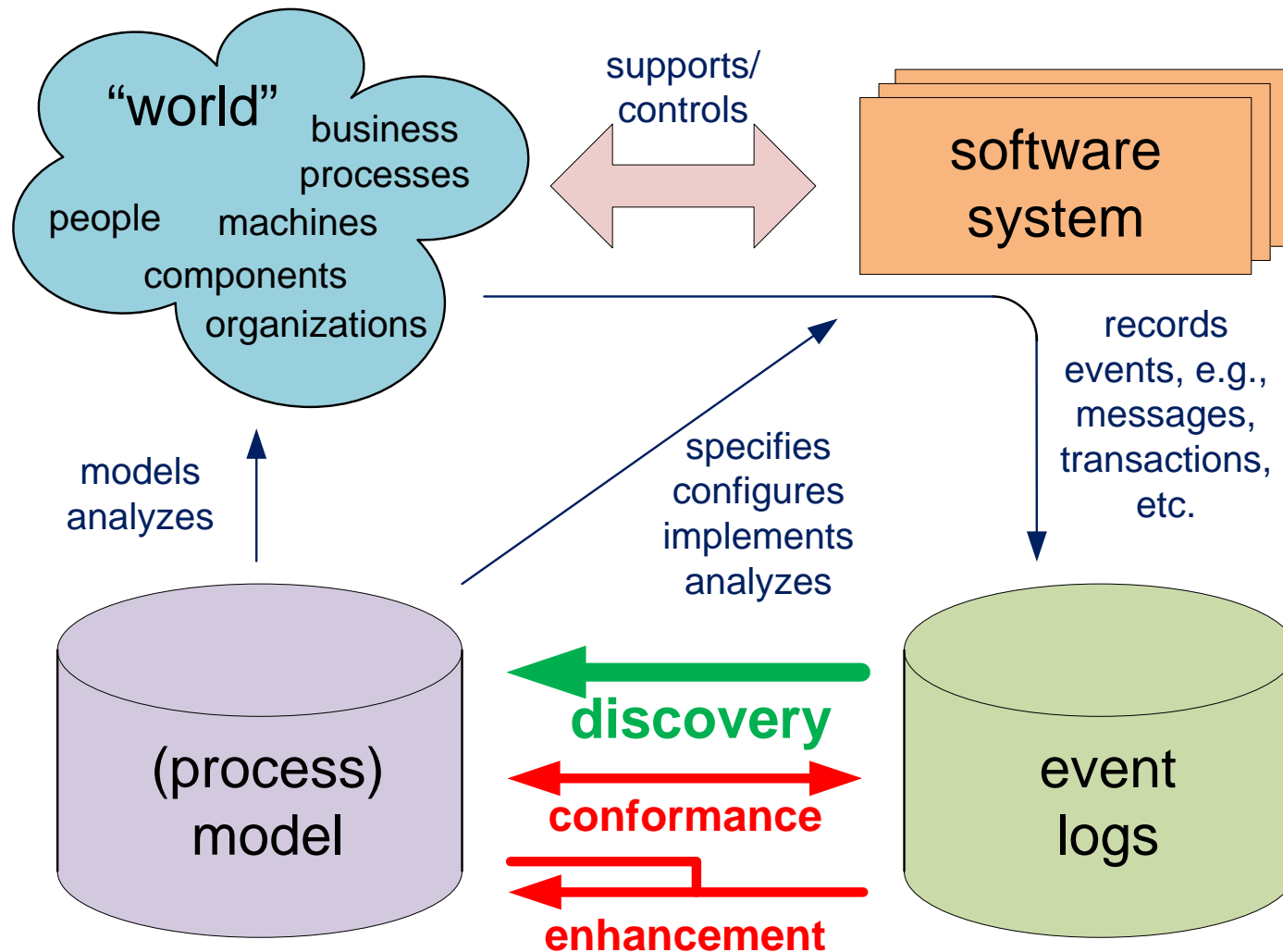
Chapter 12
Analyzing “Spaghetti
Processes”

Part V: Reflection

Chapter 13
Cartography and
Navigation

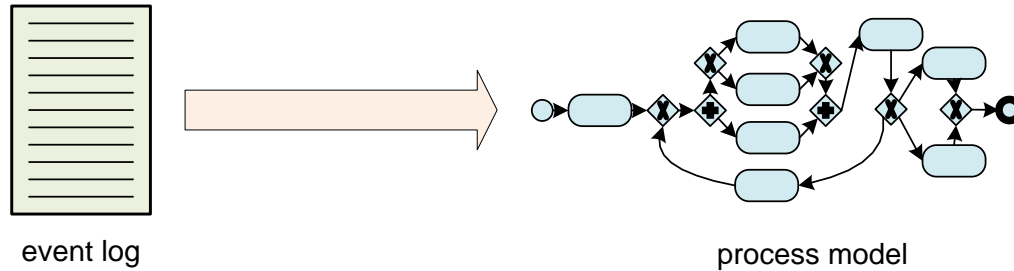
Chapter 14
Epilogue

Process discovery

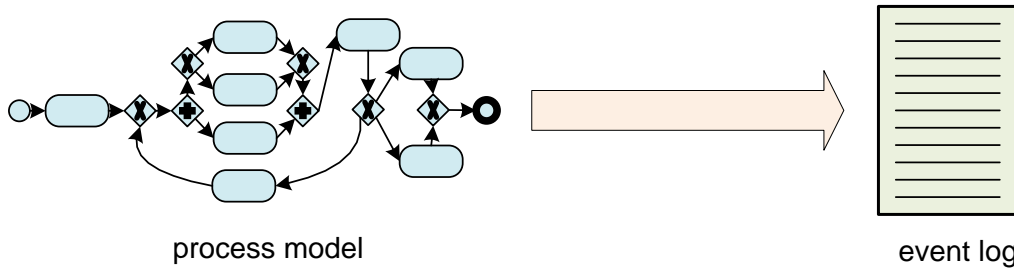


Process discovery = Play-In

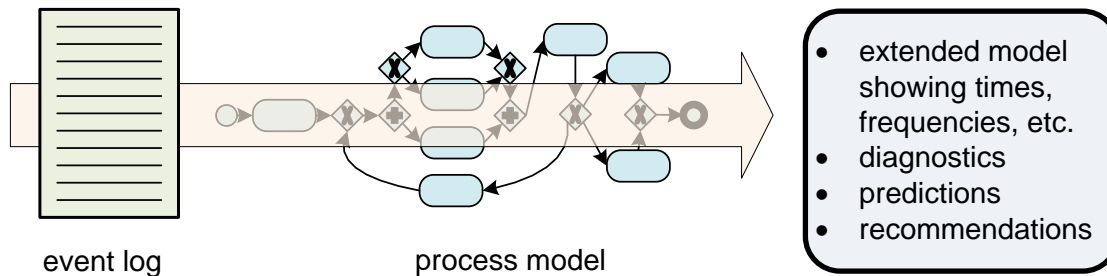
Play-In



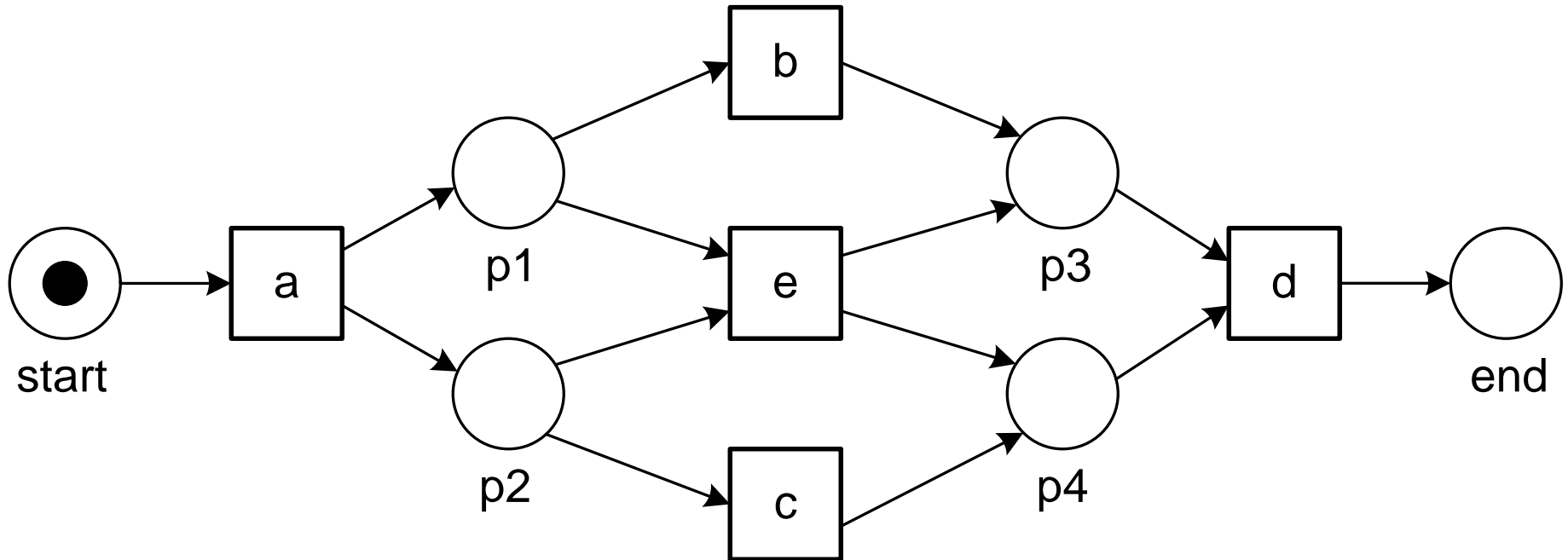
Play-Out



Replay



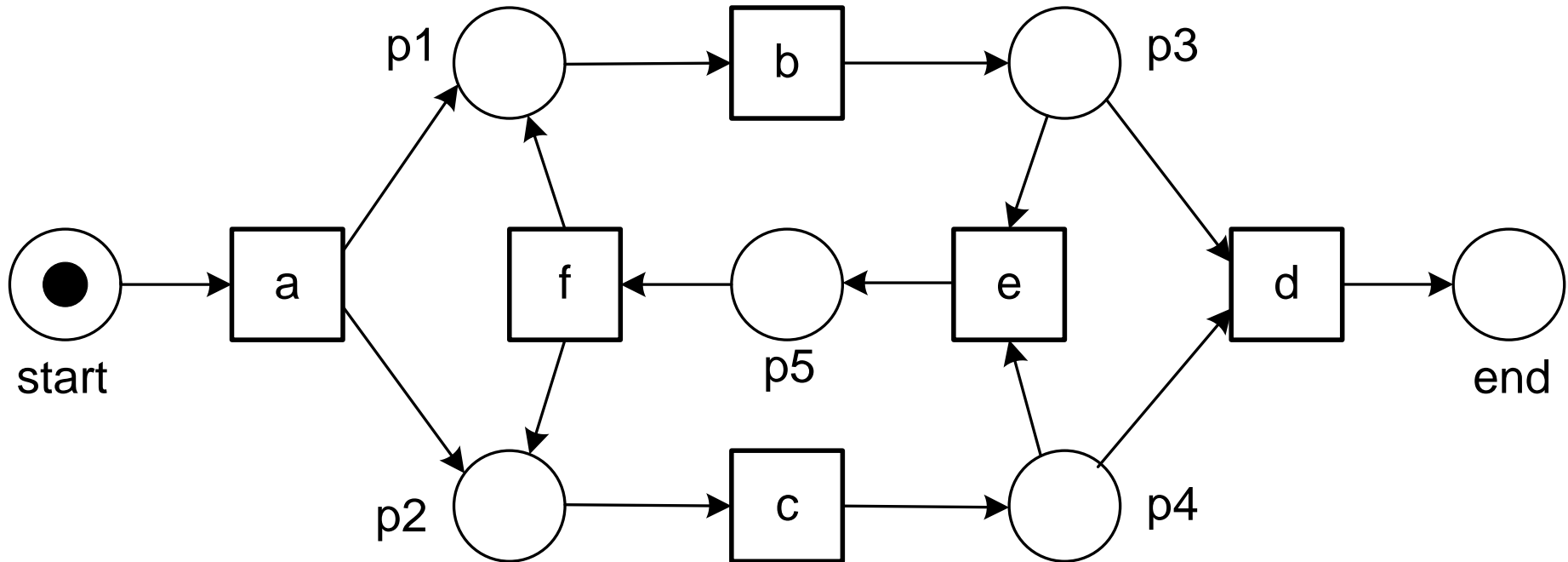
Example



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

Event log contains all possible traces of model and vice versa.

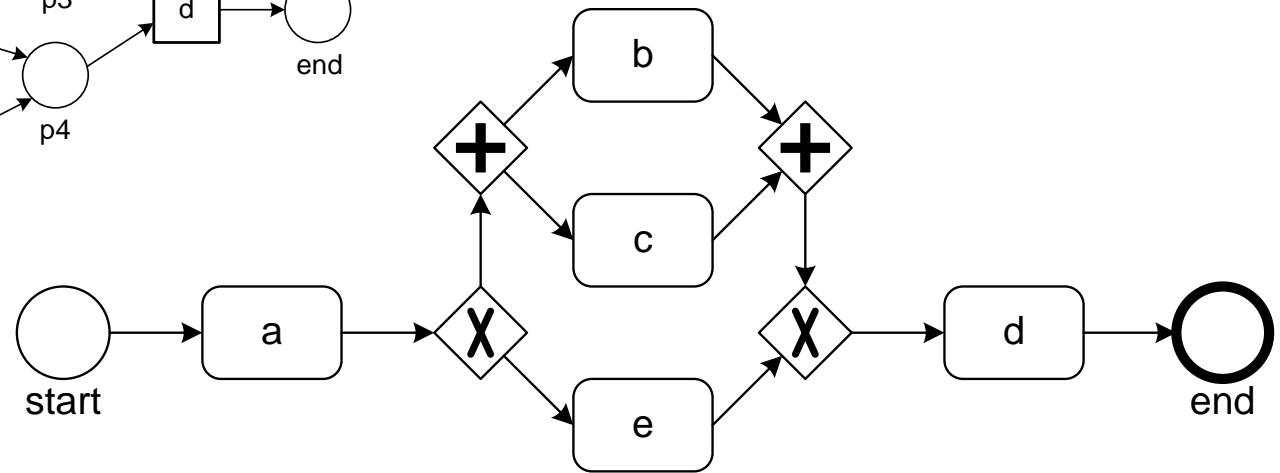
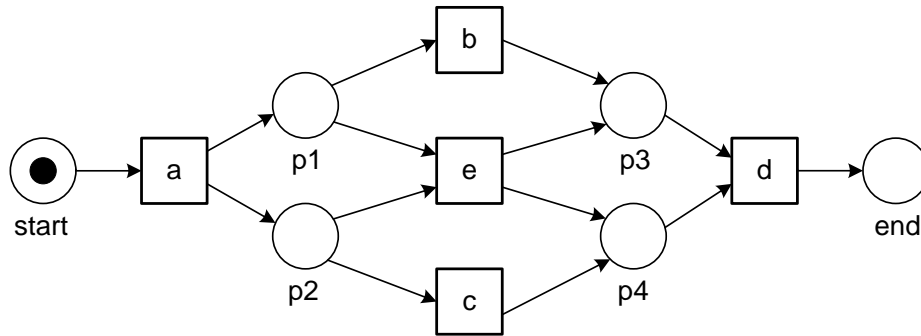
Another example



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

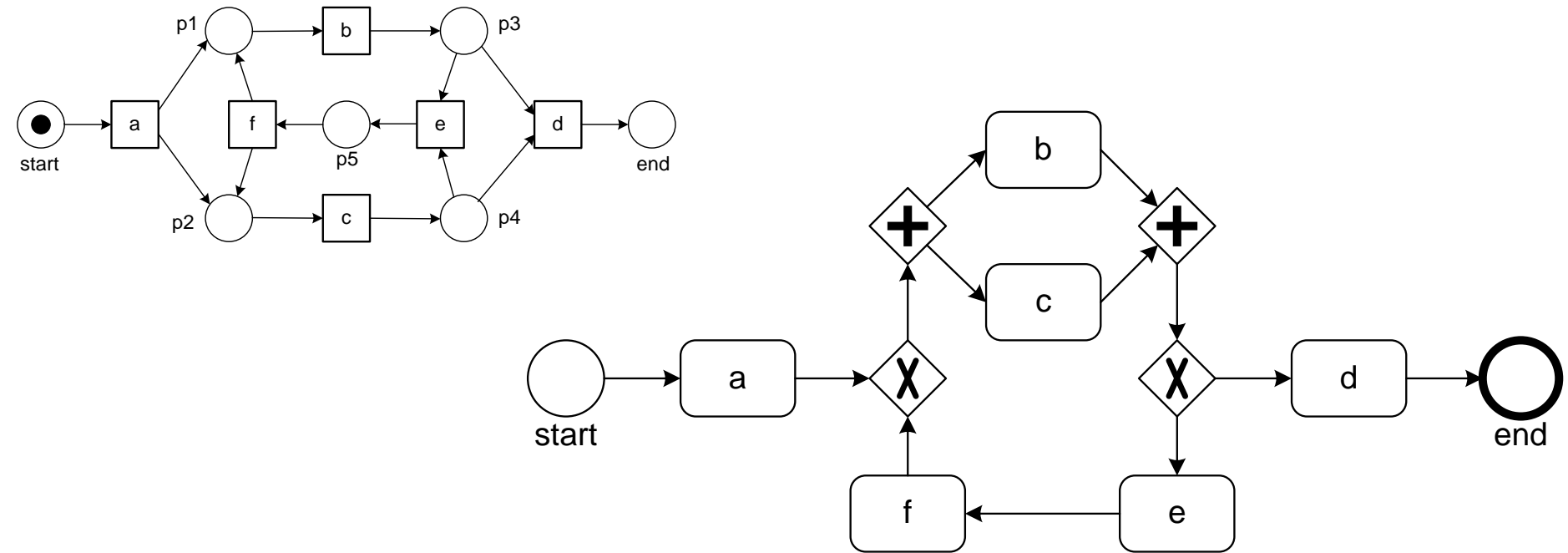
Generalization: event log contains only subset of all possible traces of model.

Notation is less relevant (e.g. BPMN)



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

Another BPMN example



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

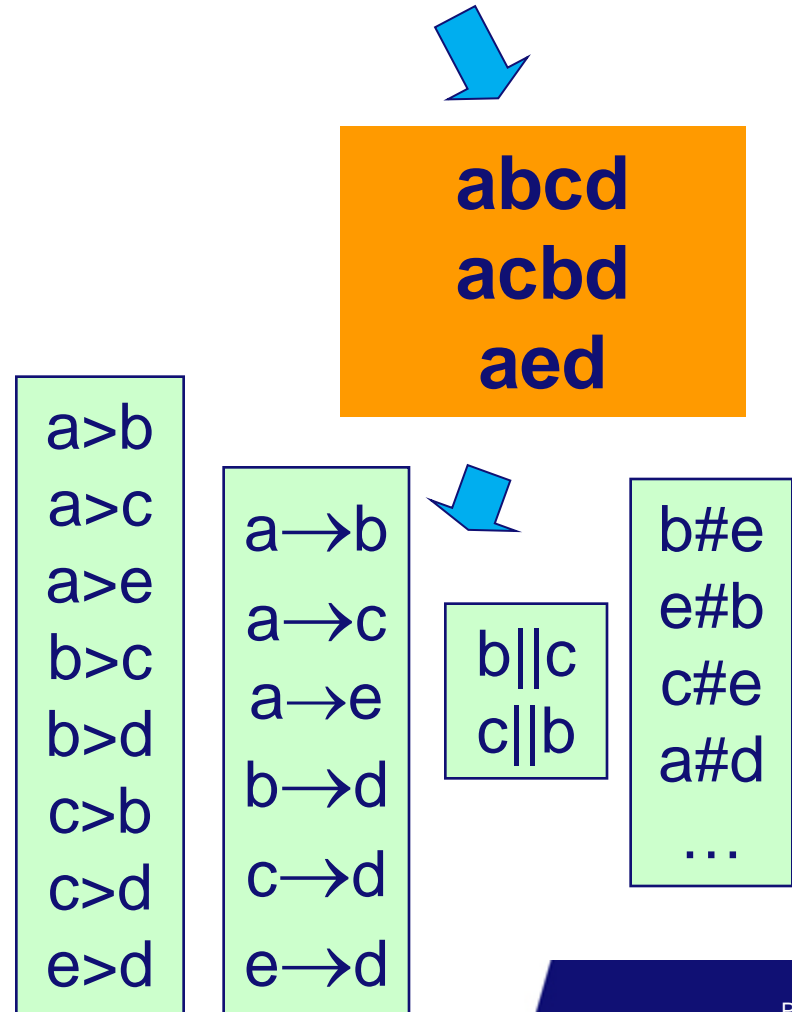
Challenge

- In general, there is a trade-off between the following four quality criteria:
 1. **Fitness**: the discovered model should allow for the behavior seen in the event log.
 2. **Precision (avoid underfitting)**: the discovered model should not allow for behavior completely unrelated to what was seen in the event log.
 3. **Generalization (avoid overfitting)**: the discovered model should generalize the example behavior seen in the event log.
 4. **Simplicity**: the discovered model should be as simple as possible.

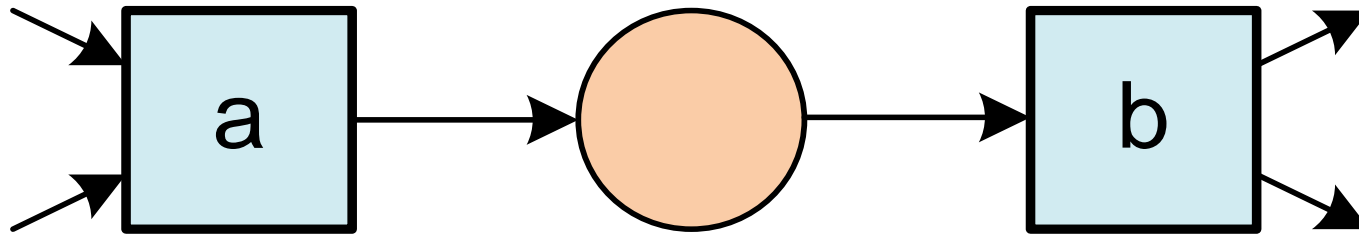
>, →, ||, # relations

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

- Direct succession: $x > y$ iff for some case x is directly followed by y .
- Causality: $x \rightarrow y$ iff $x > y$ and not $y > x$.
- Parallel: $x || y$ iff $x > y$ and $y > x$
- Choice: $x \# y$ iff not $x > y$ and not $y > x$.

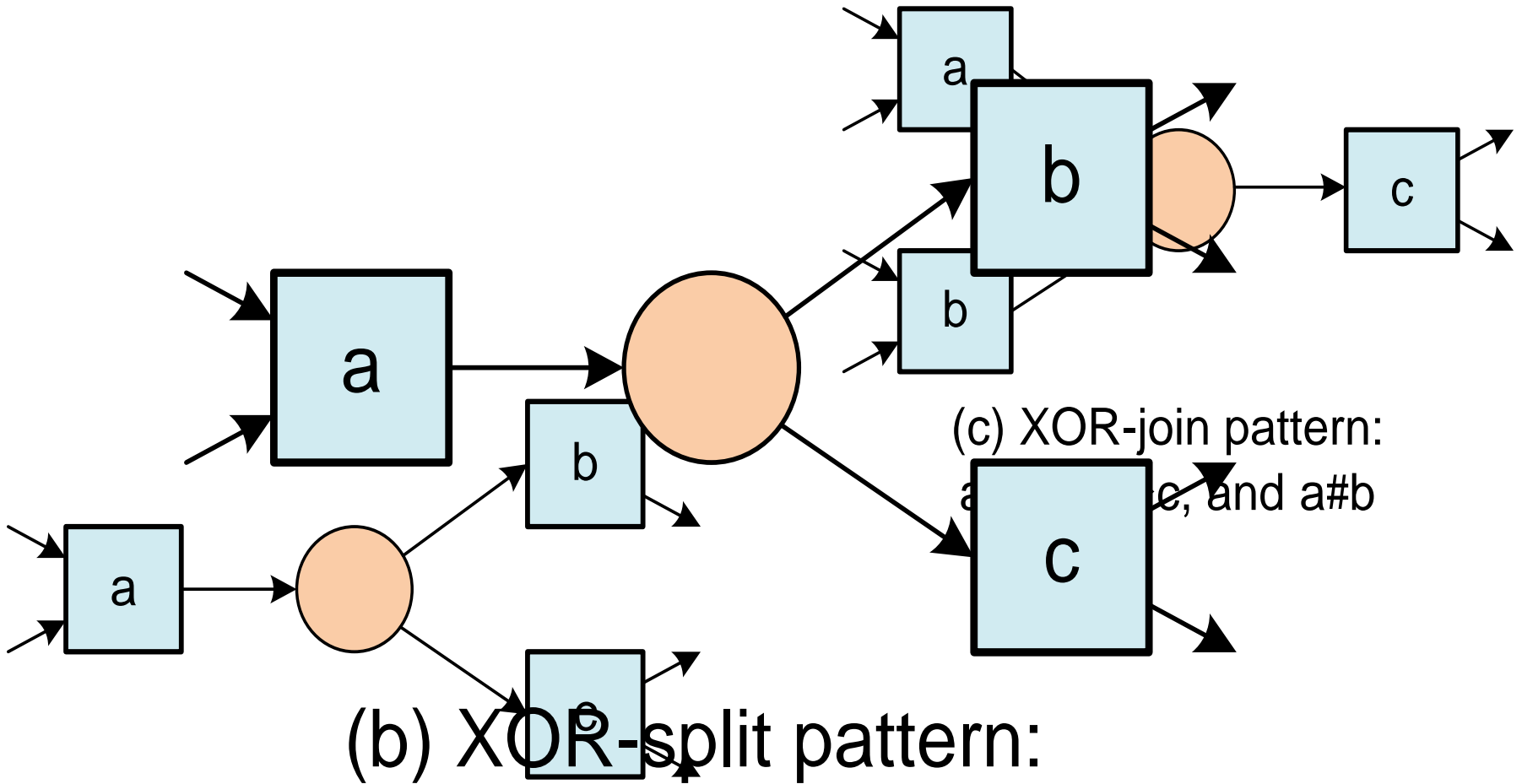


Basic Idea Used by α Algorithm (1)



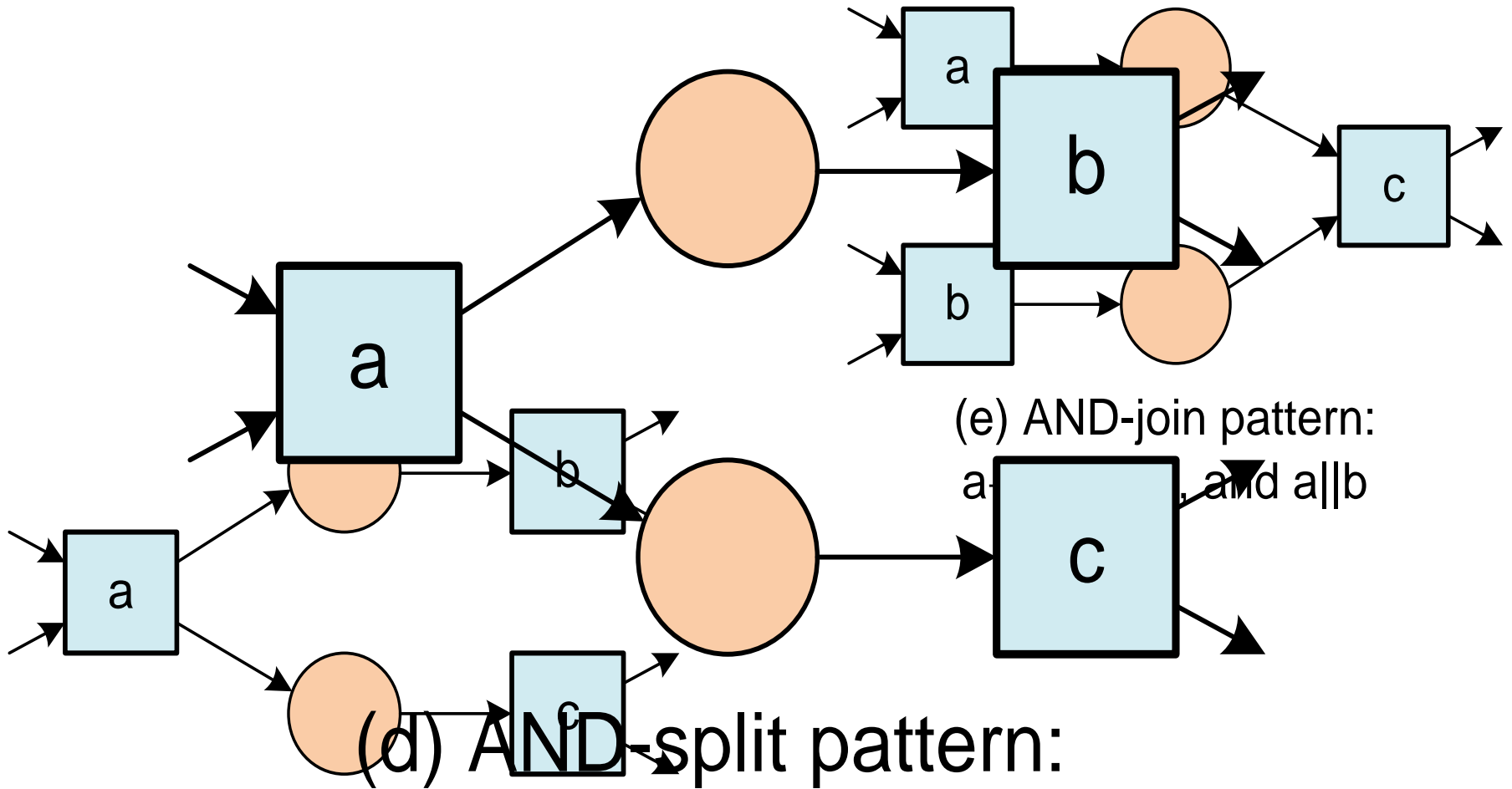
(a) sequence pattern: $a \rightarrow b$

Basic Idea Used by α Algorithm (2)



(b) XOR-split pattern:
 $a \rightarrow b$, $a \rightarrow c$, and $b \# c$

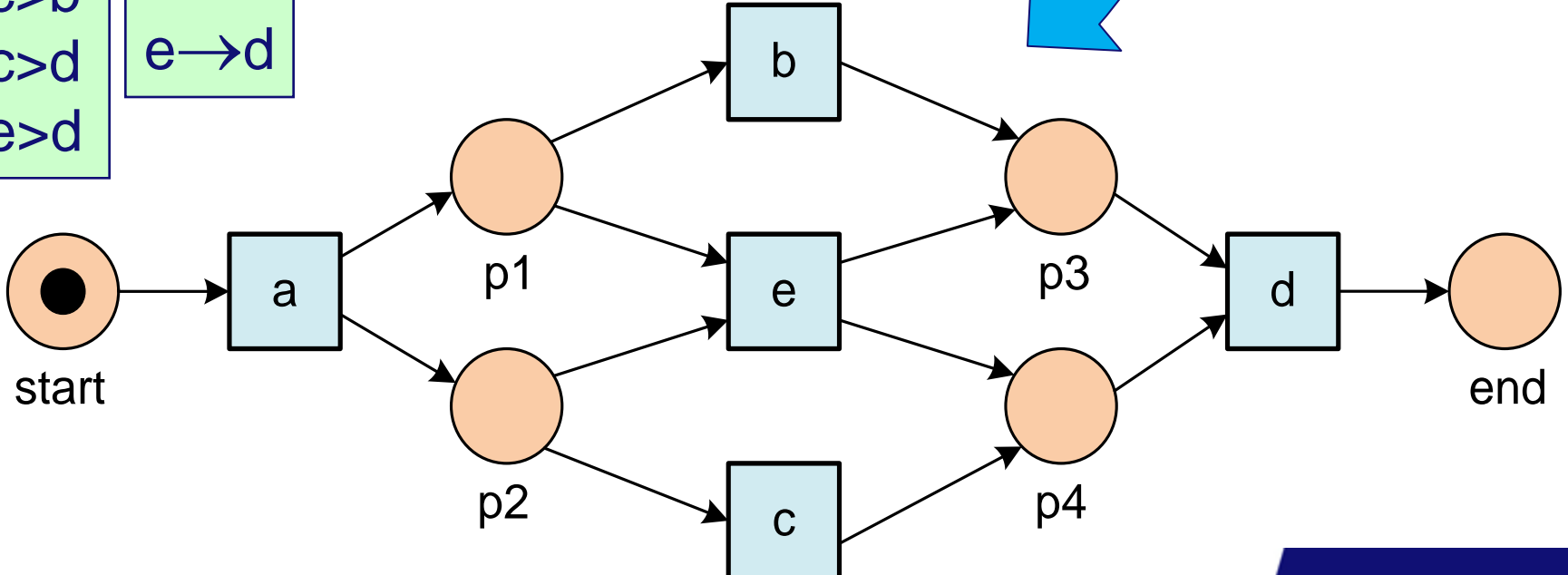
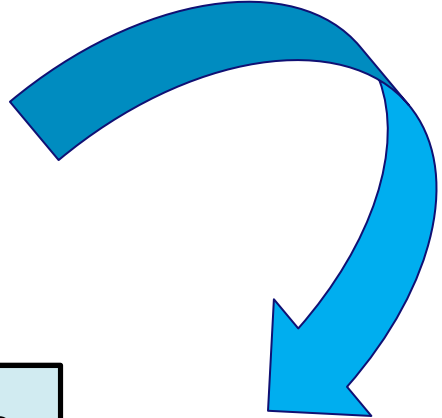
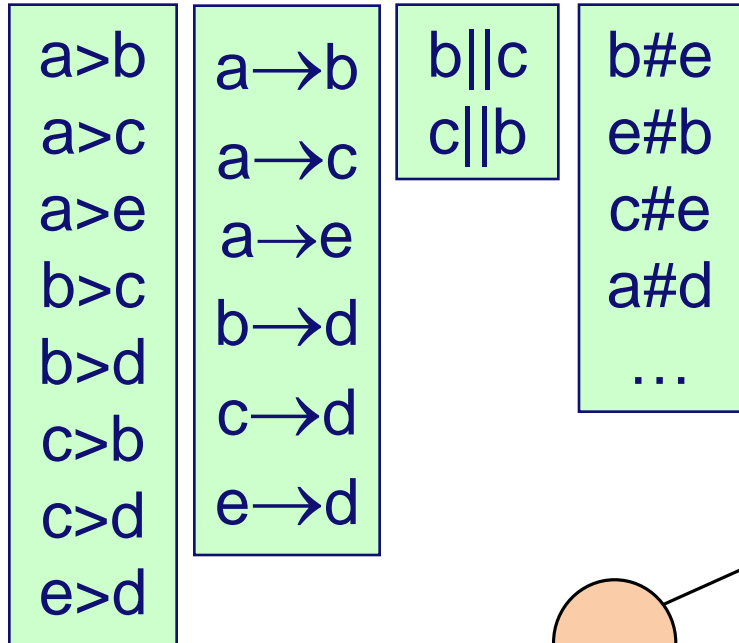
Basic Idea Used by α Algorithm (3)



(d) AND-split pattern:
 $a \rightarrow b$, $a \rightarrow c$, and $b \parallel c$

Example Revisited

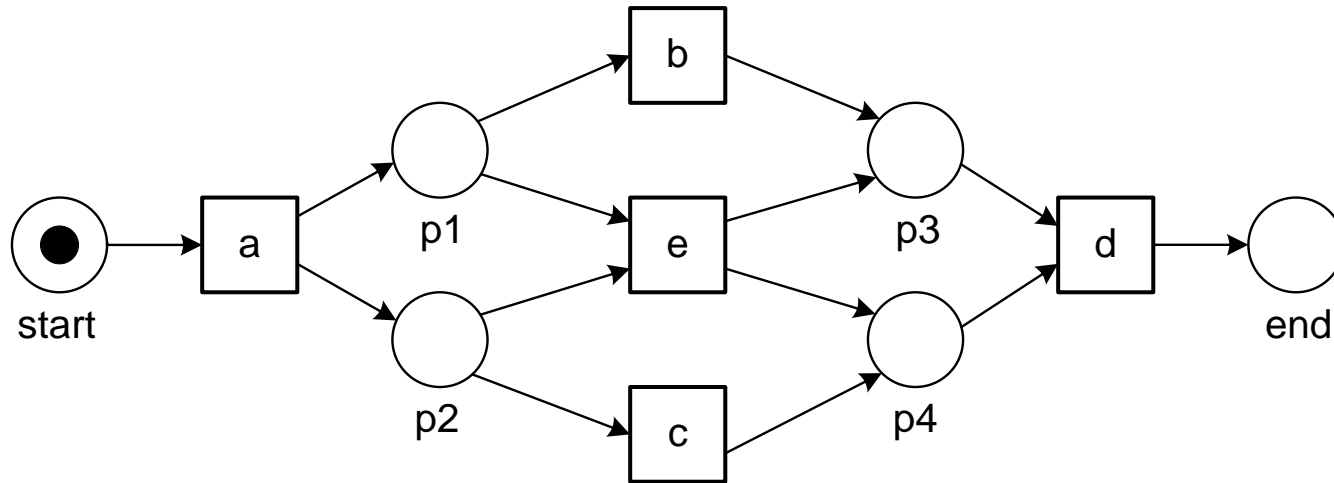
$$L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle]$$



Result produced by α algorithm

Footprint of L_1

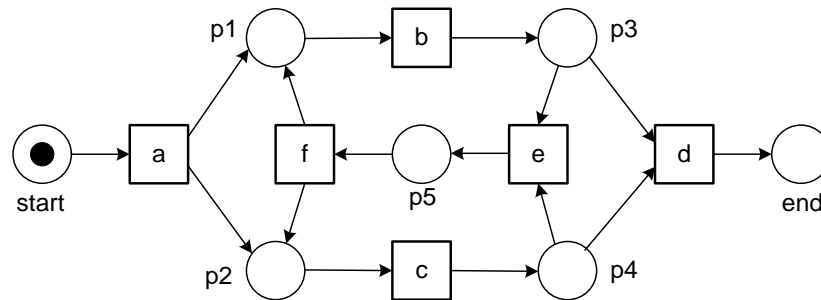
$$L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle]$$



	a	b	c	d	e
a	$\#_{L_1}$	\rightarrow_{L_1}	\rightarrow_{L_1}	$\#_{L_1}$	\rightarrow_{L_1}
b	\leftarrow_{L_1}	$\#_{L_1}$	\parallel_{L_1}	\rightarrow_{L_1}	$\#_{L_1}$
c	\leftarrow_{L_1}	\parallel_{L_1}	$\#_{L_1}$	\rightarrow_{L_1}	$\#_{L_1}$
d	$\#_{L_1}$	\leftarrow_{L_1}	\leftarrow_{L_1}	$\#_{L_1}$	\leftarrow_{L_1}
e	\leftarrow_{L_1}	$\#_{L_1}$	$\#_{L_1}$	\rightarrow_{L_1}	$\#_{L_1}$

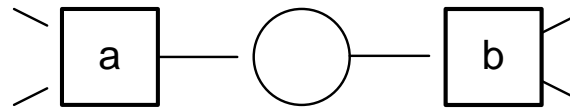
Footprint of L_2

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

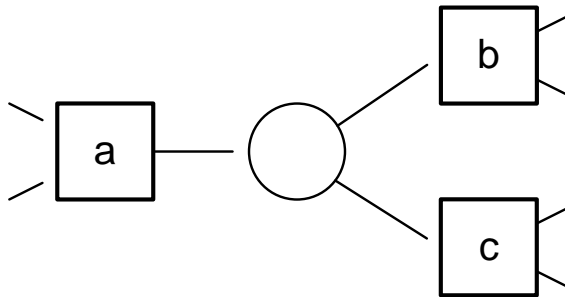


	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
<i>a</i>	#	→	→	#	#	#
<i>b</i>	←	#		→	→	←
<i>c</i>	←		#	→	→	←
<i>d</i>	#	←	←	#	#	#
<i>e</i>	#	←	←	#	#	→
<i>f</i>	#	→	→	#	←	#

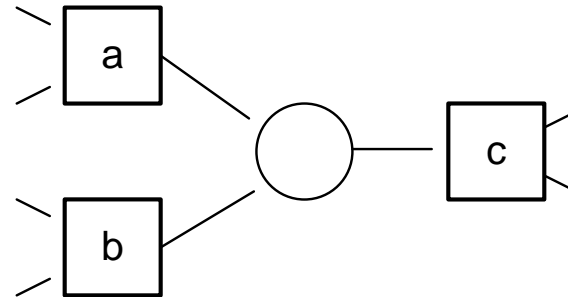
Simple patterns



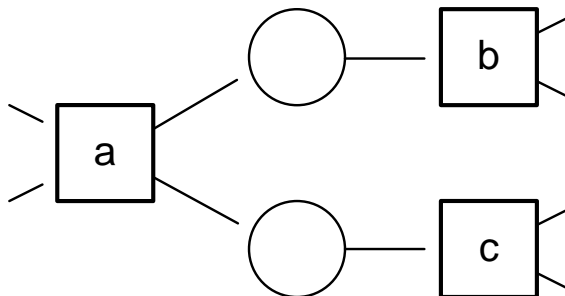
(a) sequence pattern: $a \rightarrow b$



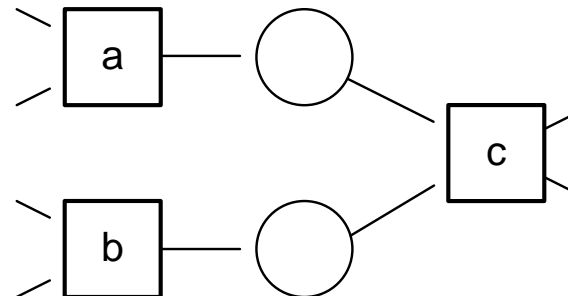
(b) XOR-split pattern:
 $a \rightarrow b$, $a \rightarrow c$, and $b \# c$



(c) XOR-join pattern:
 $a \rightarrow c$, $b \rightarrow c$, and $a \# b$



(d) AND-split pattern:
 $a \rightarrow b$, $a \rightarrow c$, and $b || c$



(e) AND-join pattern:
 $a \rightarrow c$, $b \rightarrow c$, and $a || b$

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. $T_I = \{ t \in T \mid \exists \sigma \in L \ t = \text{first}(\sigma) \},$

3. $T_O = \{ t \in T \mid \exists \sigma \in L \ t = \text{last}(\sigma) \},$

4. $X_L = \{ (A,B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge$
 $\forall_{a \in A} \forall_{b \in B} a \rightarrow_L b \wedge \forall_{a_1, a_2 \in A} a_1 \#_L a_2 \wedge \forall_{b_1, b_2 \in B} b_1 \#_L b_2 \},$

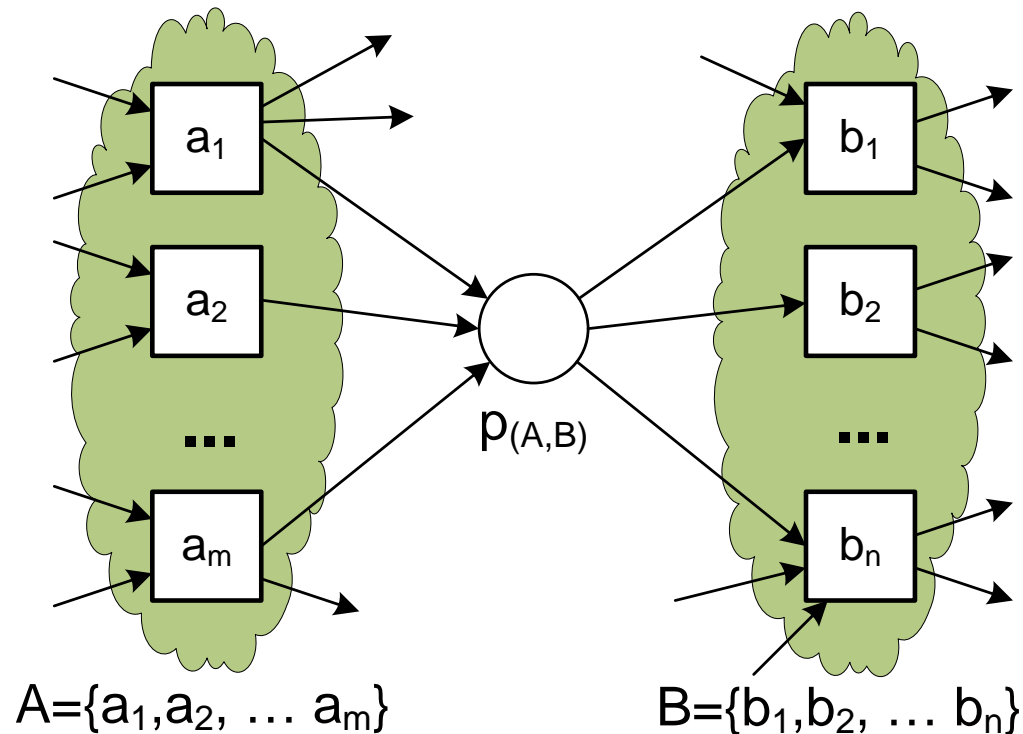
5. $Y_L = \{ (A,B) \in X_L \mid \forall_{(A',B') \in X_L} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \},$

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

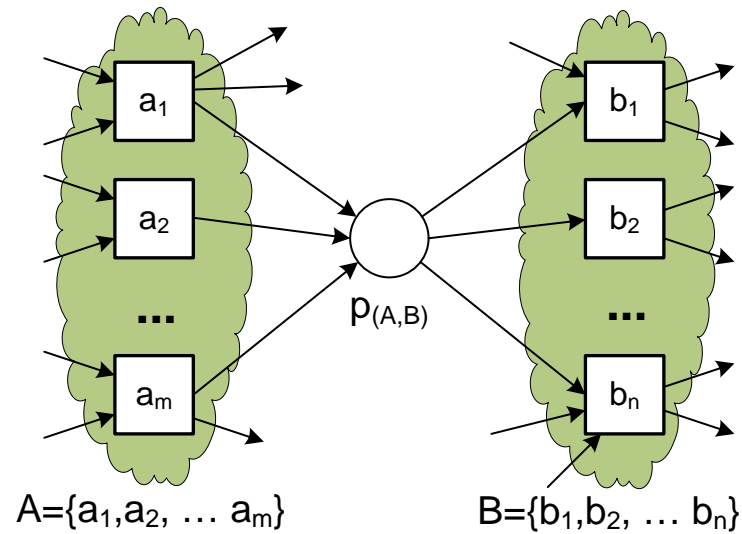
8. $\alpha(L) = (P_L, T_L, F_L).$

Key idea: find places



4. $X_L = \{ (A,B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall a \in A \forall b \in B a \rightarrow_L b \wedge \forall a_1, a_2 \in A a_1 \#_L a_2 \wedge \forall b_1, b_2 \in B b_1 \#_L b_2 \}$,
5. $Y_L = \{ (A,B) \in X_L \mid \forall (A',B') \in X_L A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,

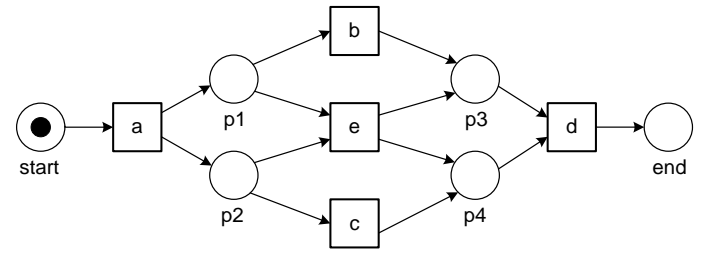
Places as footprints



	a_1	a_2	...	a_m	b_1	b_2	...	b_n
a_1	#	#	...	#	→	→	...	→
a_2	#	#	...	#	→	→	...	→
...
a_m	#	#	...	#	→	→	...	→
b_1	←	←	...	←	#	#	...	#
b_2	←	←	...	←	#	#	...	#
...
b_n	←	←	...	←	#	#	...	#

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

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>a</i>	# L_1	\rightarrow_{L_1}	\rightarrow_{L_1}	# L_1	\rightarrow_{L_1}
<i>b</i>	\leftarrow_{L_1}	# L_1	\parallel_{L_1}	\rightarrow_{L_1}	# L_1
<i>c</i>	\leftarrow_{L_1}	\parallel_{L_1}	# L_1	\rightarrow_{L_1}	# L_1
<i>d</i>	# L_1	\leftarrow_{L_1}	\leftarrow_{L_1}	# L_1	\leftarrow_{L_1}
<i>e</i>	\leftarrow_{L_1}	# L_1	# L_1	\rightarrow_{L_1}	# L_1



$$X_{L_1} = \{(\{a\}, \{b\}), (\{a\}, \{c\}), (\{a\}, \{e\}), (\{a\}, \{b, e\}), (\{a\}, \{c, e\}), (\{b\}, \{d\}), (\{c\}, \{d\}), (\{e\}, \{d\}), (\{b, e\}, \{d\}), (\{c, e\}, \{d\})\}$$

$$Y_{L_1} = \{(\{a\}, \{b, e\}), (\{a\}, \{c, e\}), (\{b, e\}, \{d\}), (\{c, e\}, \{d\})\}$$

Another event log L_3

$$L_3 = [\langle a, b, c, d, e, f, b, d, c, e, g \rangle, \\ \langle a, b, d, c, e, g \rangle^2, \\ \langle a, b, c, d, e, f, b, c, d, e, f, b, d, c, e, g \rangle]$$

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
<i>a</i>	#	→	#	#	#	#	#
<i>b</i>	←	#	→	→	#	←	#
<i>c</i>	#	←	#		→	#	#
<i>d</i>	#	←		#	→	#	#
<i>e</i>	#	#	←	←	#	→	→
<i>f</i>	#	→	#	#	←	#	#
<i>g</i>	#	#	#	#	←	#	#

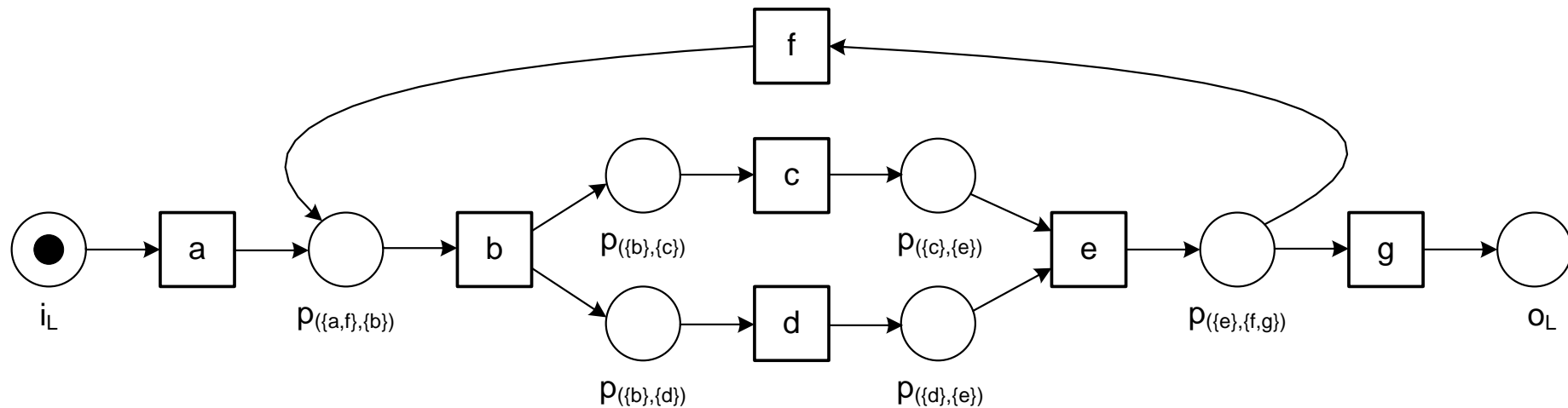
Model for L_3

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
<i>a</i>	#	→	#	#	#	#	#
<i>b</i>	←	#	→	→	#	←	#
<i>c</i>	#	←	#		→	#	#
<i>d</i>	#	←		#	→	#	#
<i>e</i>	#	#	←	←	#	→	→
<i>f</i>	#	→	#	#	←	#	#
<i>g</i>	#	#	#	#	←	#	#

$$L_3 = [\langle a, b, c, d, e, f, b, d, c, e, g \rangle,$$

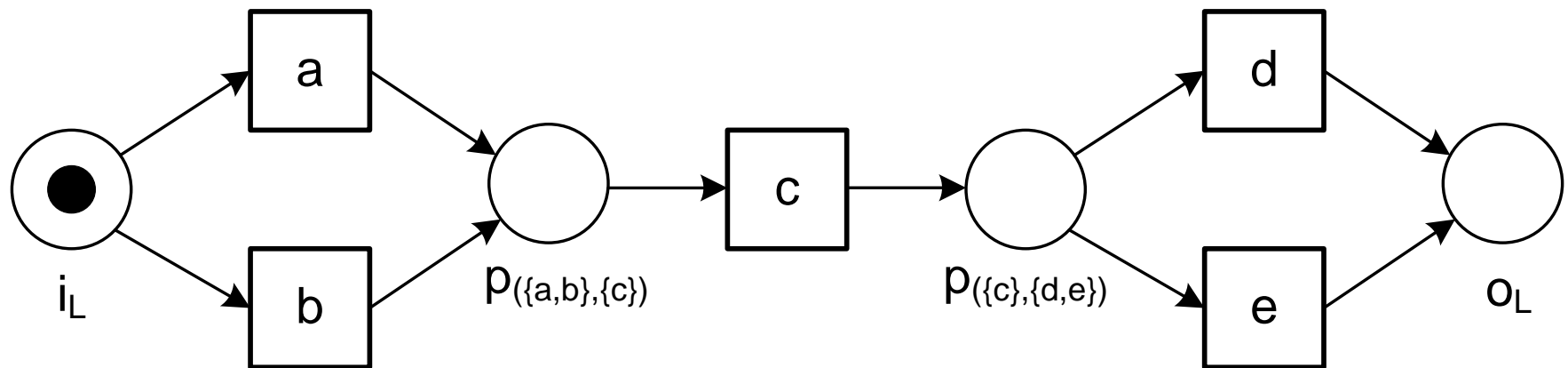
$$\langle a, b, d, c, e, g \rangle^2,$$

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



Another event log L_4

$$L_4 = [\langle a, c, d \rangle^{45}, \langle b, c, d \rangle^{42}, \langle a, c, e \rangle^{38}, \langle b, c, e \rangle^{22}]$$



Event log L_5

$$L_5 = [\langle a, b, e, f \rangle^2, \langle a, b, e, c, d, b, f \rangle^3, \langle a, b, c, e, d, b, f \rangle^2, \langle a, b, c, d, e, b, f \rangle^4, \langle a, e, b, c, d, b, f \rangle^3]$$

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
<i>a</i>	#	→	#	#	→	#
<i>b</i>	←	#	→	←		→
<i>c</i>	#	←	#	→		#
<i>d</i>	#	→	←	#		#
<i>e</i>	←				#	→
<i>f</i>	#	←	#	#	←	#

$$T_L = \{a, b, c, d, e, f\}$$

$$T_I = \{a\}$$

$$T_I = \{f\}$$

$$X_L = \{(\{a\}, \{b\}), (\{a\}, \{e\}), (\{b\}, \{c\}), (\{b\}, \{f\}), (\{c\}, \{d\}), \\ (\{d\}, \{b\}), (\{e\}, \{f\}), (\{a, d\}, \{b\}), (\{b\}, \{c, f\})\}$$

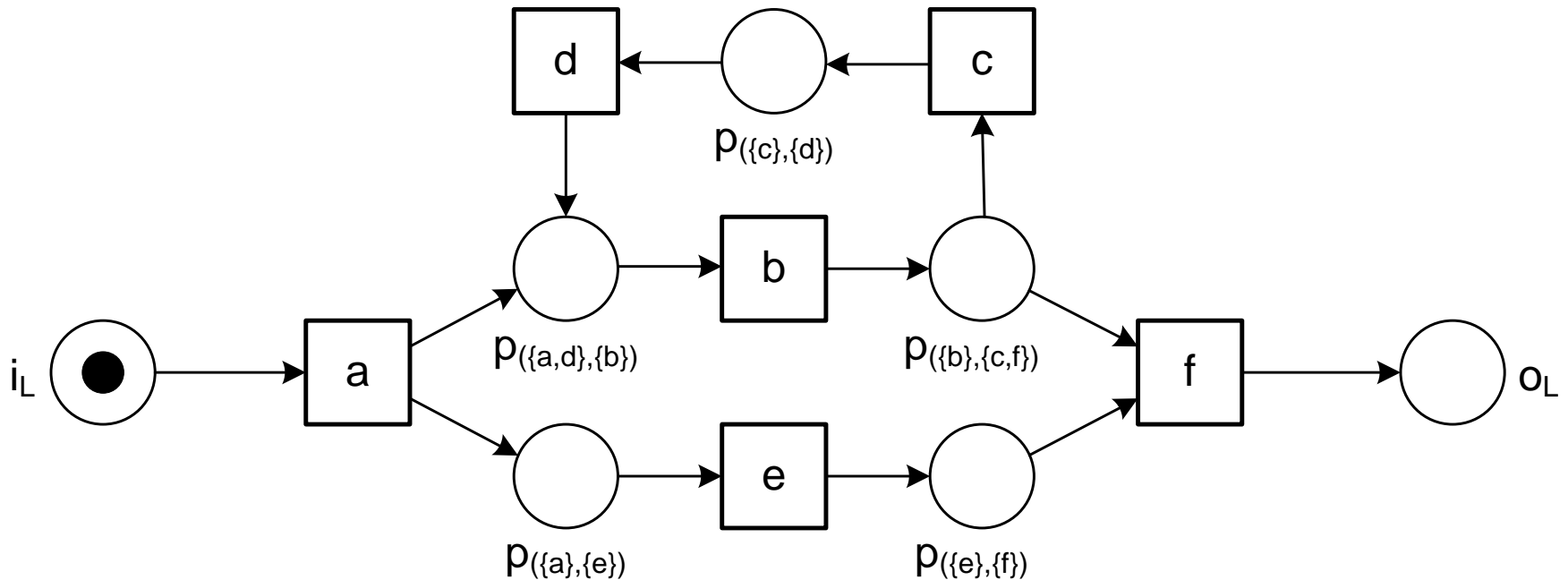
$$Y_L = \{(\{a\}, \{e\}), (\{c\}, \{d\}), (\{e\}, \{f\}), (\{a, d\}, \{b\}), (\{b\}, \{c, f\})\}$$

$$P_L = \{p(\{a\}, \{e\}), p(\{c\}, \{d\}), p(\{e\}, \{f\}), p(\{a, d\}, \{b\}), p(\{b\}, \{c, f\}), i_L, o_L\}$$

$$F_L = \{(a, p(\{a\}, \{e\})), (p(\{a\}, \{e\}), e), (c, p(\{c\}, \{d\})), (p(\{c\}, \{d\}), d), \\ (e, p(\{e\}, \{f\})), (p(\{e\}, \{f\}), f), (a, p(\{a, d\}, \{b\})), (d, p(\{a, d\}, \{b\})), \\ (p(\{a, d\}, \{b\}), b), (b, p(\{b\}, \{c, f\})), (p(\{b\}, \{c, f\}), c), (p(\{b\}, \{c, f\}), f), \\ (i_L, a), (f, o_L)\}$$

$$\alpha(L) = (P_L, T_L, F_L)$$

Discovered model

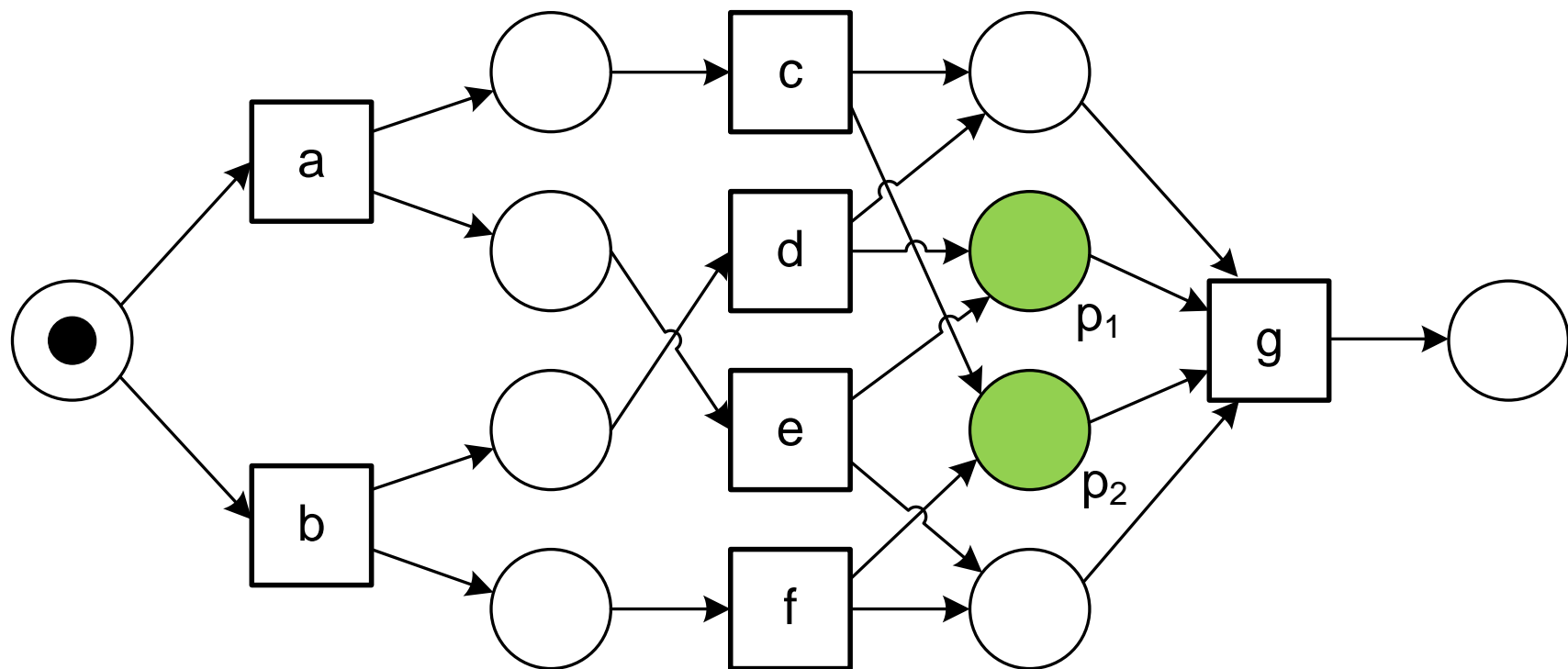


$$X_L = \{(\{a\}, \{b\}), (\{a\}, \{e\}), (\{b\}, \{c\}), (\{b\}, \{f\}), (\{c\}, \{d\}), (\{d\}, \{b\}), (\{e\}, \{f\}), (\{a, d\}, \{b\}), (\{b\}, \{c, f\})\}$$

$$Y_L = \{(\{a\}, \{e\}), (\{c\}, \{d\}), (\{e\}, \{f\}), (\{a, d\}, \{b\}), (\{b\}, \{c, f\})\}$$

Limitation of α algorithm (implicit places)

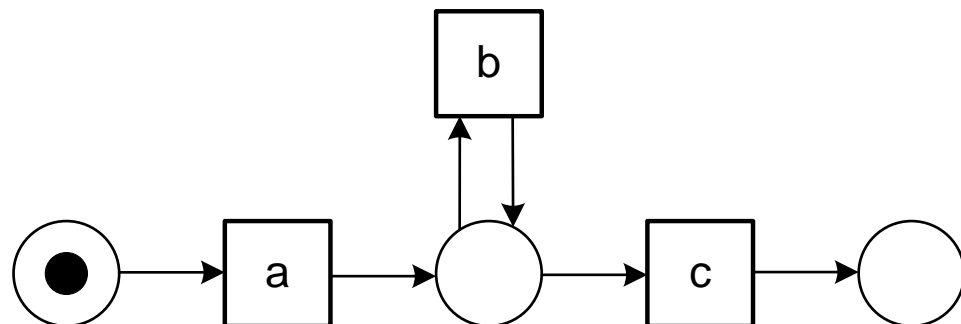
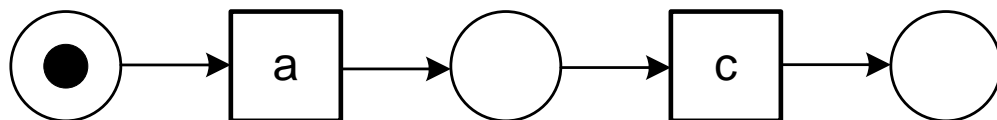
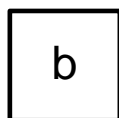
$$L_6 = [\langle a, c, e, g \rangle^2, \langle a, e, c, g \rangle^3, \langle b, d, f, g \rangle^2, \langle b, f, d, g \rangle^4]$$



Green places are implicit!

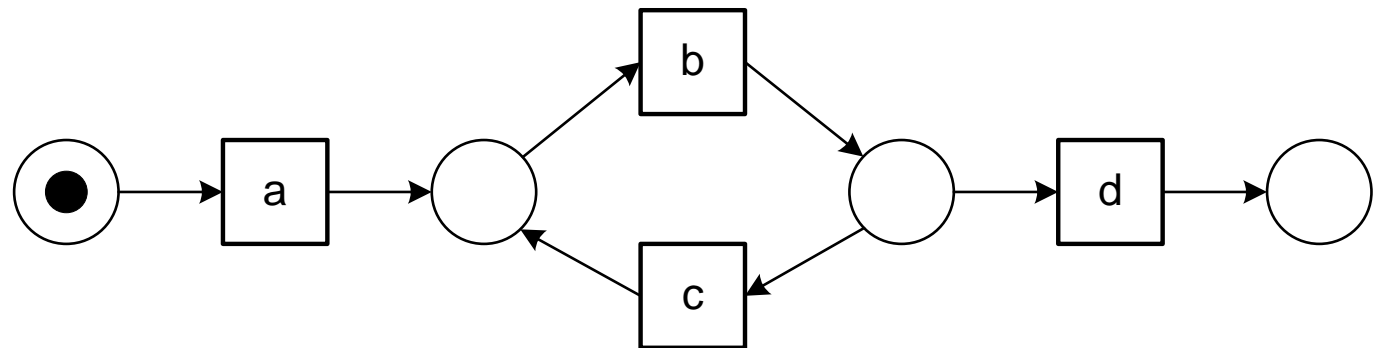
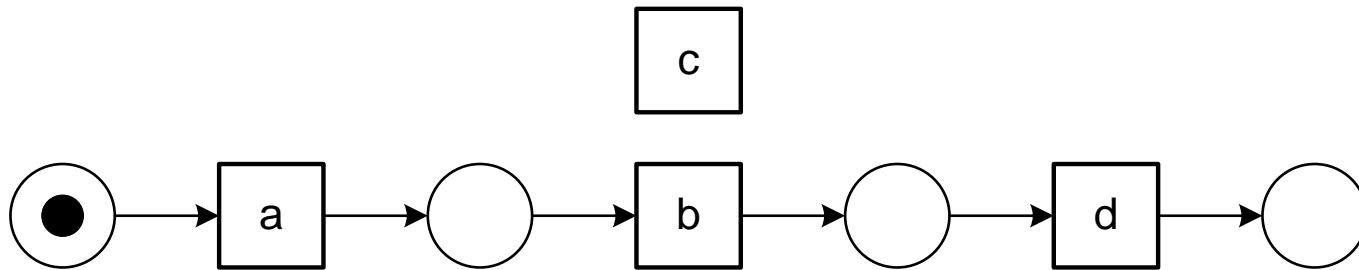
Limitation of α algorithm (loops of length 1)

$$L_7 = [\langle a, c \rangle^2, \langle a, b, c \rangle^3, \langle a, b, b, c \rangle^2, \langle a, b, b, b, b, c \rangle^1]$$



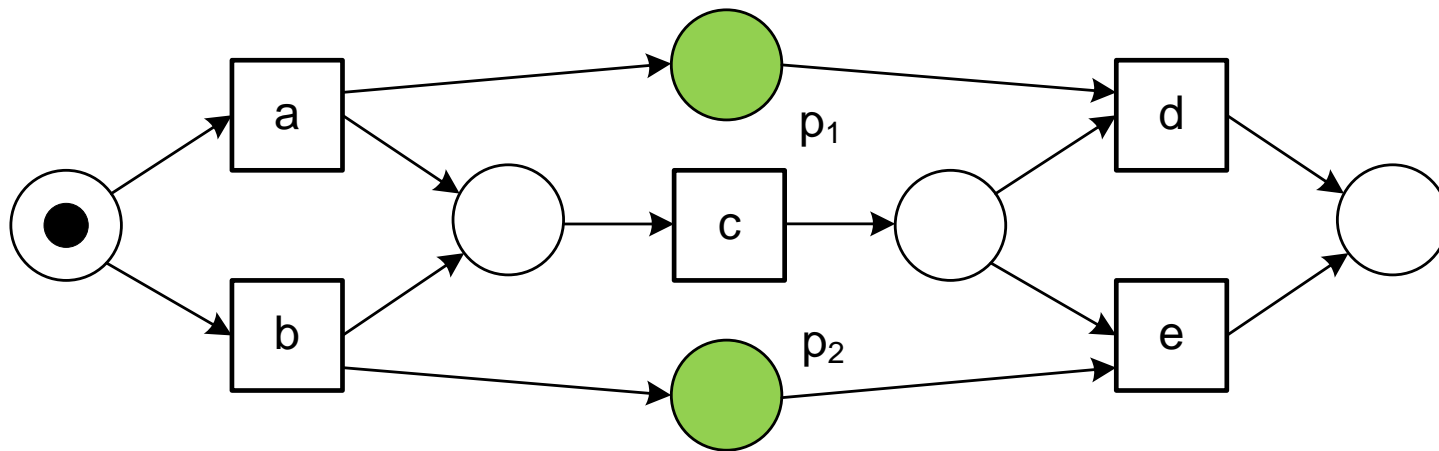
Limitation of α algorithm (loops of length 2)

$$L_8 = [\langle a, b, d \rangle^3, \langle a, b, c, b, d \rangle^2, \langle a, b, c, b, c, b, d \rangle]$$



Limitation of α algorithm (non-local dependencies)

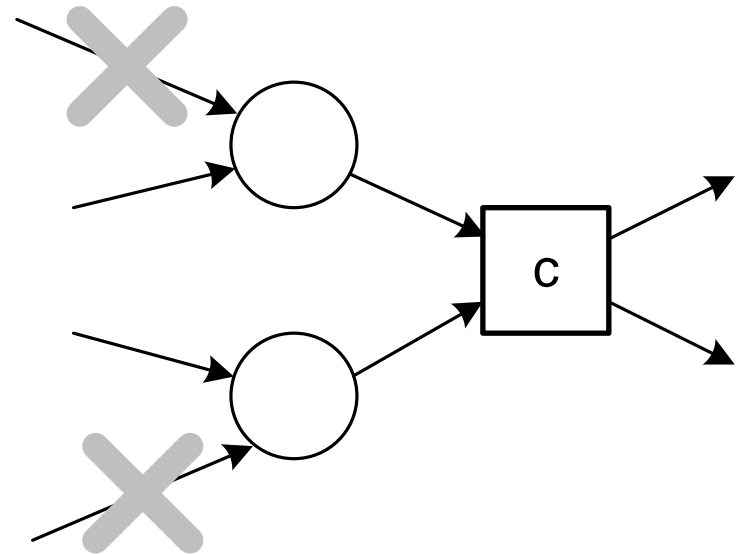
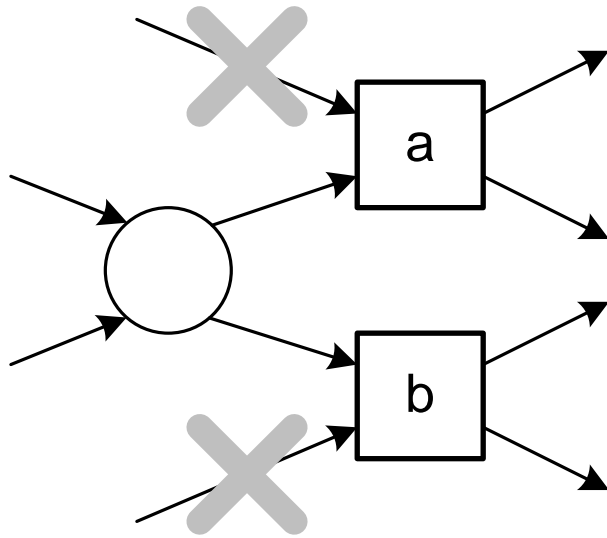
$$L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}]$$



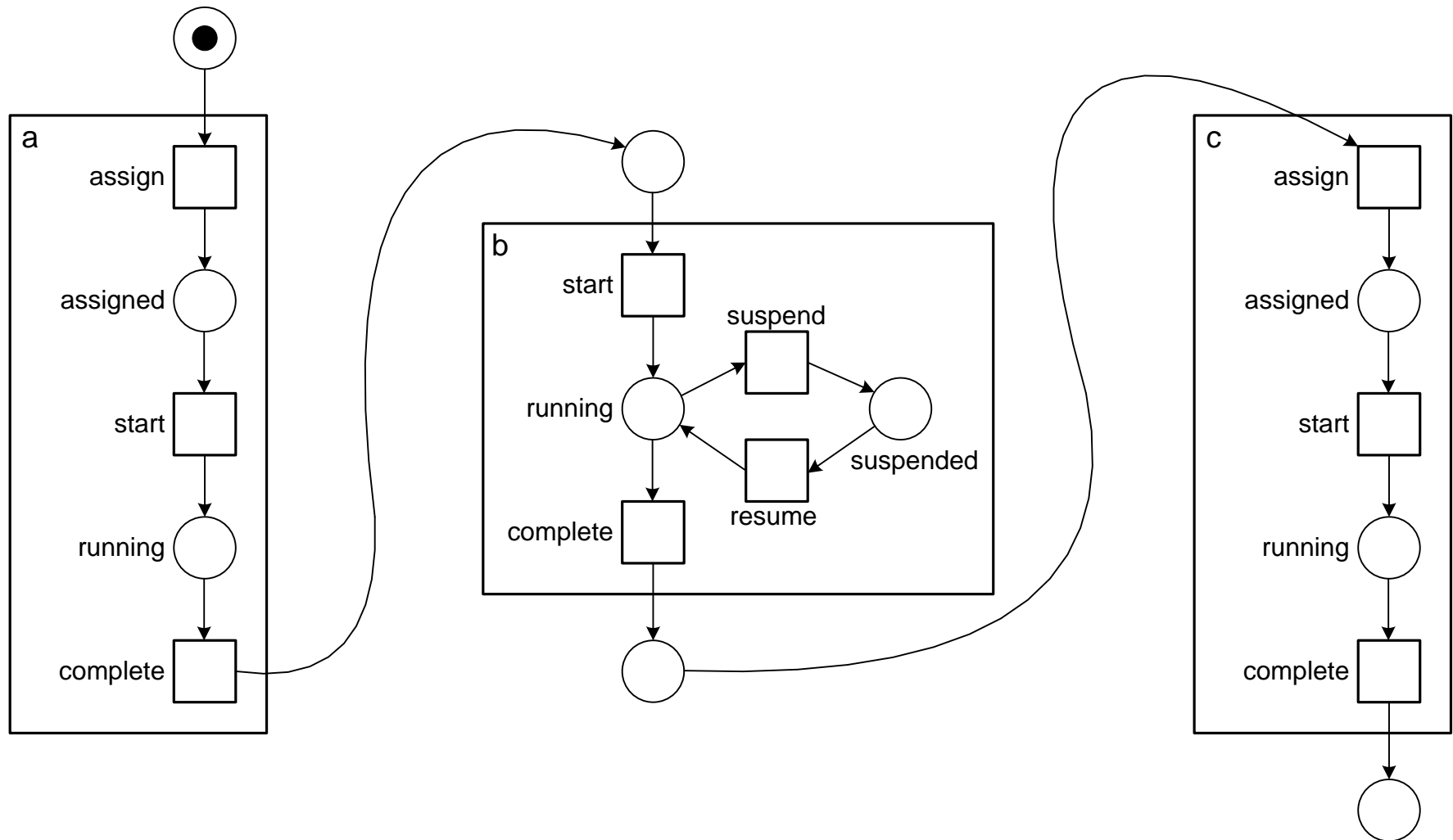
Green places are not discovered!

$$L_4 = [\langle a, c, d \rangle^{45}, \langle b, c, d \rangle^{42}, \langle a, c, e \rangle^{38}, \langle b, c, e \rangle^{22}]$$

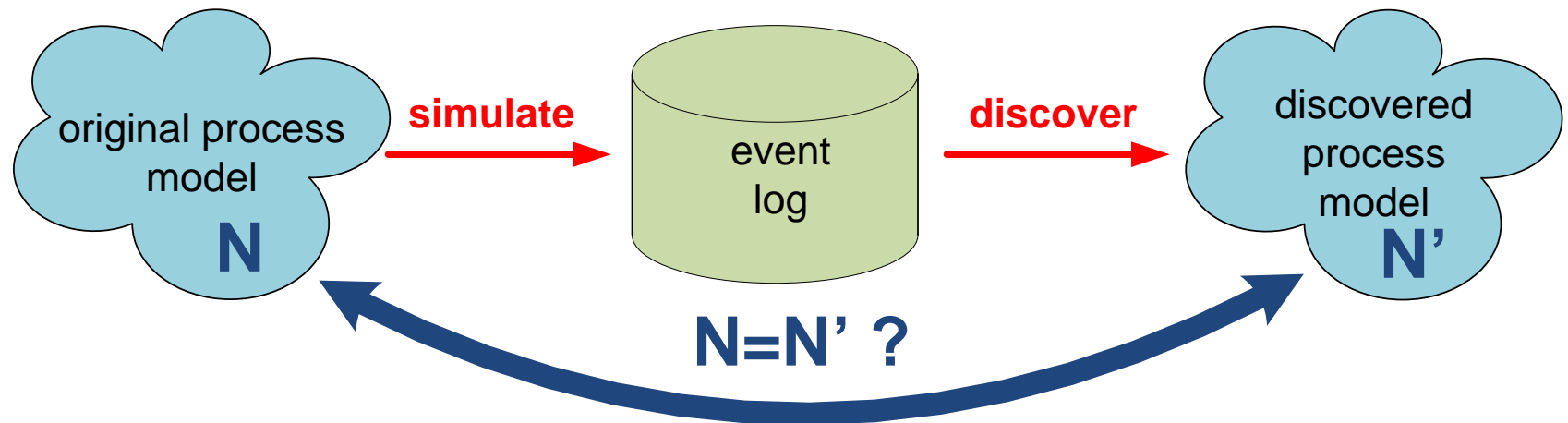
Difficult constructs for α algorithm



Taking the transactional life-cycle into account

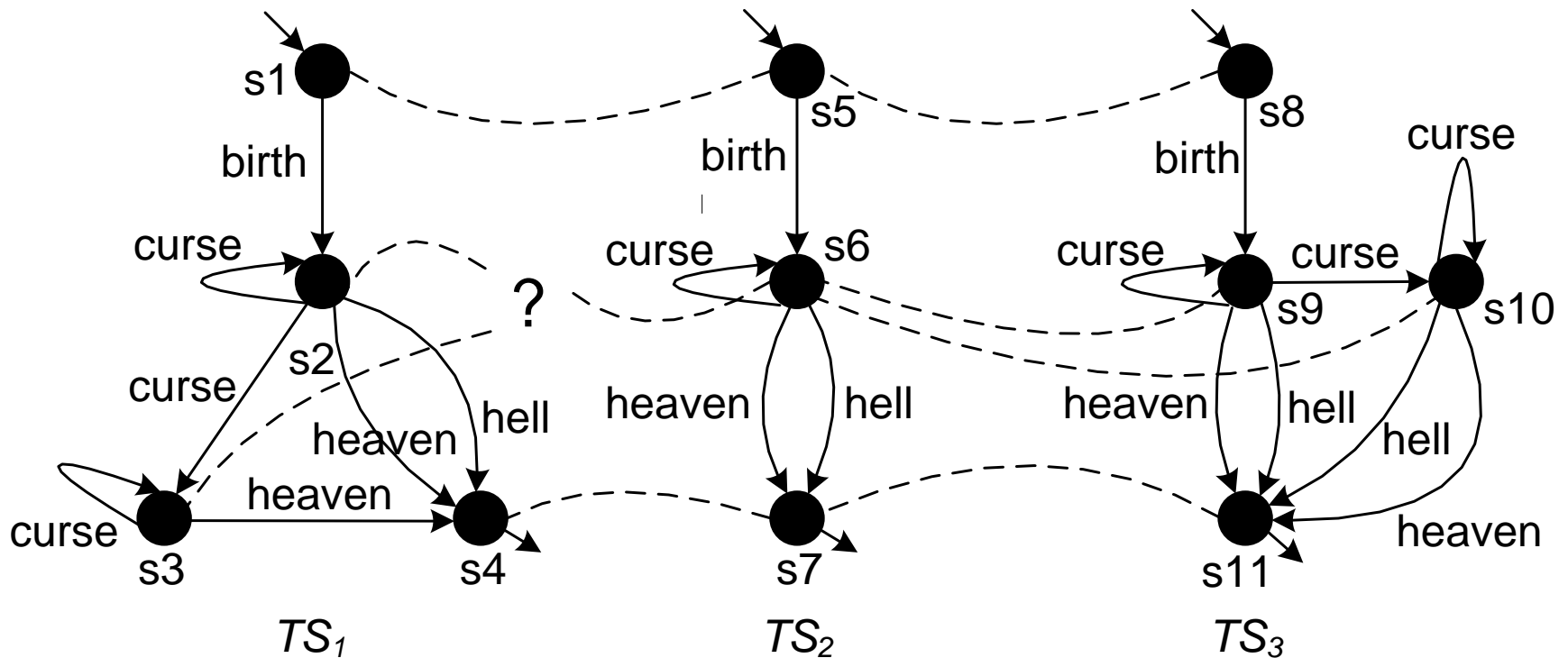


Rediscovering process models



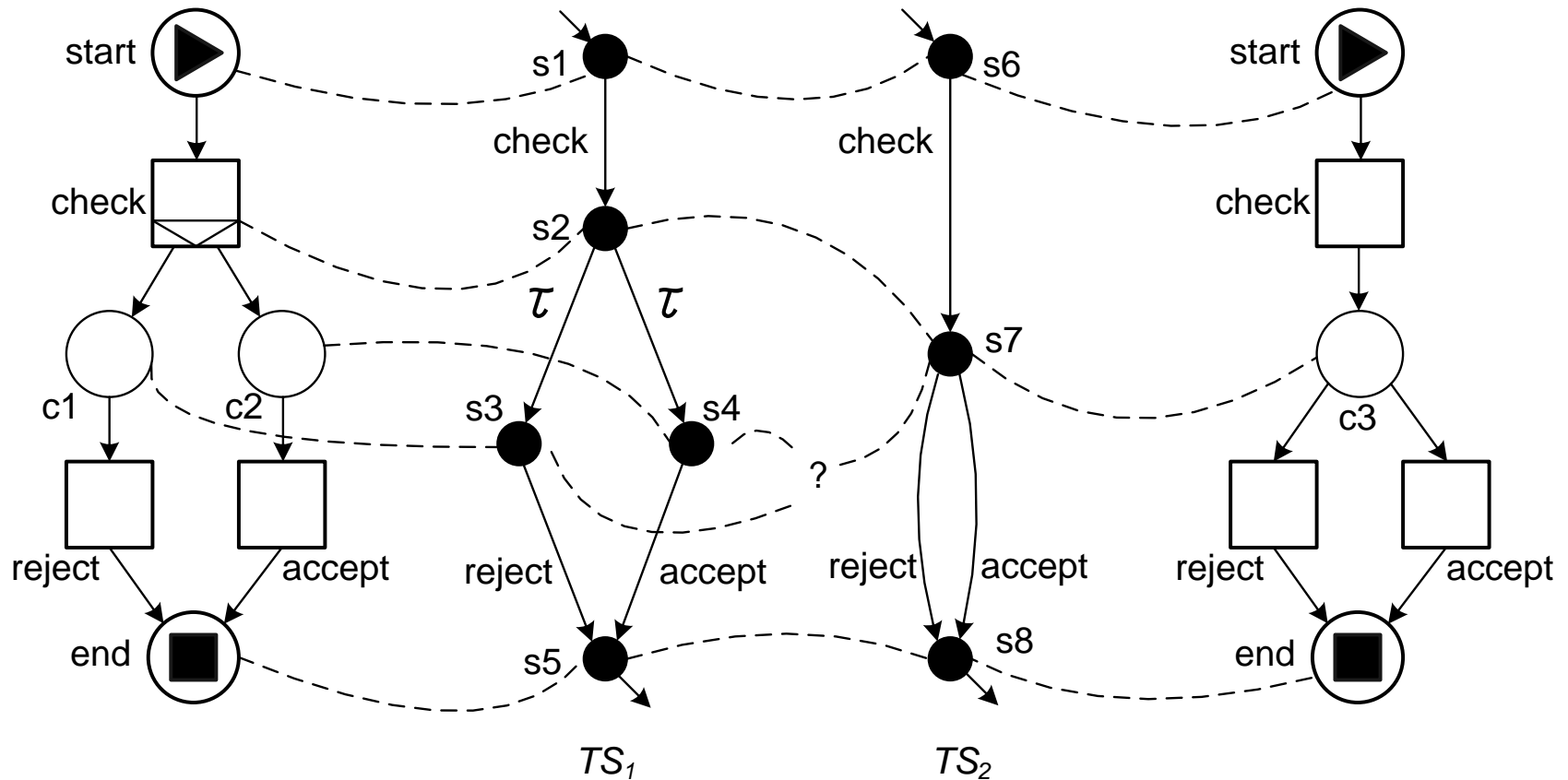
The rediscovery problem: Is the discovered model N' equivalent to the original model N?

Equivalence: trace equivalence, bisimilarity, and branching bisimilarity



Three trace equivalent transition systems: TS_1 and TS_2 are not bisimilar, but TS_2 and TS_3 are bisimilar

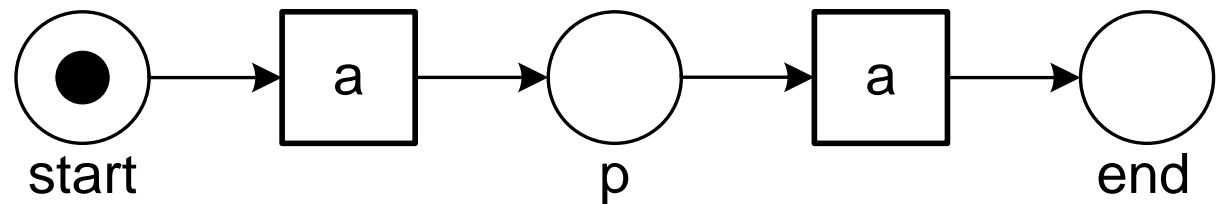
Branching bisimilarity defined for YAWL



TS_1 and TS_2 are not branching bisimilar (although trace equivalent).

Challenge: finding the right representational bias

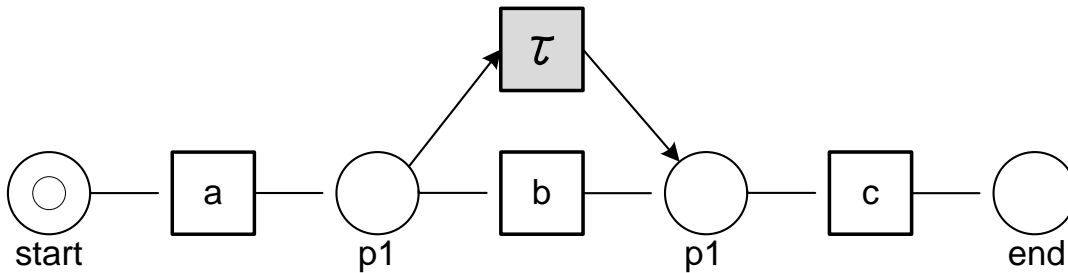
$$L_{10} = [\langle a, a \rangle^{55}]$$



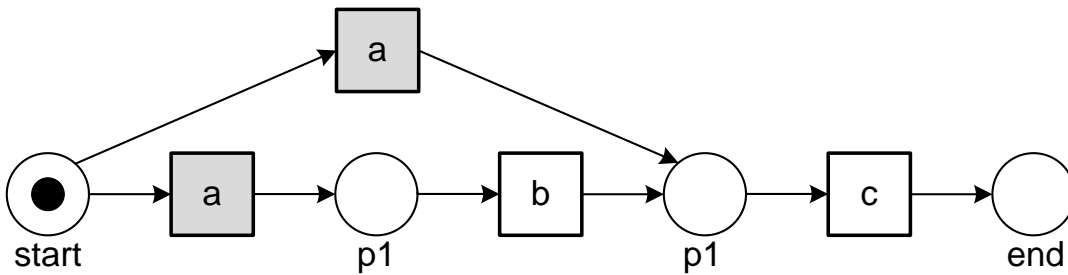
There is no WF-net with unique visible labels that exhibits this behavior.

Another example

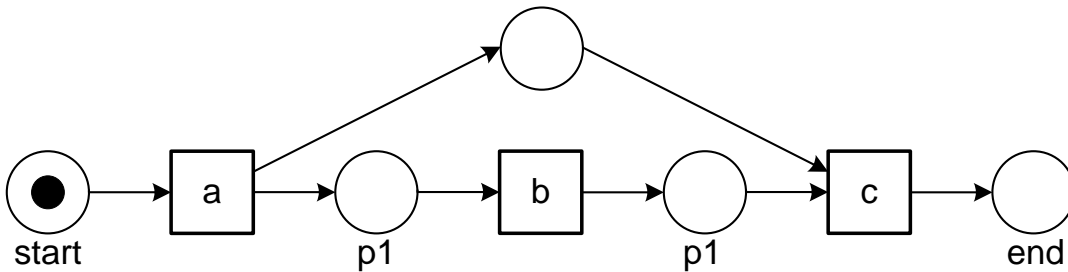
$$L_{11} = [\langle a, b, c \rangle^{20}, \langle a, c \rangle^{30}]$$



(a)



(b)



(c)

There is no WF-net with unique visible labels that exhibits this behavior.

Challenge: noise and incompleteness

- To discover a suitable process model it is assumed that the event log contains a representative sample of behavior.
- Two related phenomena:
 - **Noise**: the event log contains rare and infrequent behavior not representative for the typical behavior of the process.
 - **Incompleteness**: the event log contains too few events to be able to discover some of the underlying control-flow structures.

More on incompleteness

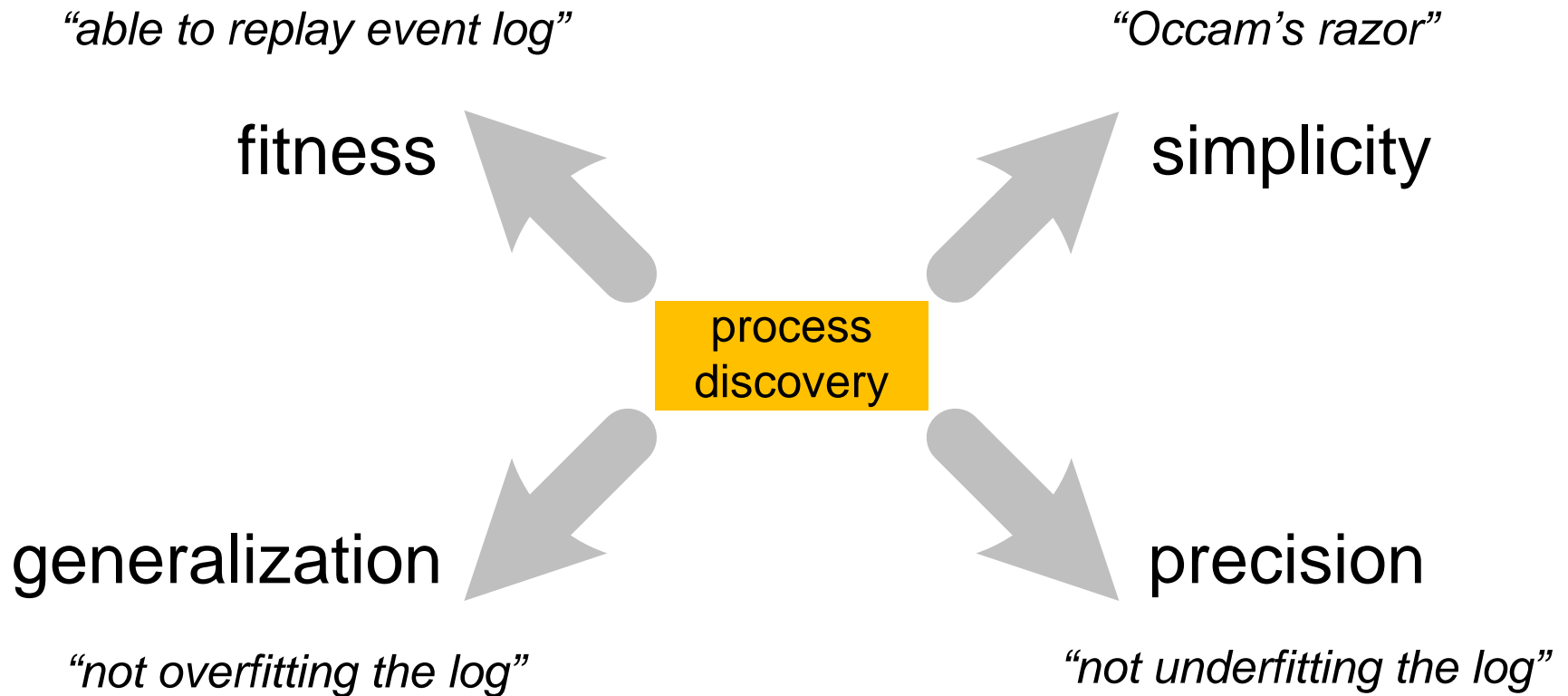
To illustrate the relevance of completeness, consider a process consisting of 10 activities that can be executed in parallel and a corresponding log that contains information about 10,000 cases. The total number of possible interleavings in the model with 10 concurrent activities is $10! = 3,628,800$. Hence, it is impossible that each interleaving is present in the log as there are fewer cases (10,000) than potential traces (3,628,800). Even if there are 3,628,800 cases in the log, it is extremely unlikely that all possible variations are present. To motivate this consider the following analogy. In a group of 365 people it is very unlikely that everyone has a different birthdate. The probability is $365!/365^{365} \approx 1.454955 \times 10^{-157} \approx 0$, i.e., incredibly small. The number of atoms in the universe is often estimated to be approximately 10^{79} [129].

See also chapter 3 (cross-validation, precision, recall, etc.)

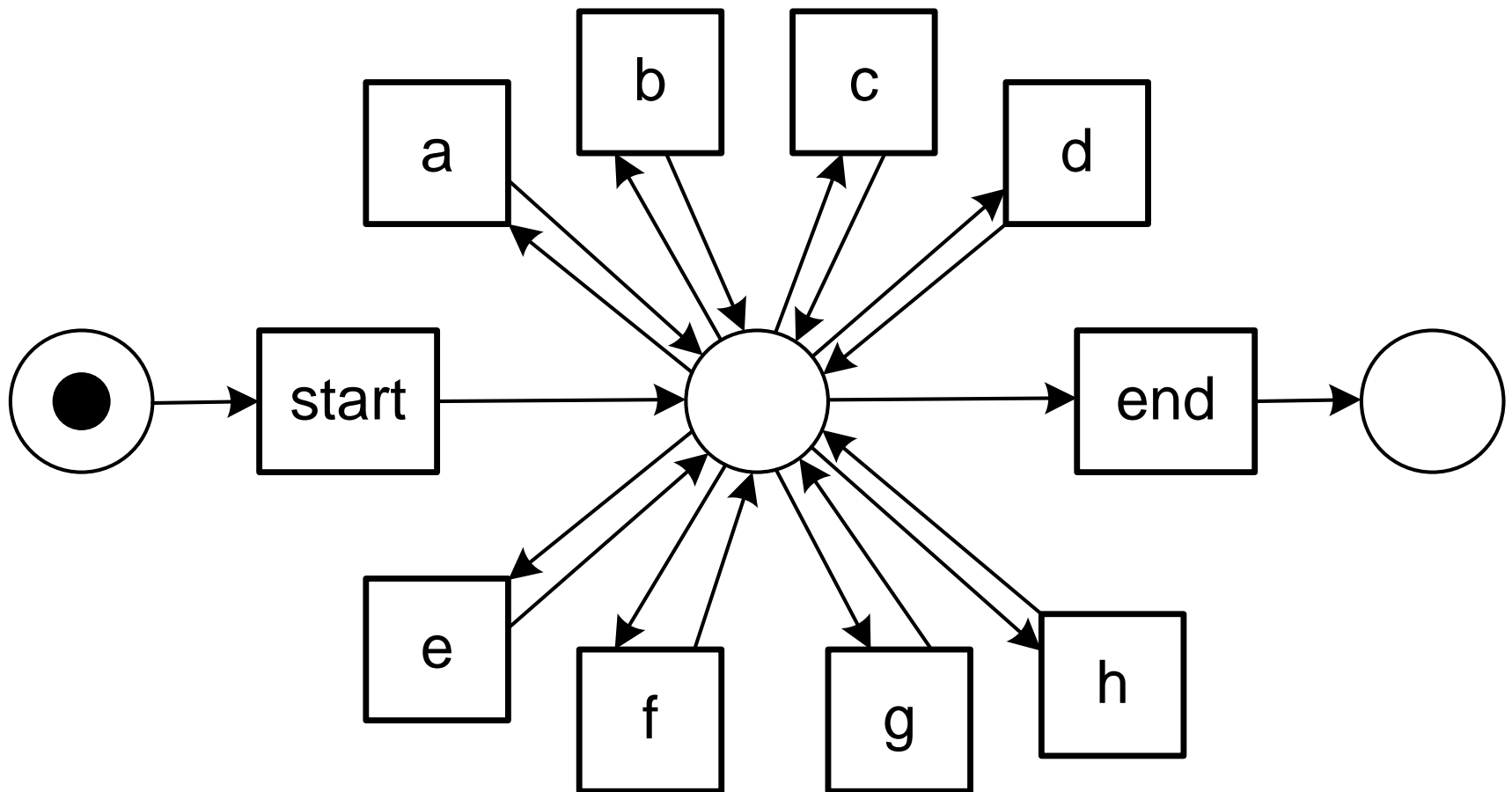


Challenge: Balancing Between Underfitting and Overfitting

Challenge: four competing quality criteria

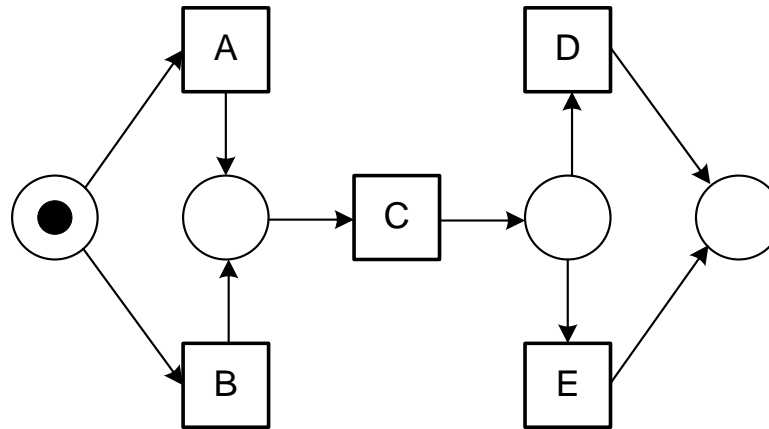
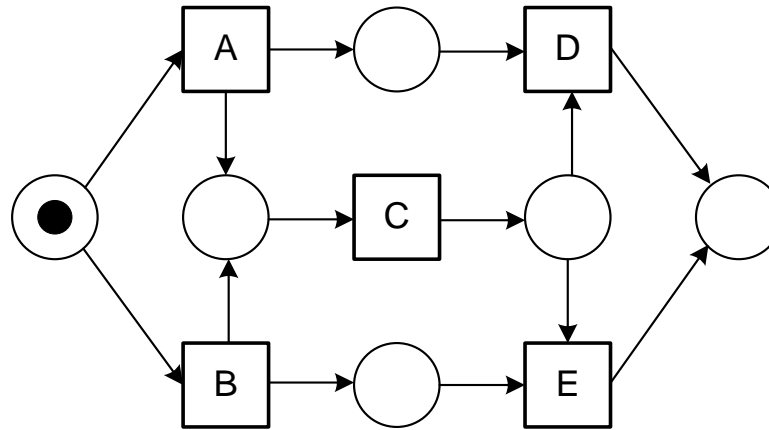


Flower model



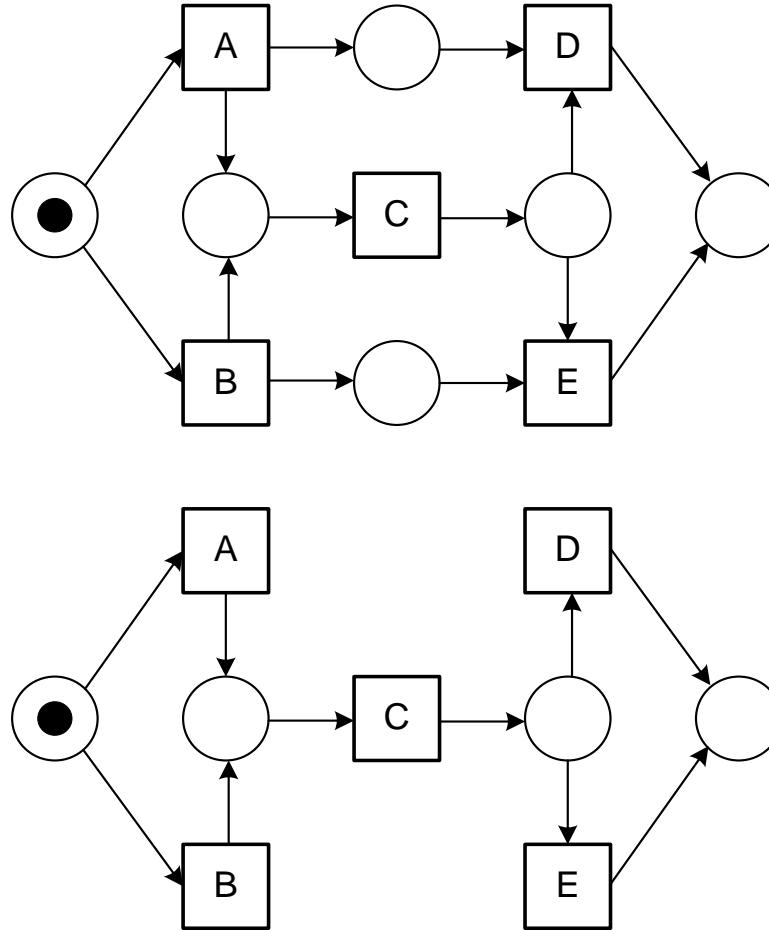
What is the best model?

ACD	99
ACE	0
BCE	85
BCD	0



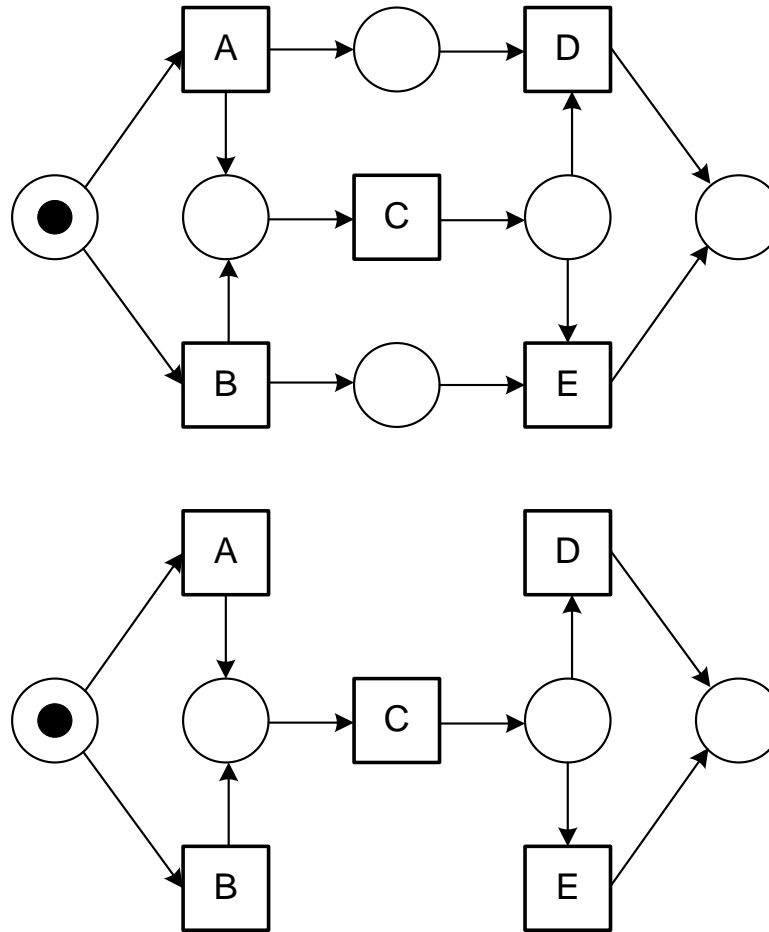
What is the best model?

ACD	99
ACE	88
BCE	85
BCD	78

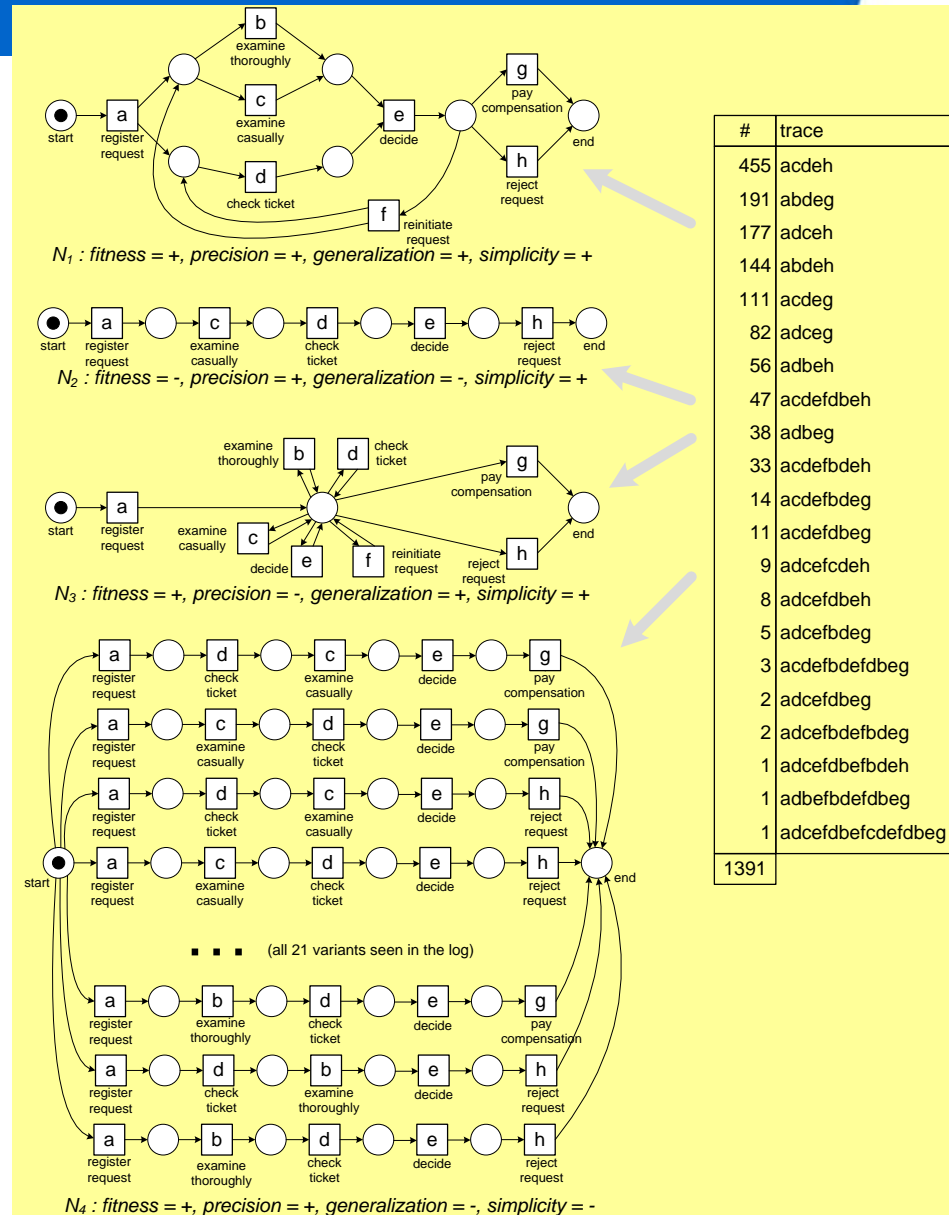
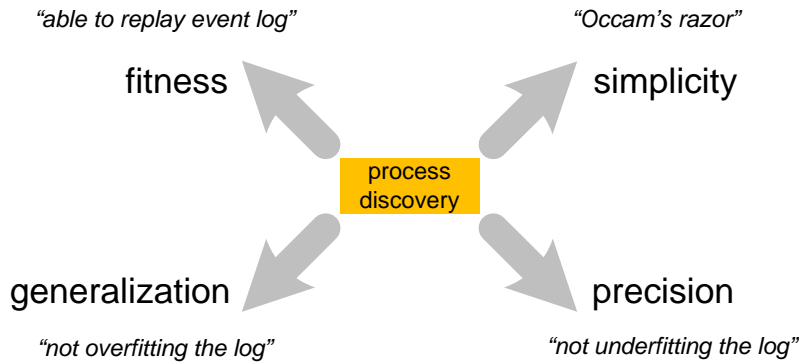


What is the best model?

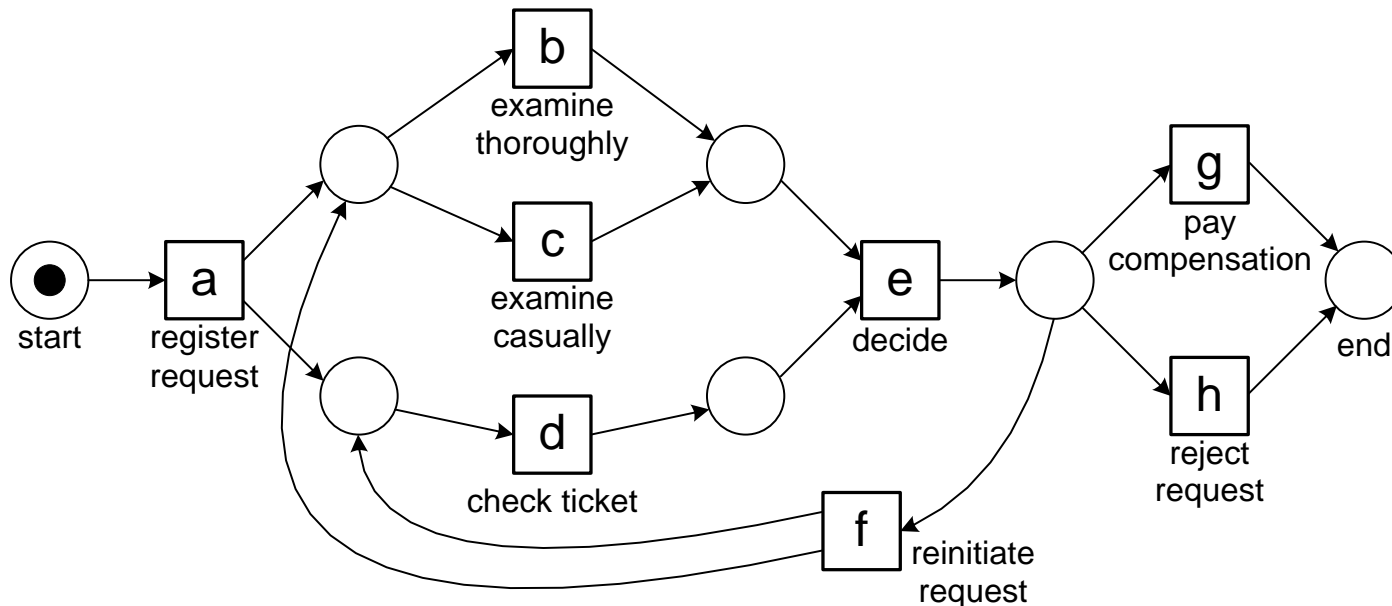
ACD	99
ACE	2
BCE	85
BCD	3



Example: one log four models



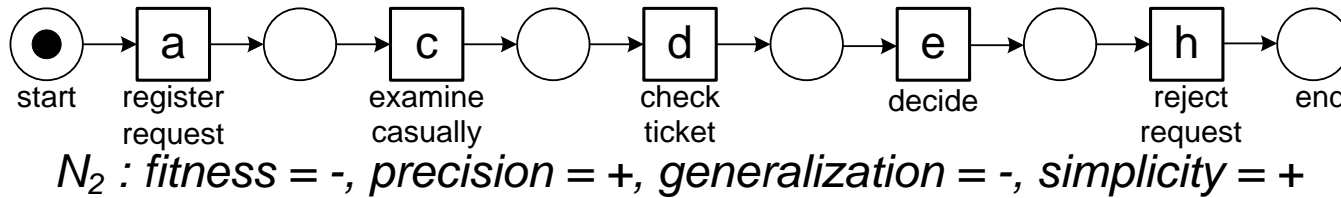
Model N_1



N_1 : fitness = +, precision = +, generalization = +, simplicity = +

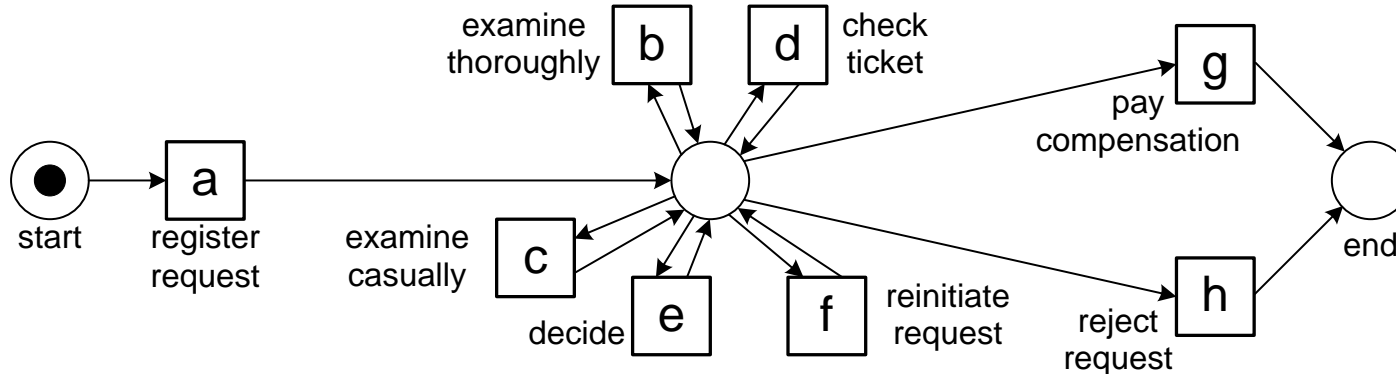
#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Model N_2



#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefdbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

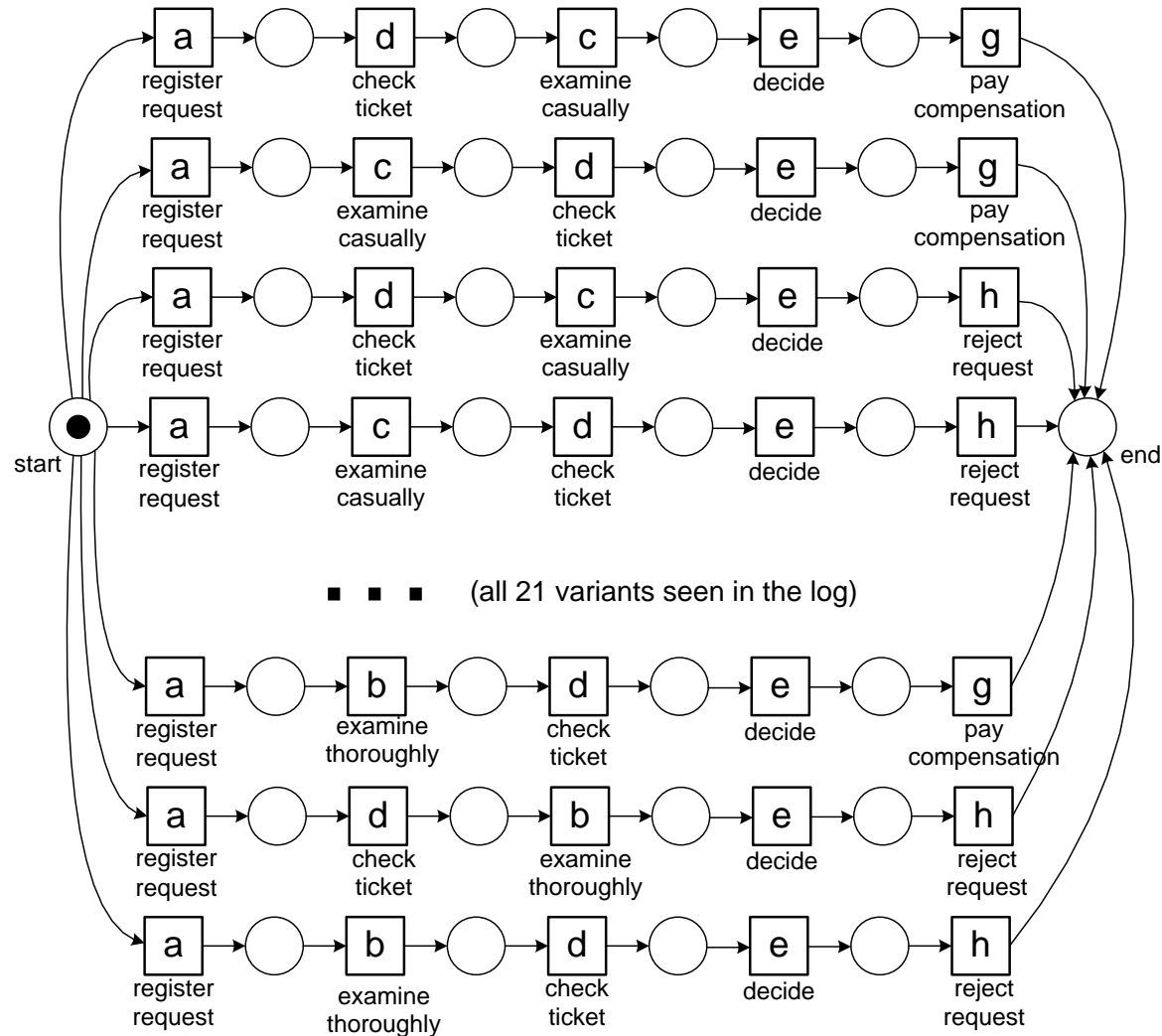
Model N_3



N_3 : fitness = +, precision = -, generalization = +, simplicity = +

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Model N₄



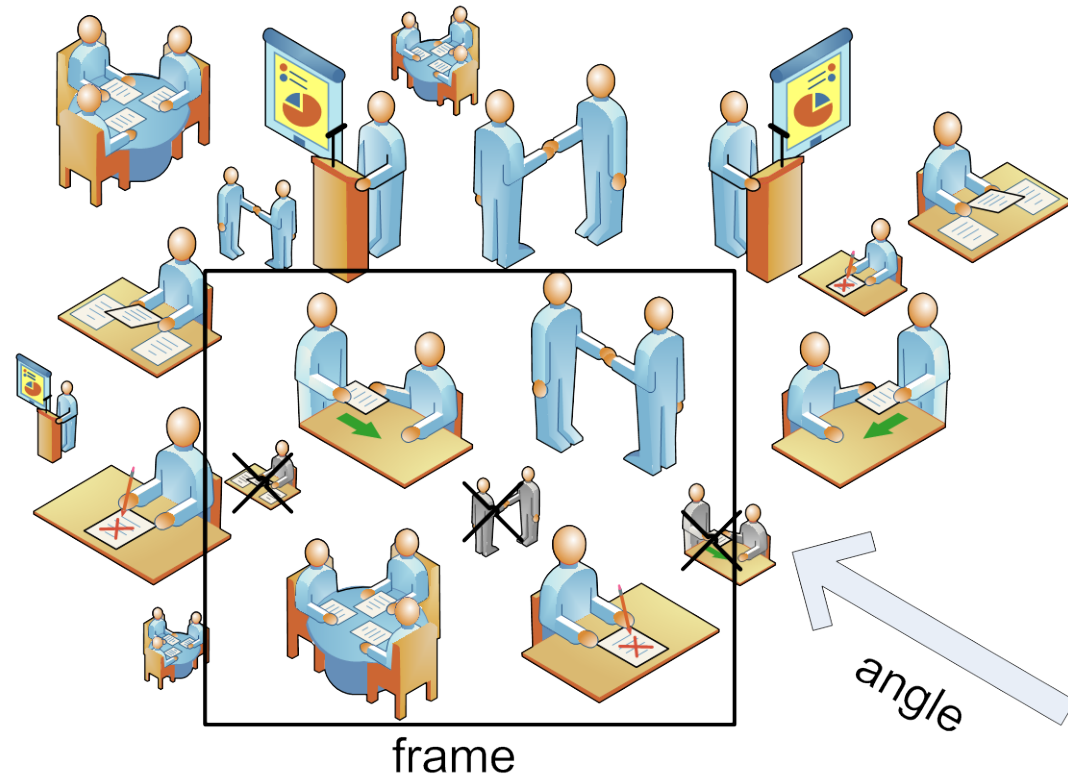
N_4 : fitness = +, precision = +, generalization = -, simplicity = -

#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Why is process mining such a difficult problem?

- There are **no negative examples** (i.e., a log shows what has happened but does not show what could not happen).
- Due to concurrency, loops, and choices the **search space has a complex structure** and the log typically contains only a **fraction** of all possible behaviors.
- There is **no clear relation** between the size of a model and its behavior (i.e., a smaller model may generate more or less behavior although classical analysis and evaluation methods typically assume some monotonicity property).

Creating a 2-D slice of a 3-D reality



Creating a 2-D slice of a 3-D reality: the process is viewed from a specific angle, the process is scoped using a frame, and the resolution determines the granularity of the resulting model