

Process Mining: The α Algorithm

prof.dr.ir. Wil van der Aalst

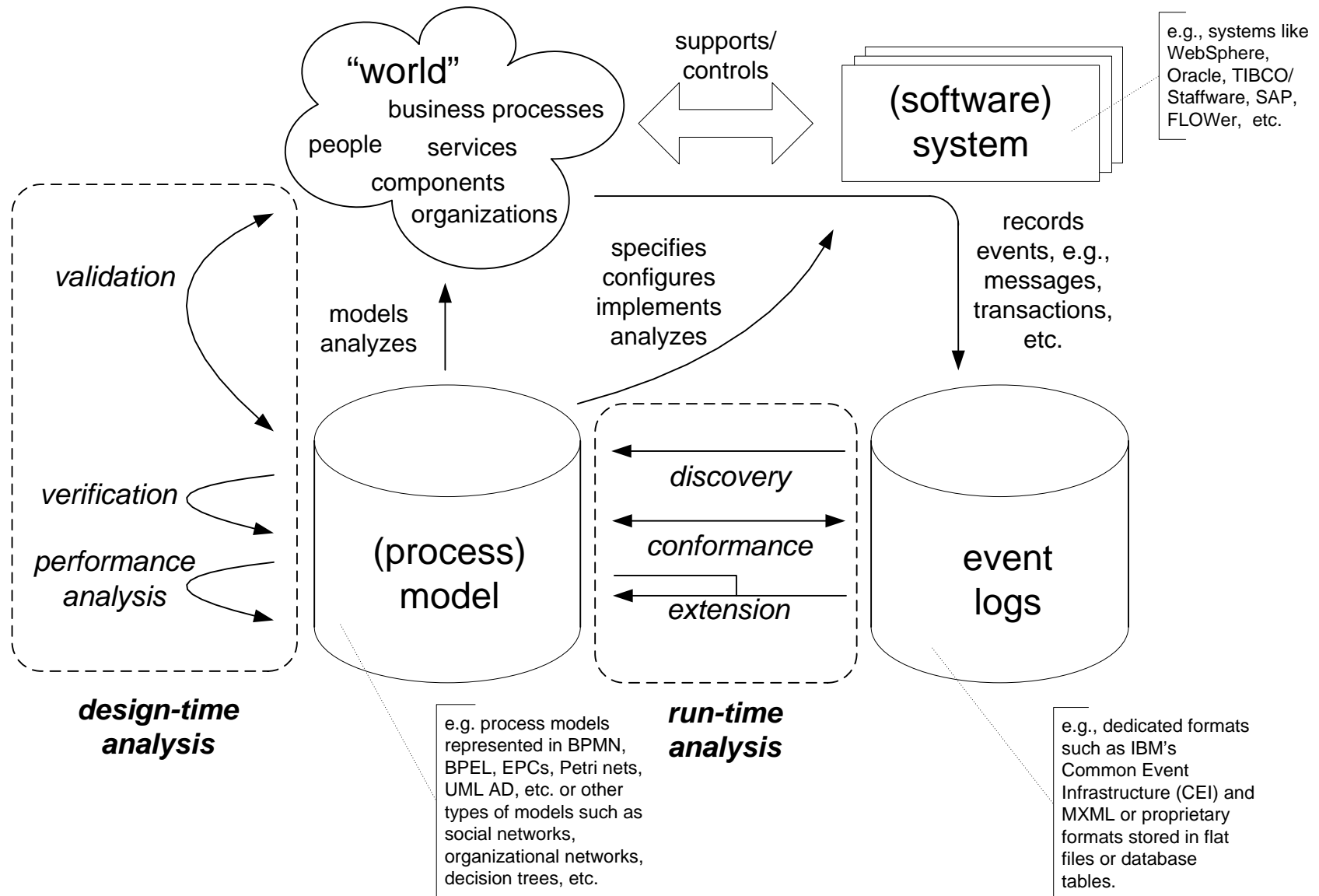


TU/e

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Design-time analysis vs run-time analysis



Relevant material

1. Jörg Desel, Wolfgang Reisig: Place/Transition Petri Nets. Petri Nets 1996: 122-173. DOI: 10.1007/3-540-65306-6_15
<http://www.springerlink.com/content/x6hn592l35866lu8/fulltext.pdf>
2. Tadao Murata, Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE. 77(4): 541-580, April, 1989. <http://dx.doi.org/10.1109/5.24143>
<http://ieeexplore.ieee.org/iel1/5/911/00024143.pdf>
3. Wil van der Aalst: Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer Verlag 2011 (chapters 1 & 5)
 - a) Chapter 1: DOI: 10.1007/978-3-642-19345-3_1
<http://www.springerlink.com/content/p443h219v3u3537l/fulltext.pdf>
 - b) Chapter 5: DOI: 10.1007/978-3-642-19345-3_5
<http://www.springerlink.com/content/u58h17n3167p0x1u/fulltext.pdf>
 - c) Events logs: <http://www.processmining.org/book/>

Today's focus is on 3.

Wil M. P. van der Aalst

Process Mining

Discovery, Conformance and Enhancement of Business Processes

More and more information about business processes is recorded by information systems in the form of so-called "event logs". Despite the omnipresence of such data, most organizations diagnose problems based on fiction rather than facts. Process mining is an emerging discipline based on process model-driven approaches and data mining. It not only allows organizations to fully benefit from the information stored in their systems, but it can also be used to check the conformance of processes, detect bottlenecks, and predict execution problems.

Wil van der Aalst delivers the first book on process mining. It aims to be self-contained while covering the entire process mining spectrum from process discovery to operational support. In Part I, the author provides the basics of business process modeling and data mining necessary to understand the remainder of the book. Part II focuses on process discovery as the most important process mining task. Part III moves beyond discovering the control flow of processes and highlights conformance checking, and organizational and time perspectives. Part IV guides the reader in successfully applying process mining in practice, including an introduction to the widely used open-source tool ProM. Finally, Part V takes a step back, reflecting on the material presented and the key open challenges.

Overall, this book provides a comprehensive overview of the state of the art in process mining. It is intended for business process analysts, business consultants, process managers, graduate students, and BPM researchers.

Features and Benefits:

- First book on process mining, bridging the gap between business process modeling and business intelligence.
- Written by one of the most influential and most-cited computer scientists and the best-known BPM researcher.
- Self-contained and comprehensive overview for a broad audience in academia and industry.
- The reader can put process mining into practice immediately due to the applicability of the techniques and the availability of the open-source process mining software ProM.

Computer Science



► springer.com

van der Aalst



Process Mining

Wil M. P. van der Aalst

Process Mining

Discovery, Conformance and
Enhancement of Business Processes

 Springer

Process Discovery

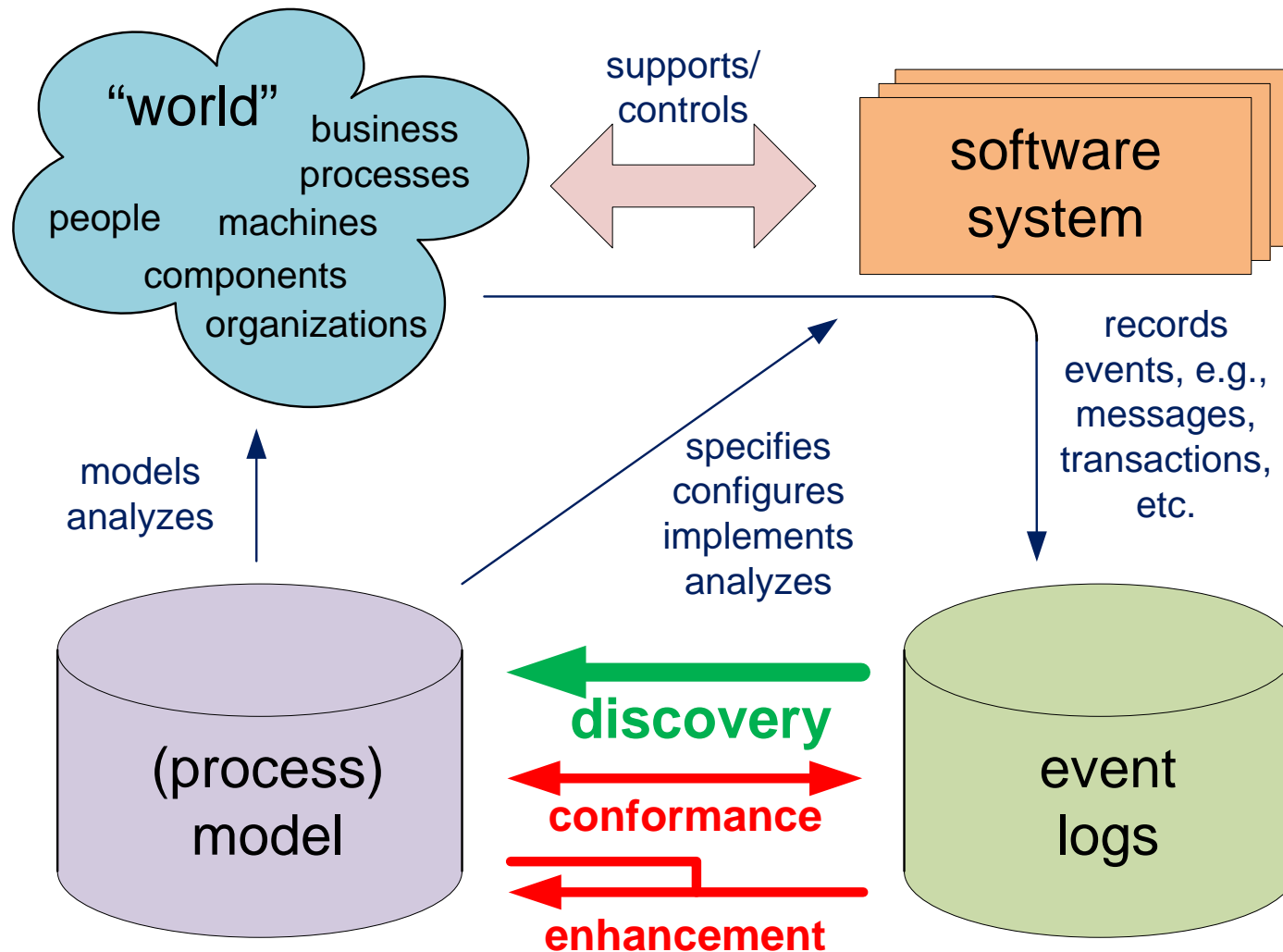


TU/e

Technische Universiteit
Eindhoven
University of Technology

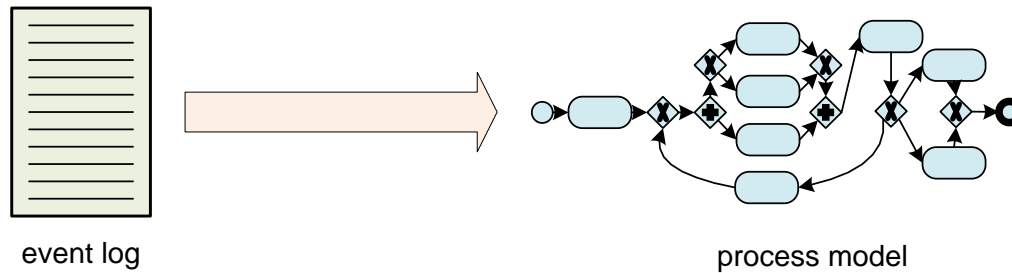
Where innovation starts

Process discovery

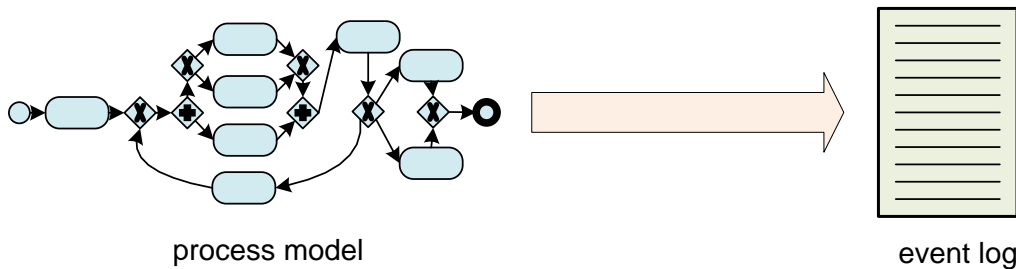


Process discovery = Play-In

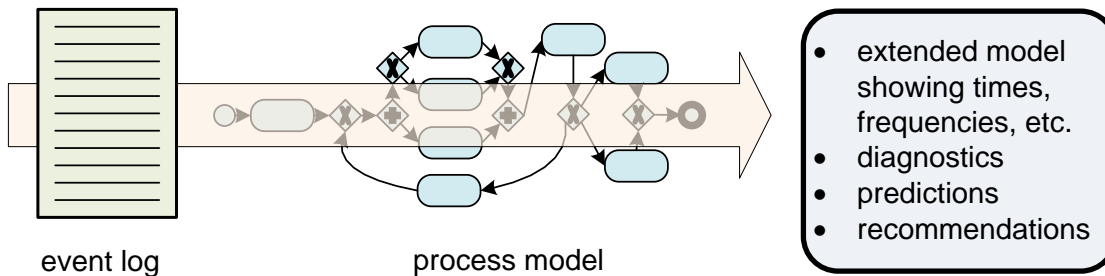
Play-In



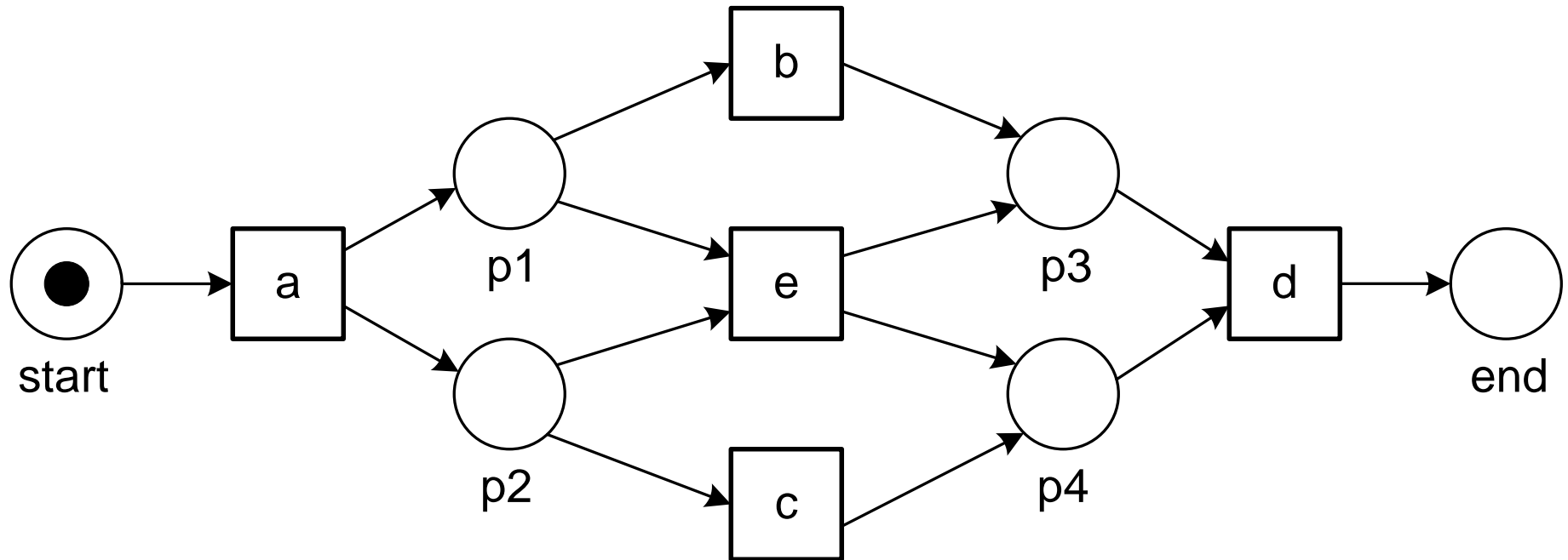
Play-Out



Replay



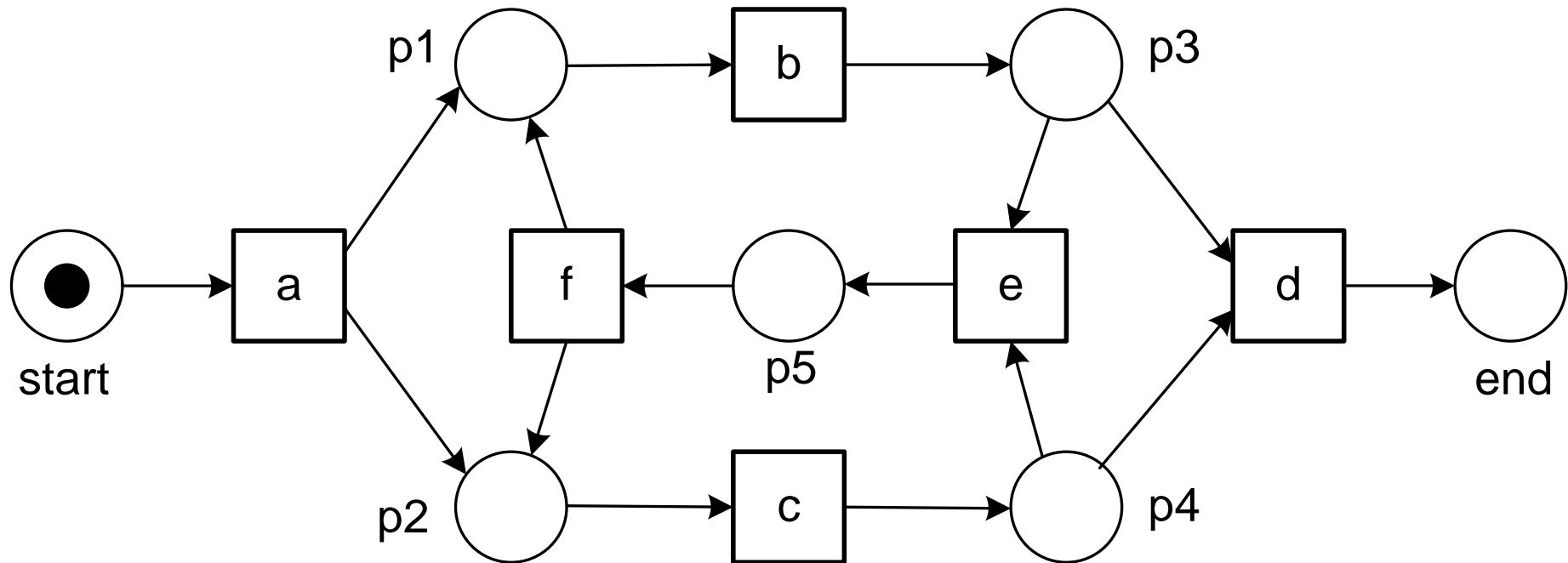
Example



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

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

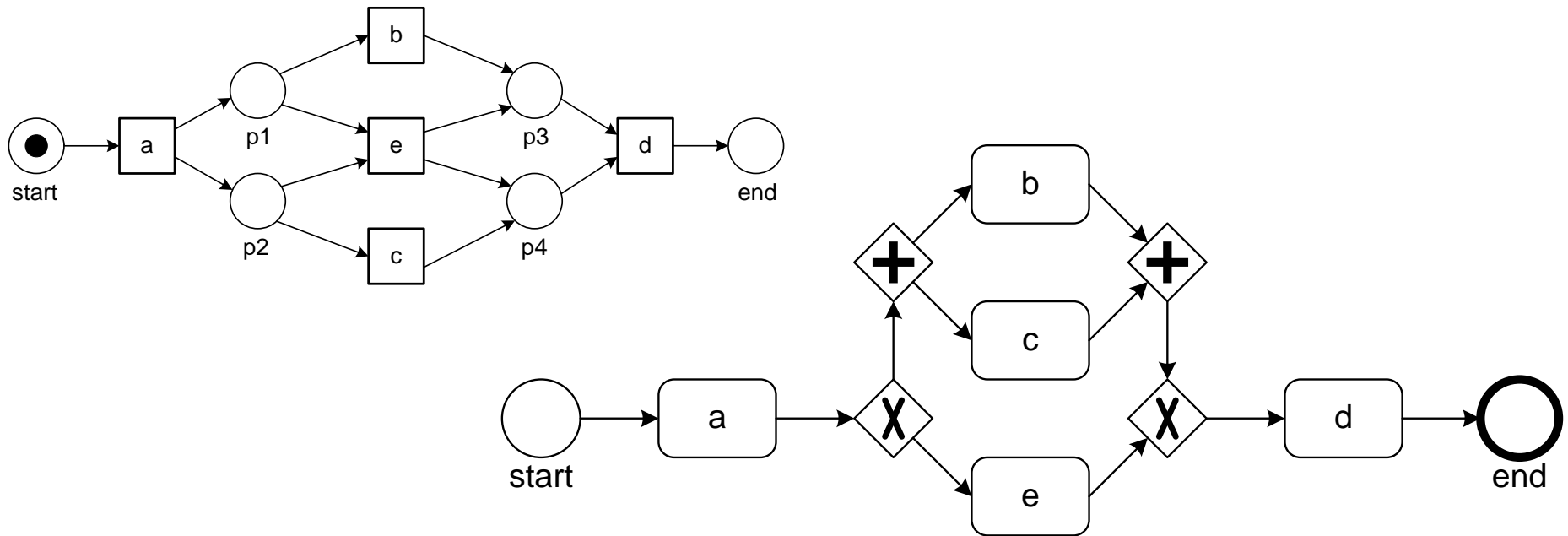
Another example



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

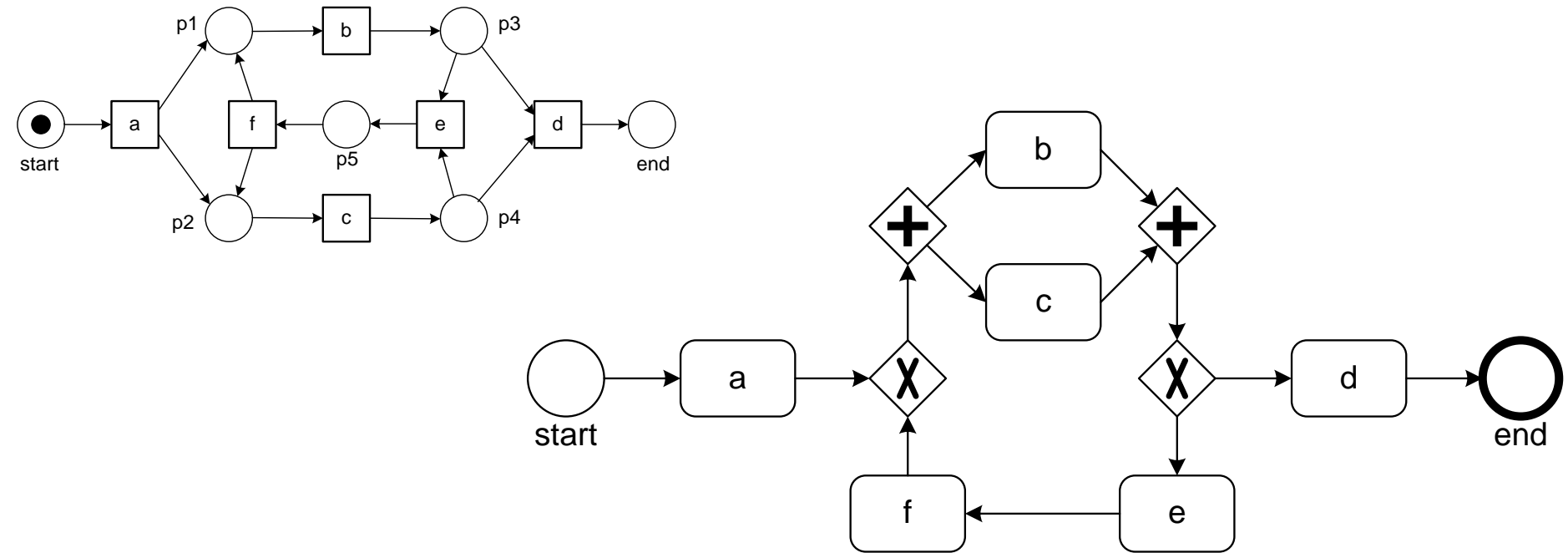
Generalization: event log contains only subset of all possible traces of model.

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.

α Algorithm

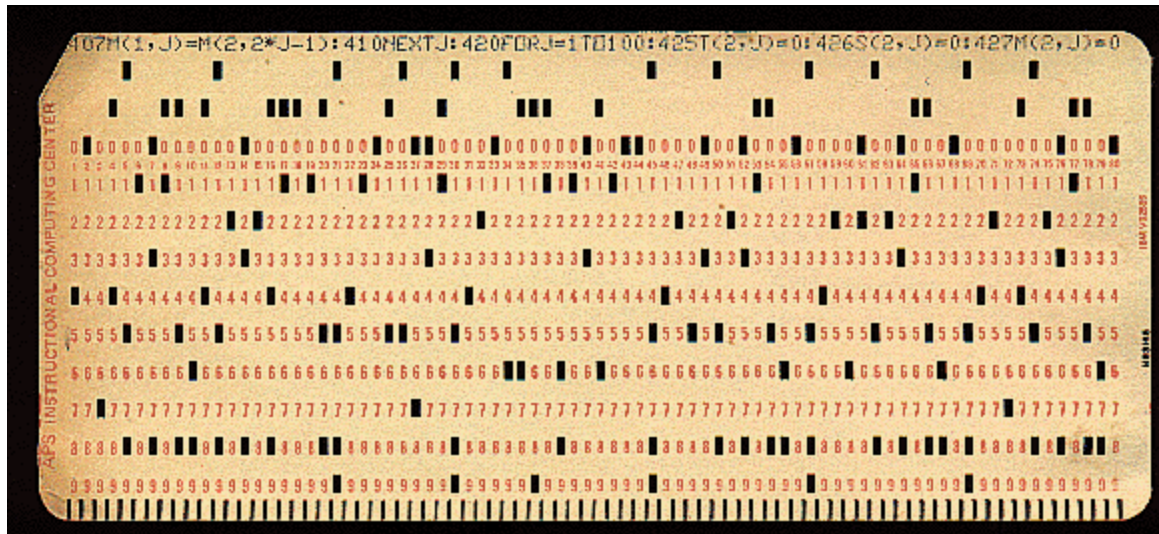



TU/e

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

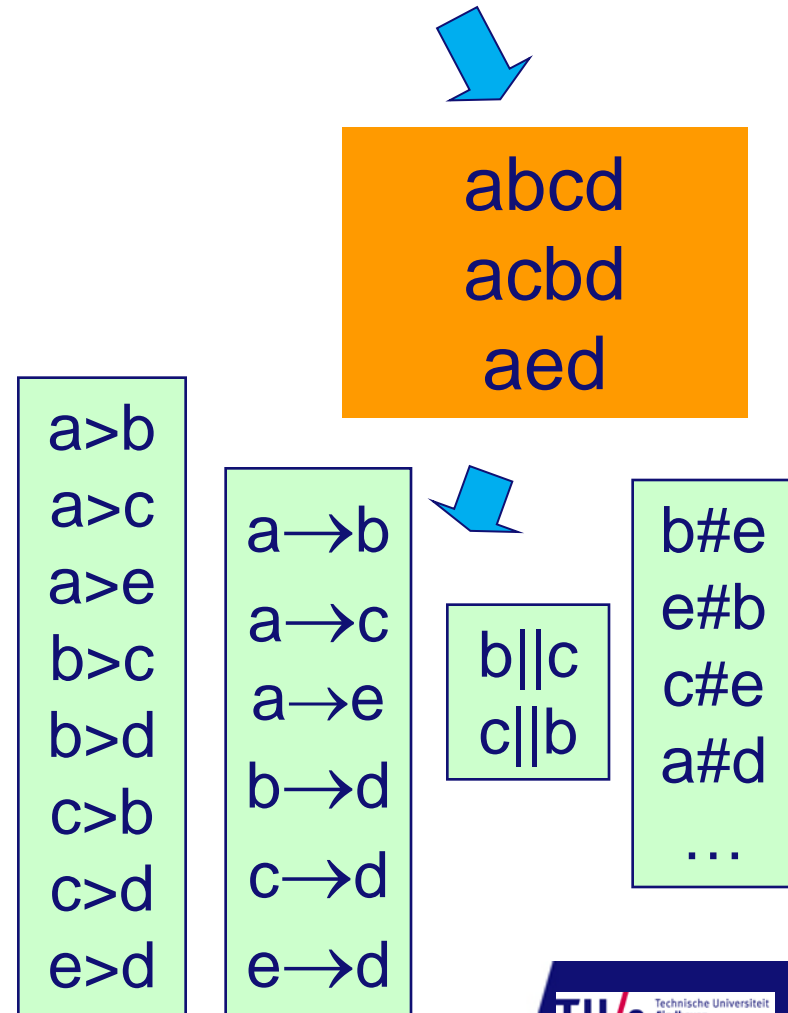
Process Discovery: example of algorithm



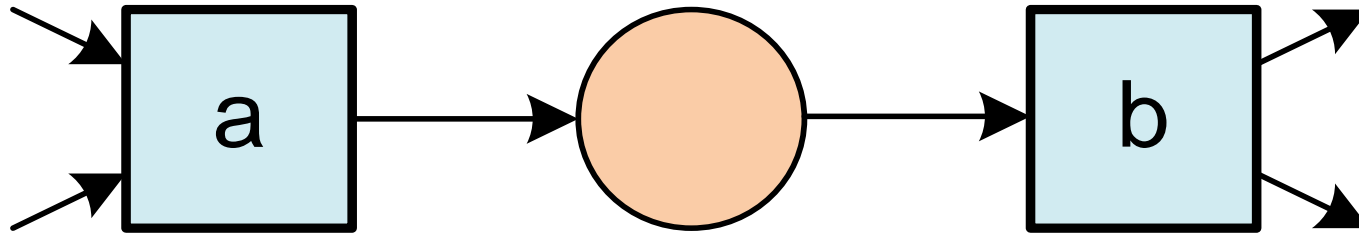
>,→,||,# relations

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

- 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 **$y > x$** .

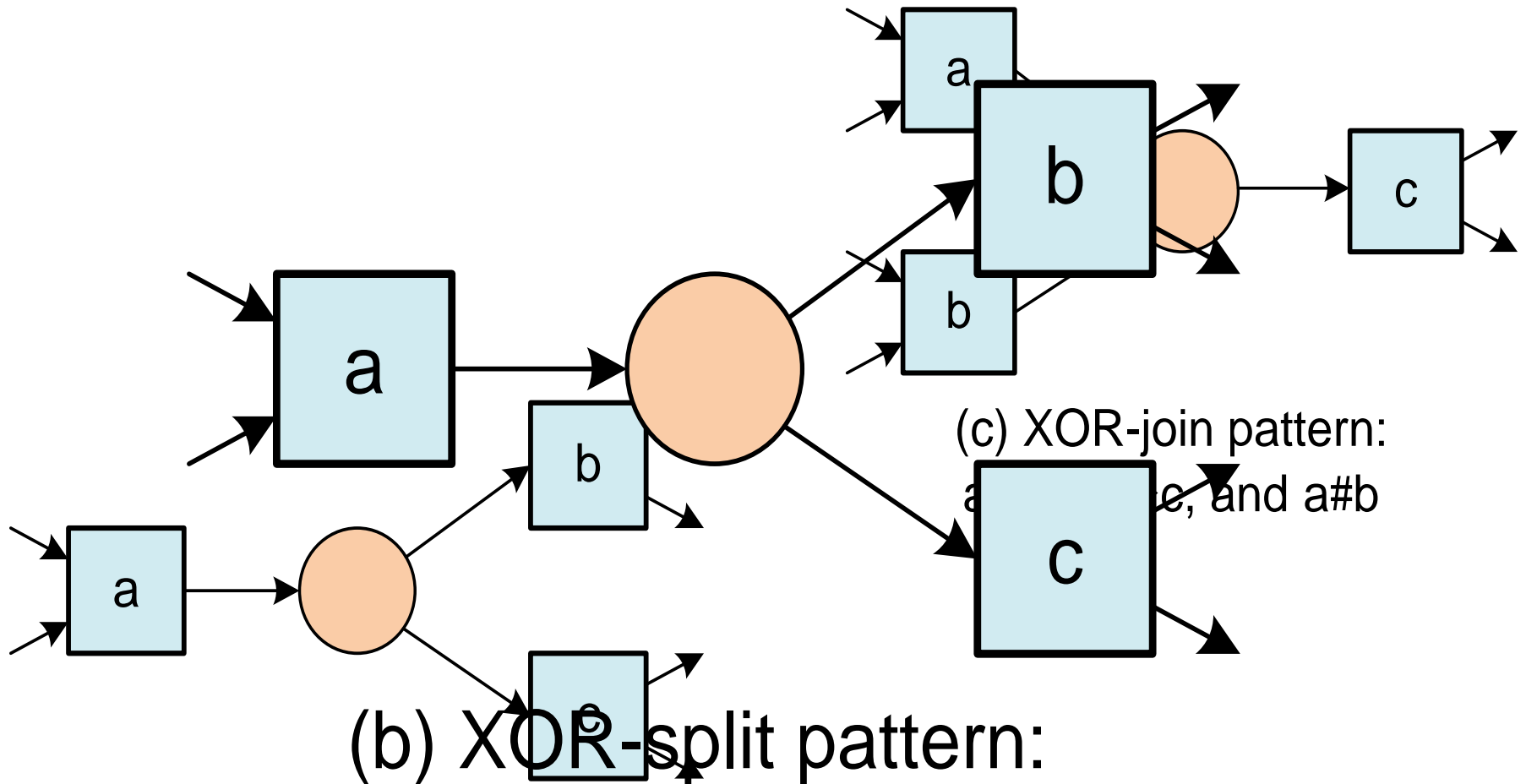


Basic Idea Used by α Algorithm (1)



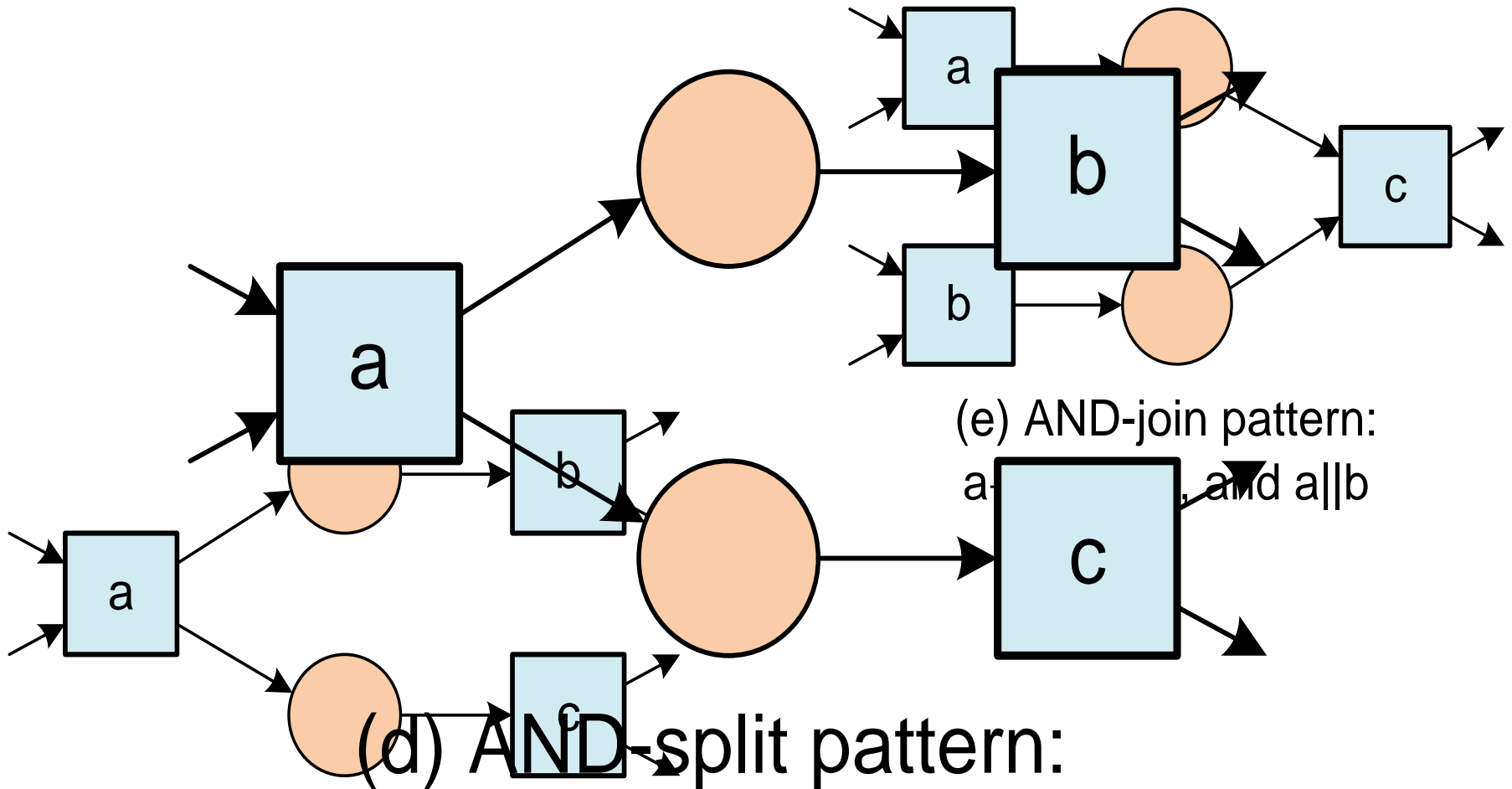
(a) sequence pattern: $a \rightarrow b$

Basic Idea Used by α Algorithm (2)



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

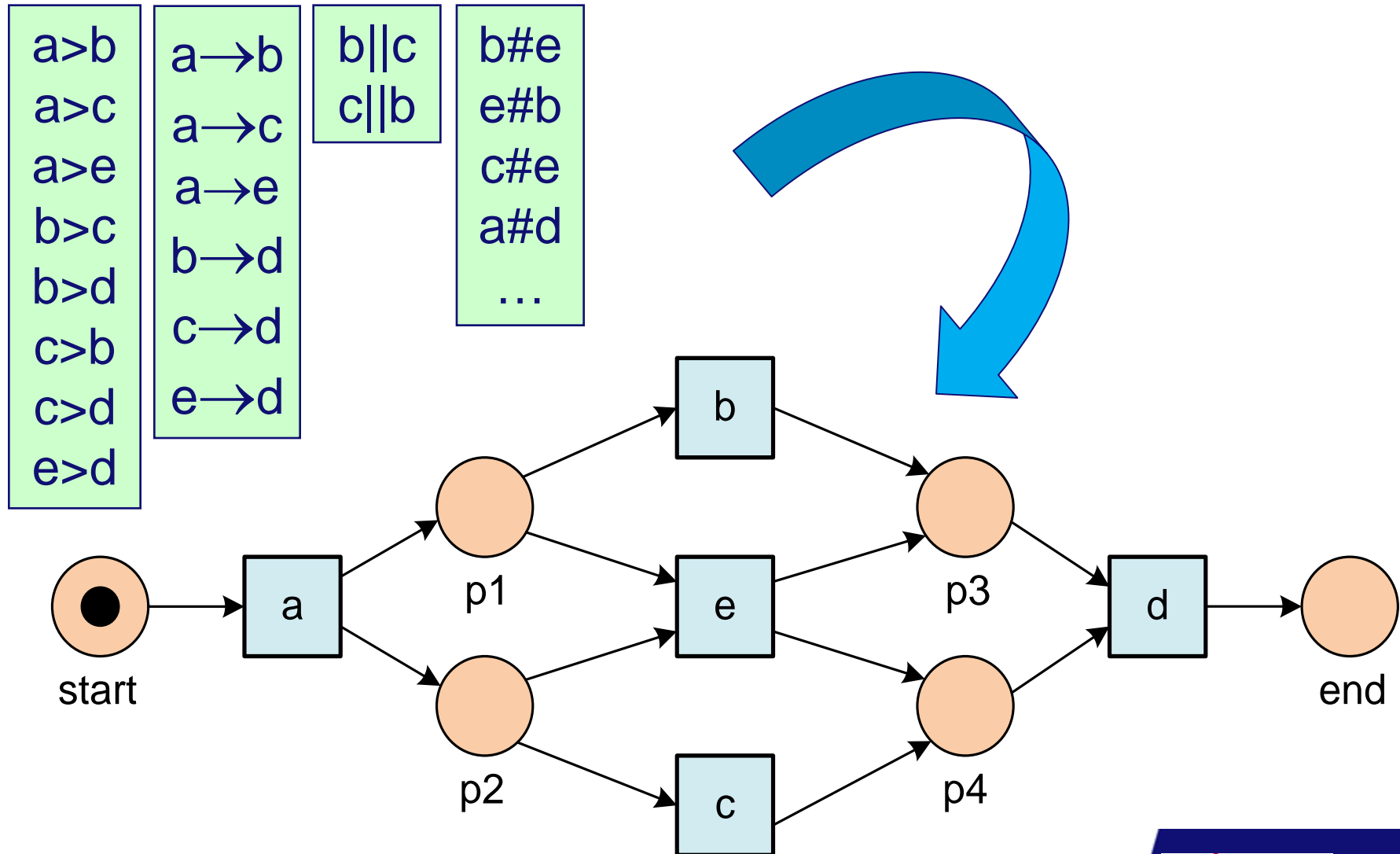
Basic Idea Used by α Algorithm (3)



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

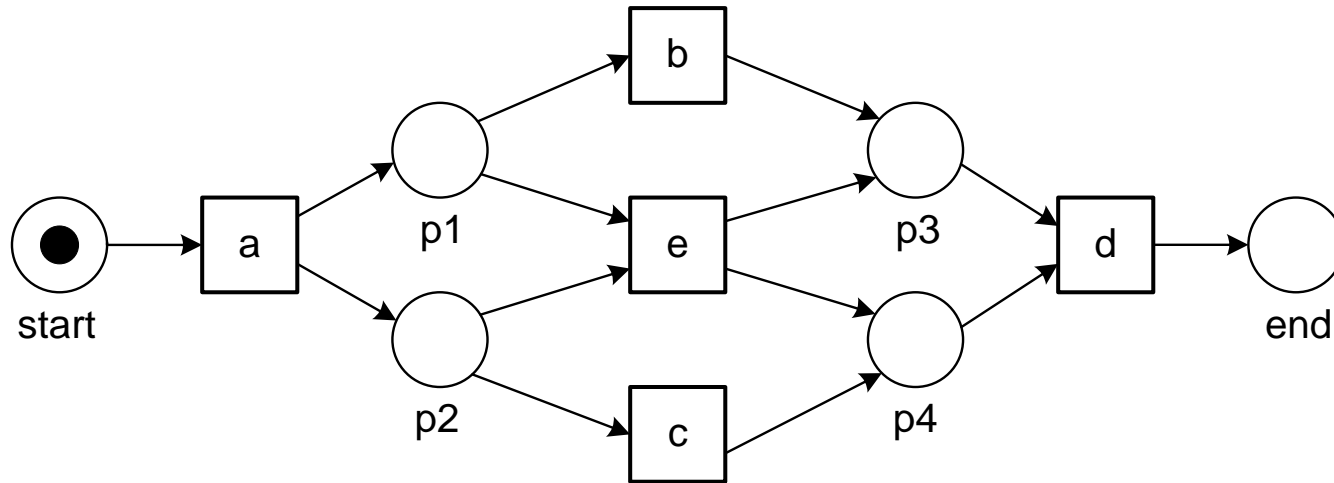
Example Revisited

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



Footprint of L_1

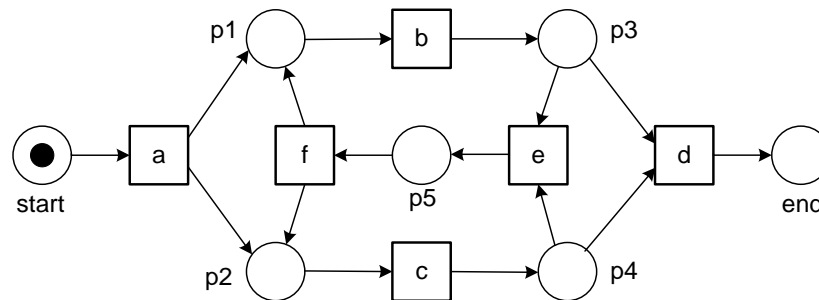
$$L_1 = [\langle a, b, c, d \rangle^3, \langle a, c, b, d \rangle^2, \langle a, e, d \rangle]$$



	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}	$\#_{L_1}$

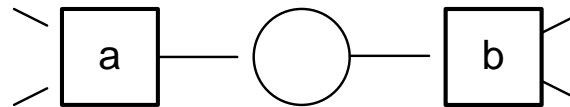
Footprint of L_2

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

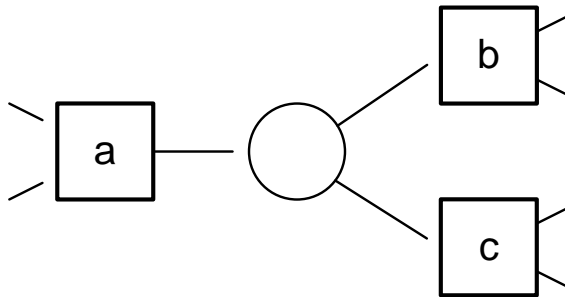


	a	b	c	d	e	f
a	#	\rightarrow	\rightarrow	#	#	#
b	\leftarrow	#	\parallel	\rightarrow	\rightarrow	\leftarrow
c	\leftarrow	\parallel	#	\rightarrow	\rightarrow	\leftarrow
d	#	\leftarrow	\leftarrow	#	#	#
e	#	\leftarrow	\leftarrow	#	#	\rightarrow
f	#	\rightarrow	\rightarrow	#	\leftarrow	#

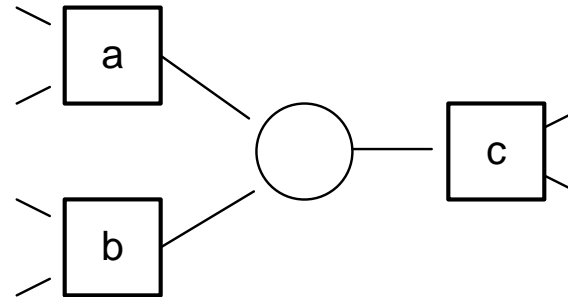
Simple patterns



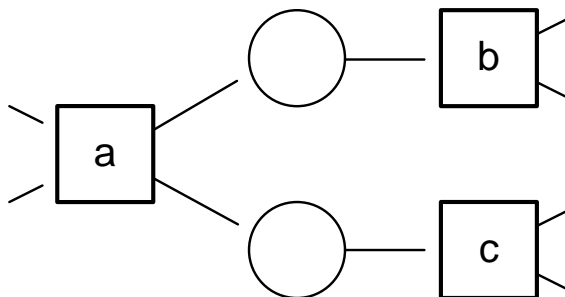
(a) sequence pattern: $a \rightarrow b$



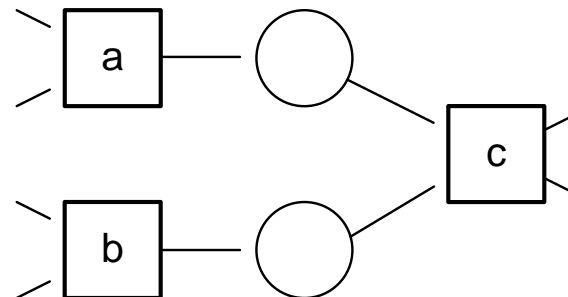
(b) XOR-split pattern:
 $a \rightarrow b$, $a \rightarrow c$, and $b \# c$



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



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



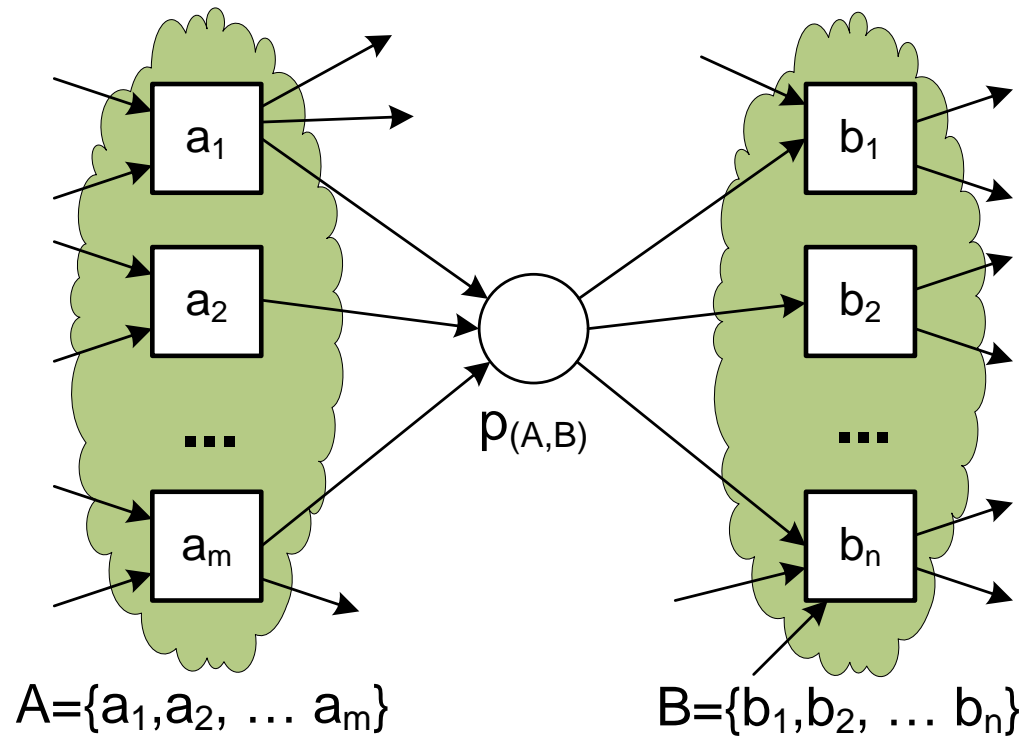
(e) AND-join pattern:
 $a \rightarrow c$, $b \rightarrow c$, and $a \parallel b$

Algorithm

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

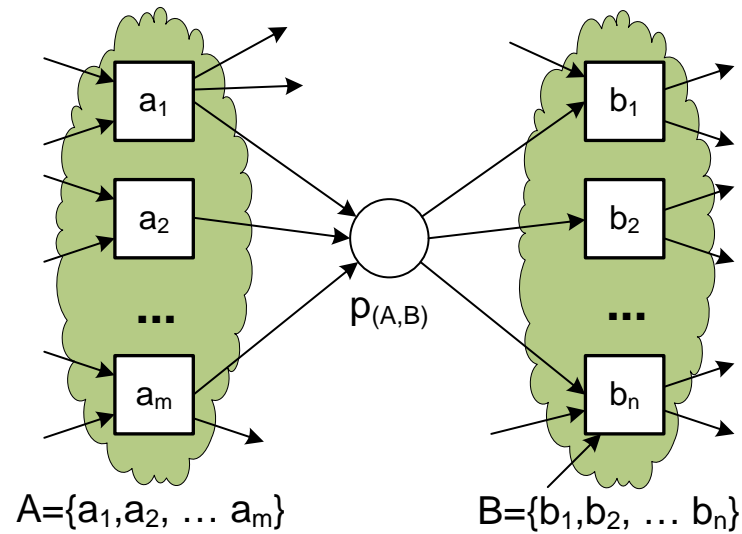
1. $T_L = \{ t \in T \mid \exists_{\sigma \in L} t \in \sigma \},$
2. $T_I = \{ t \in T \mid \exists_{\sigma \in L} t = \text{first}(\sigma) \},$
3. $T_O = \{ t \in T \mid \exists_{\sigma \in L} t = \text{last}(\sigma) \},$
4. $X_L = \{ (A,B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B} a \rightarrow_L b \wedge \forall_{a_1, a_2 \in A} a_1 \#_L a_2 \wedge \forall_{b_1, b_2 \in B} b_1 \#_L b_2 \},$
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. $Y_L = \{ (A,B) \in X_L \mid \forall_{(A',B') \in X_L} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \},$

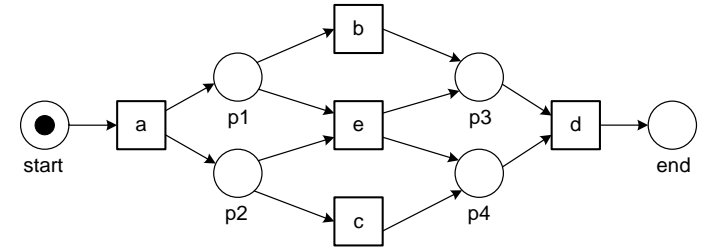
Places as footprints



	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

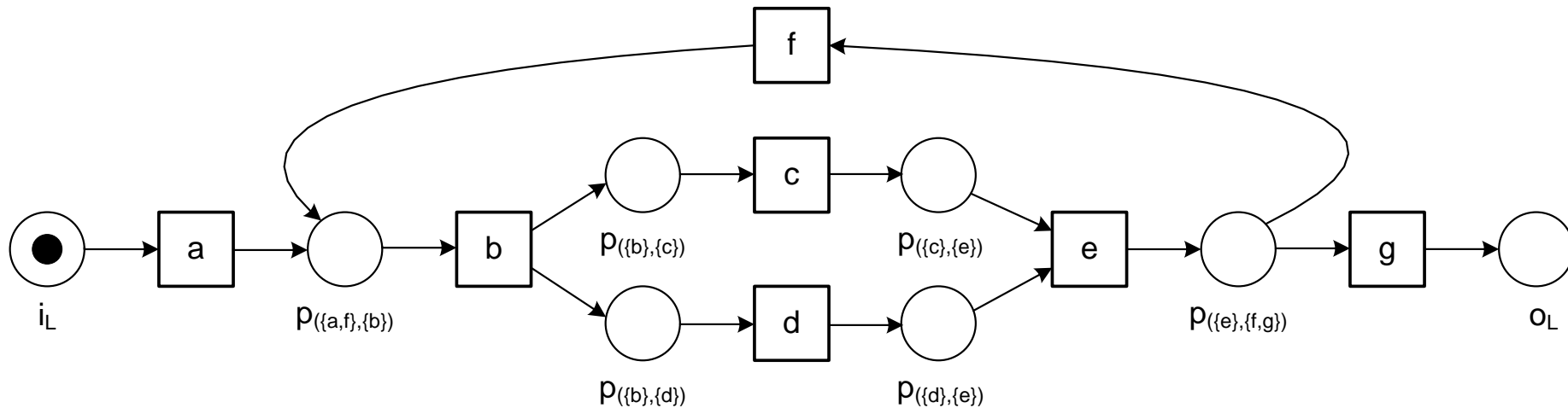
$$L_3 = [\langle a, b, c, d, e, f, b, d, c, e, g \rangle, \\ \langle a, b, d, c, e, g \rangle^2, \\ \langle a, b, c, d, e, f, b, c, d, e, f, b, d, c, e, g \rangle]$$

	<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

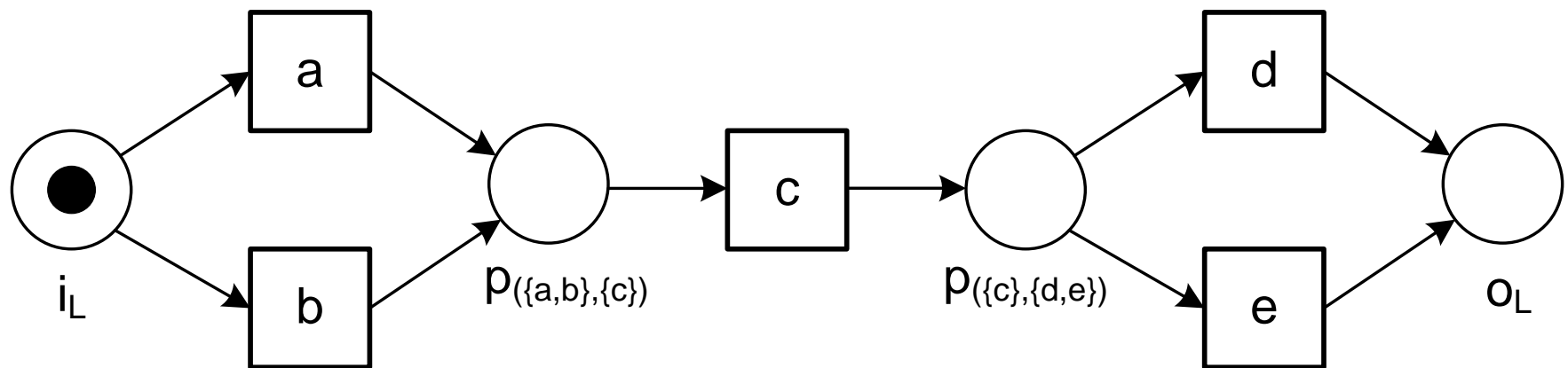
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
<i>a</i>	#	→	#	#	#	#	#
<i>b</i>	←	#	→	→	#	←	#
<i>c</i>	#	←	#		→	#	#
<i>d</i>	#	←		#	→	#	#
<i>e</i>	#	#	←	←	#	→	→
<i>f</i>	#	→	#	#	←	#	#
<i>g</i>	#	#	#	#	←	#	#

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



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 L₅

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

	<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>	#	←	#	#	←	#

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

$$T_I = \{a\}$$

$$T_I = \{f\}$$

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

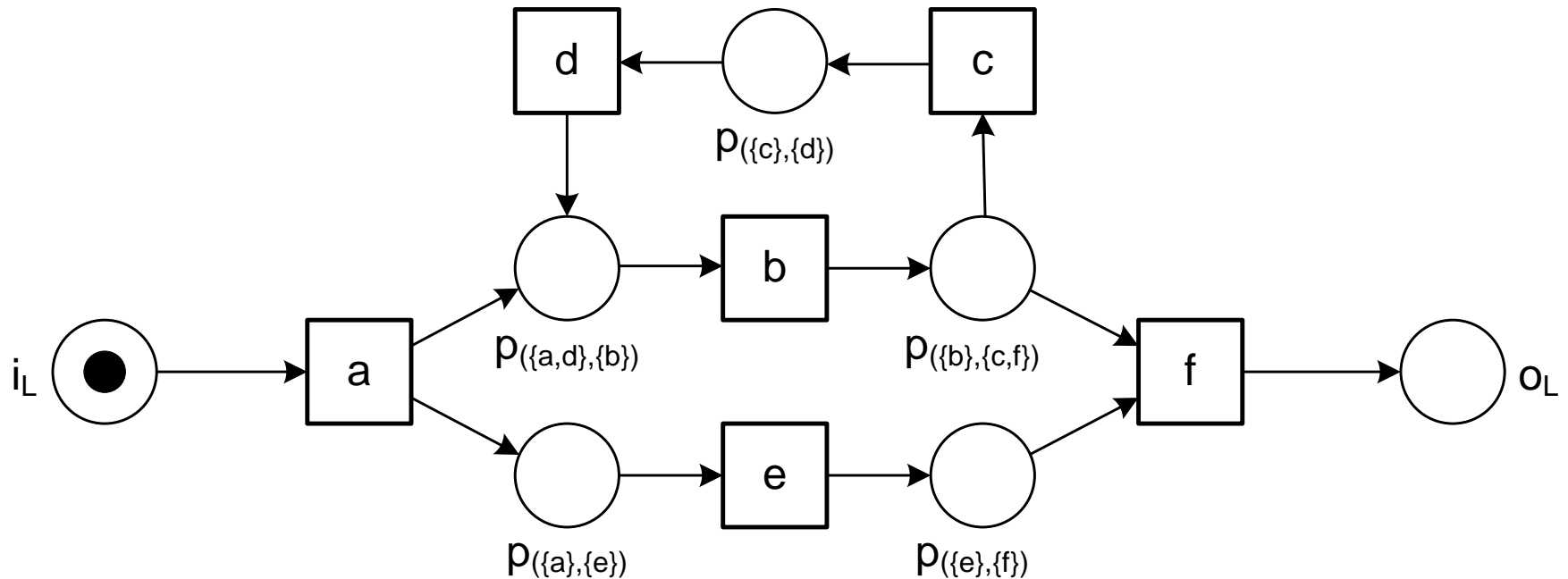
$$Y_L = \{(\{a\}, \{e\}), (\{c\}, \{d\}), (\{e\}, \{f\}), (\{a, d\}, \{b\}), (\{b\}, \{c, f\})\}$$

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

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

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

Discovered model



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

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

Limitations of the α Algorithm



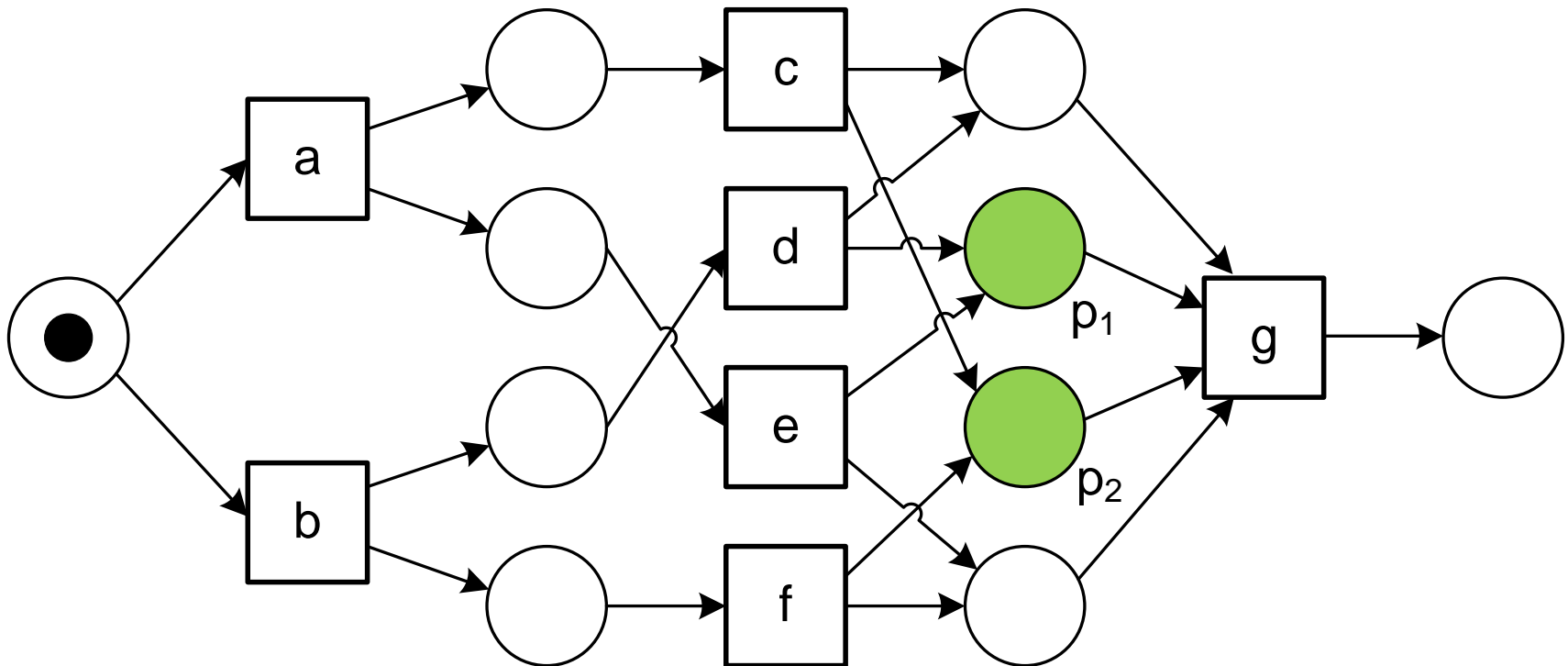
TU/e

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Limitation of α algorithm (implicit places)

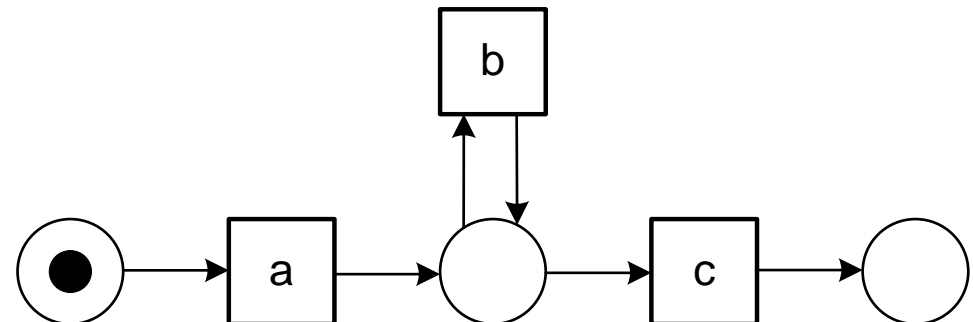
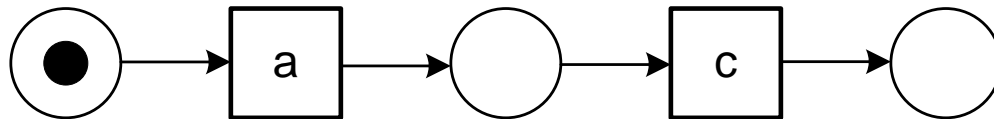
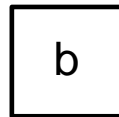
$$L_6 = [\langle a, c, e, g \rangle^2, \langle a, e, c, g \rangle^3, \langle b, d, f, g \rangle^2, \langle b, f, d, g \rangle^4]$$



Green places are implicit!

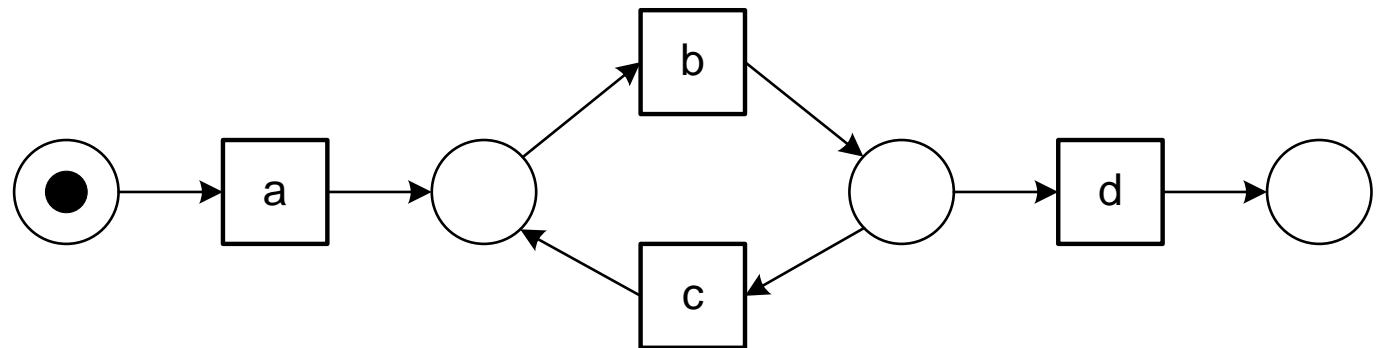
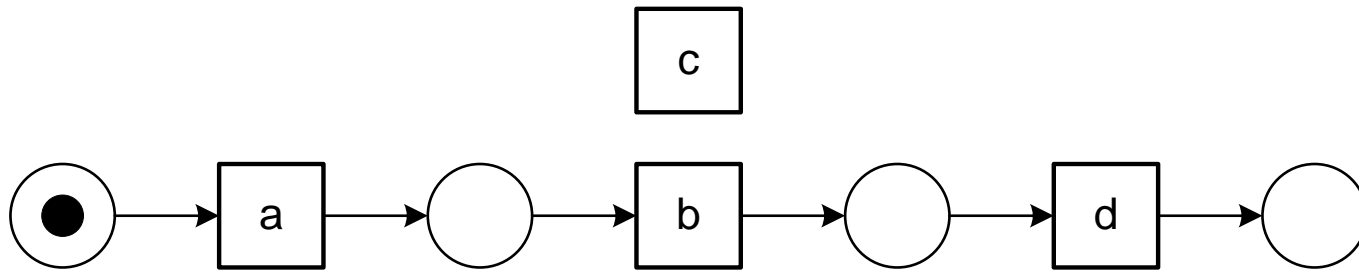
Limitation of α algorithm (loops of length 1)

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



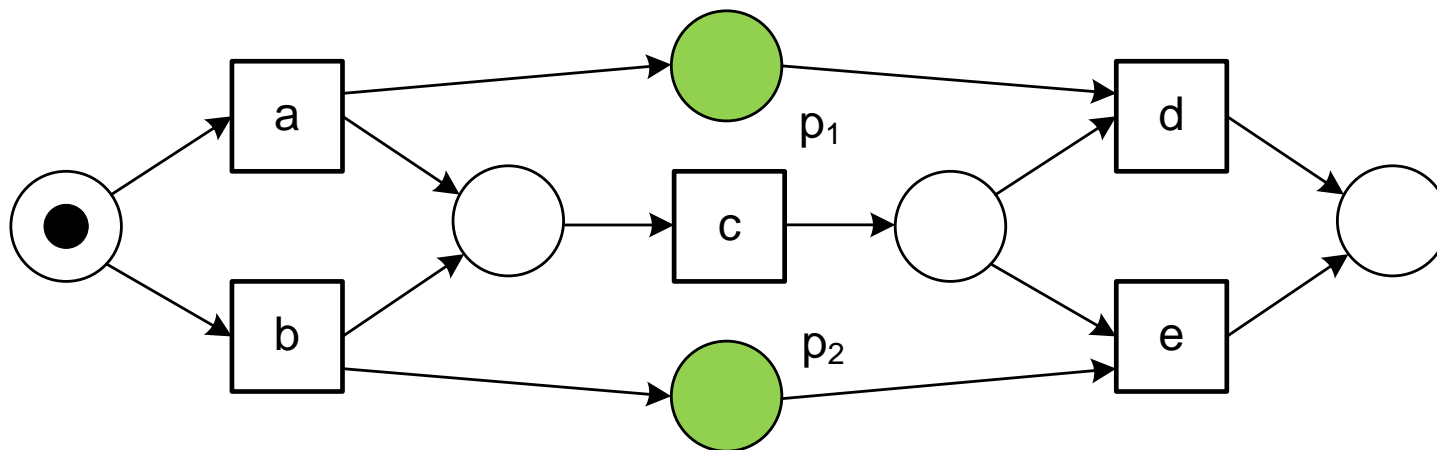
Limitation of α algorithm (loops of length 2)

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



Limitation of α algorithm (non-local dependencies)

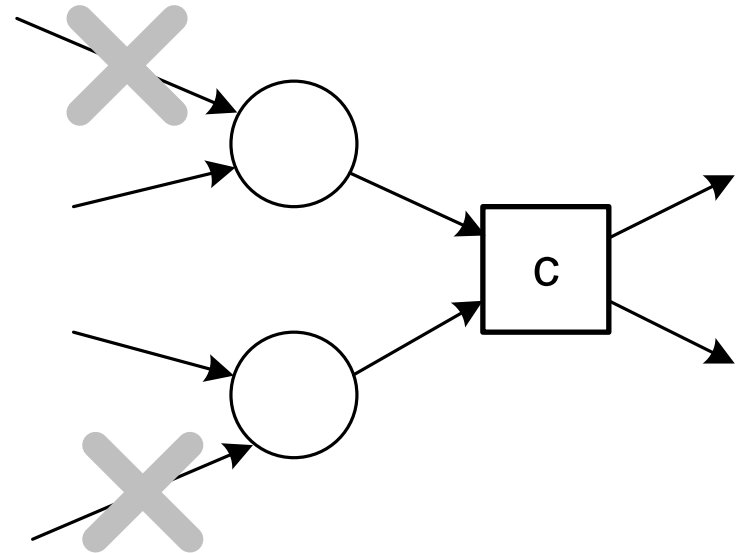
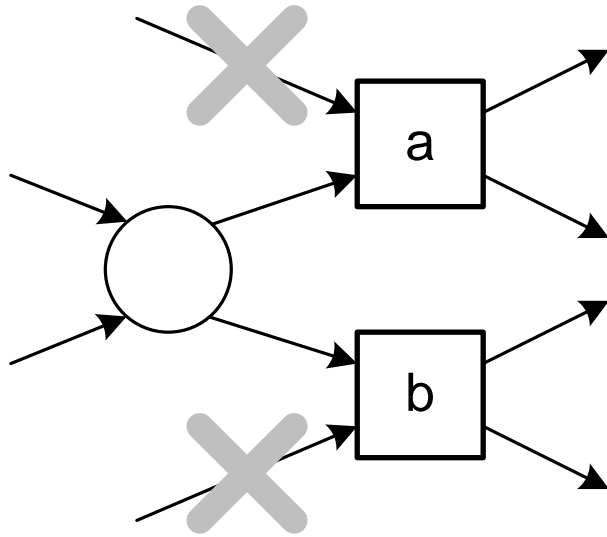
$$L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}]$$



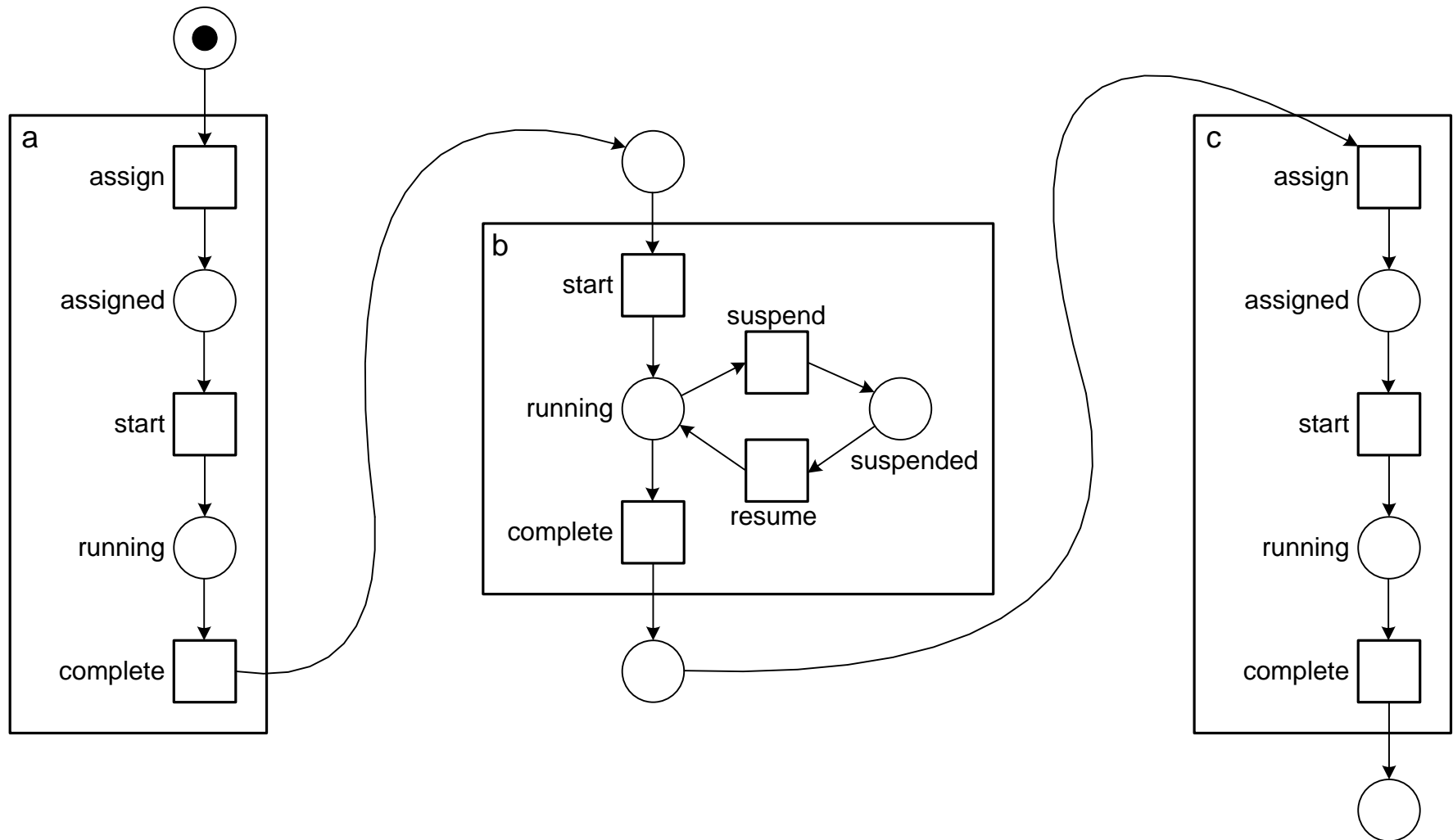
Green places are not discovered!

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

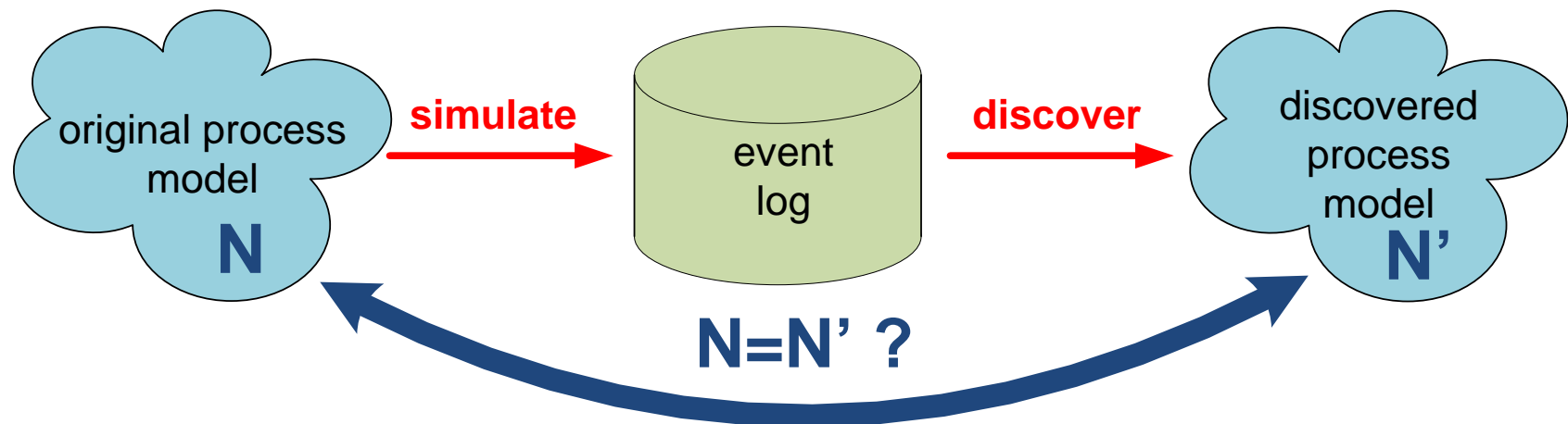
Difficult constructs for α algorithm



Taking the transactional life-cycle into account



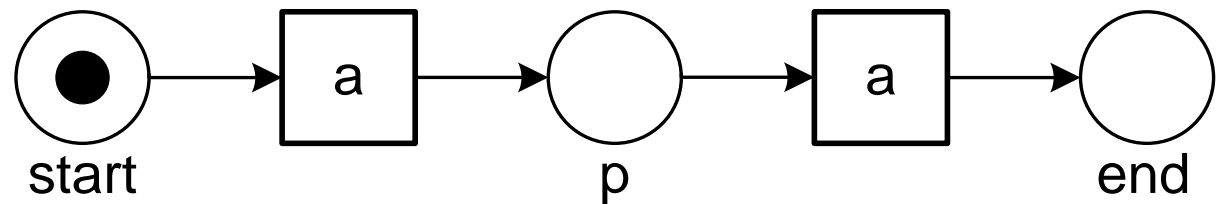
Rediscovering process models



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

Challenge: finding the right representational bias

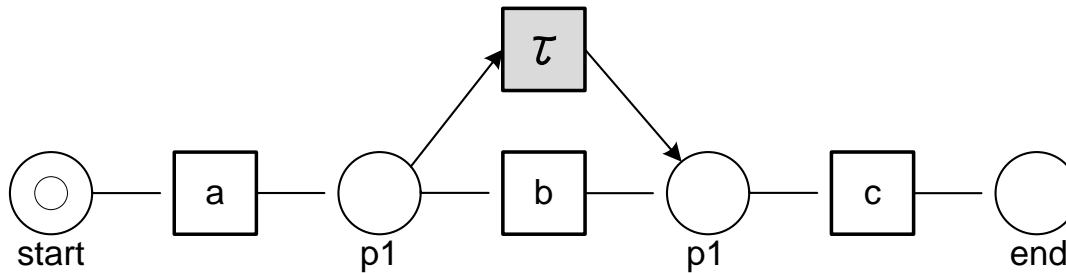
$$L_{10} = [\langle a, a \rangle^{55}]$$



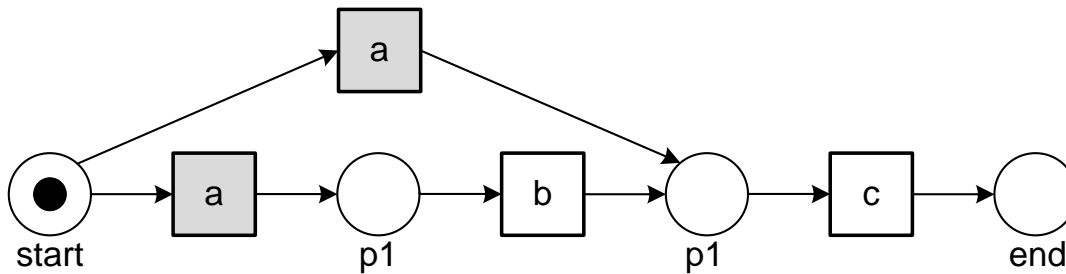
There is no WF-net with unique visible labels that exhibits this behavior.

Another example

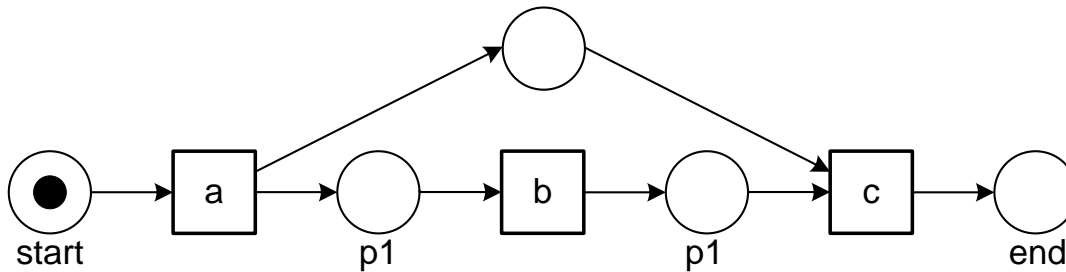
$$L_{11} = [\langle a, b, c \rangle^{20}, \langle a, c \rangle^{30}]$$



(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

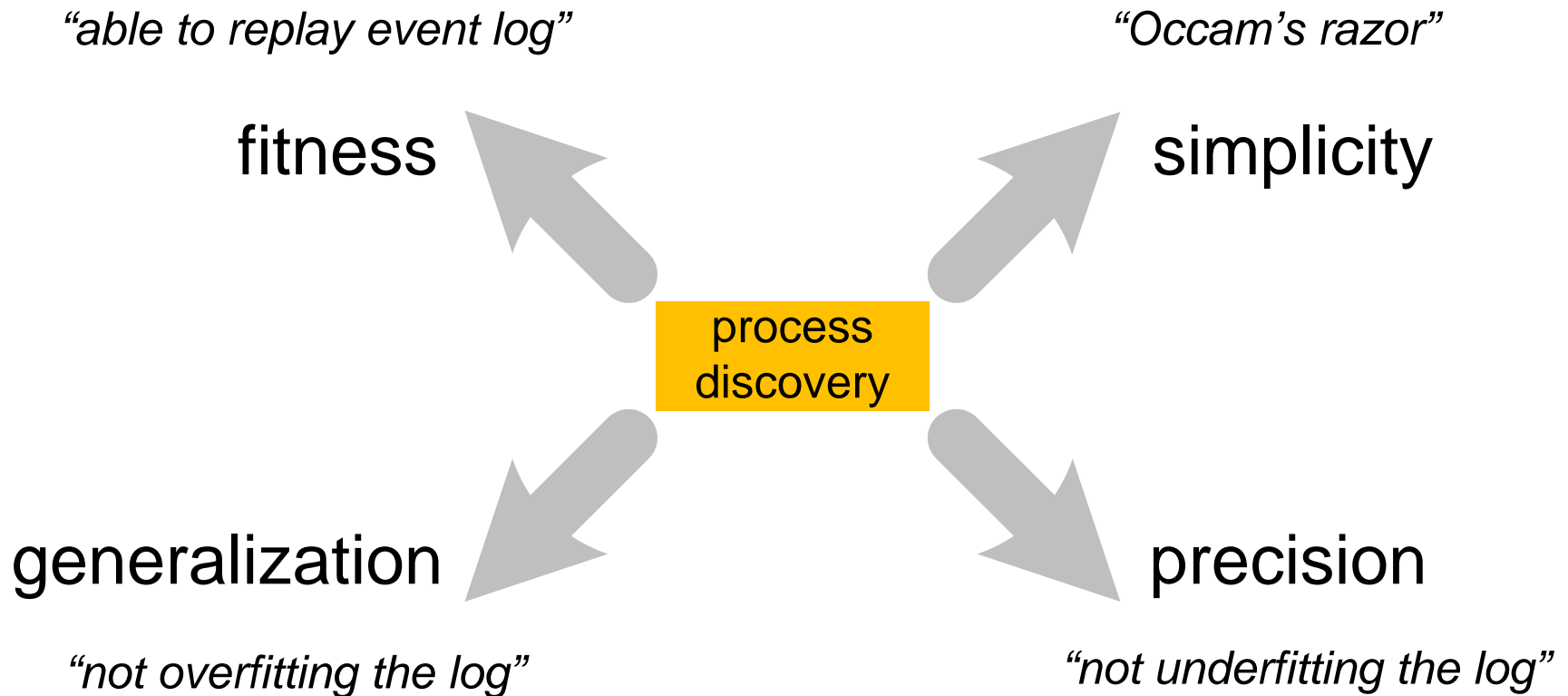
To illustrate the relevance of completeness, consider a process consisting of 10 activities that can be executed in parallel and a corresponding log that contains information about 10,000 cases. The total number of possible interleavings in the model with 10 concurrent activities is $10! = 3,628,800$. 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.)

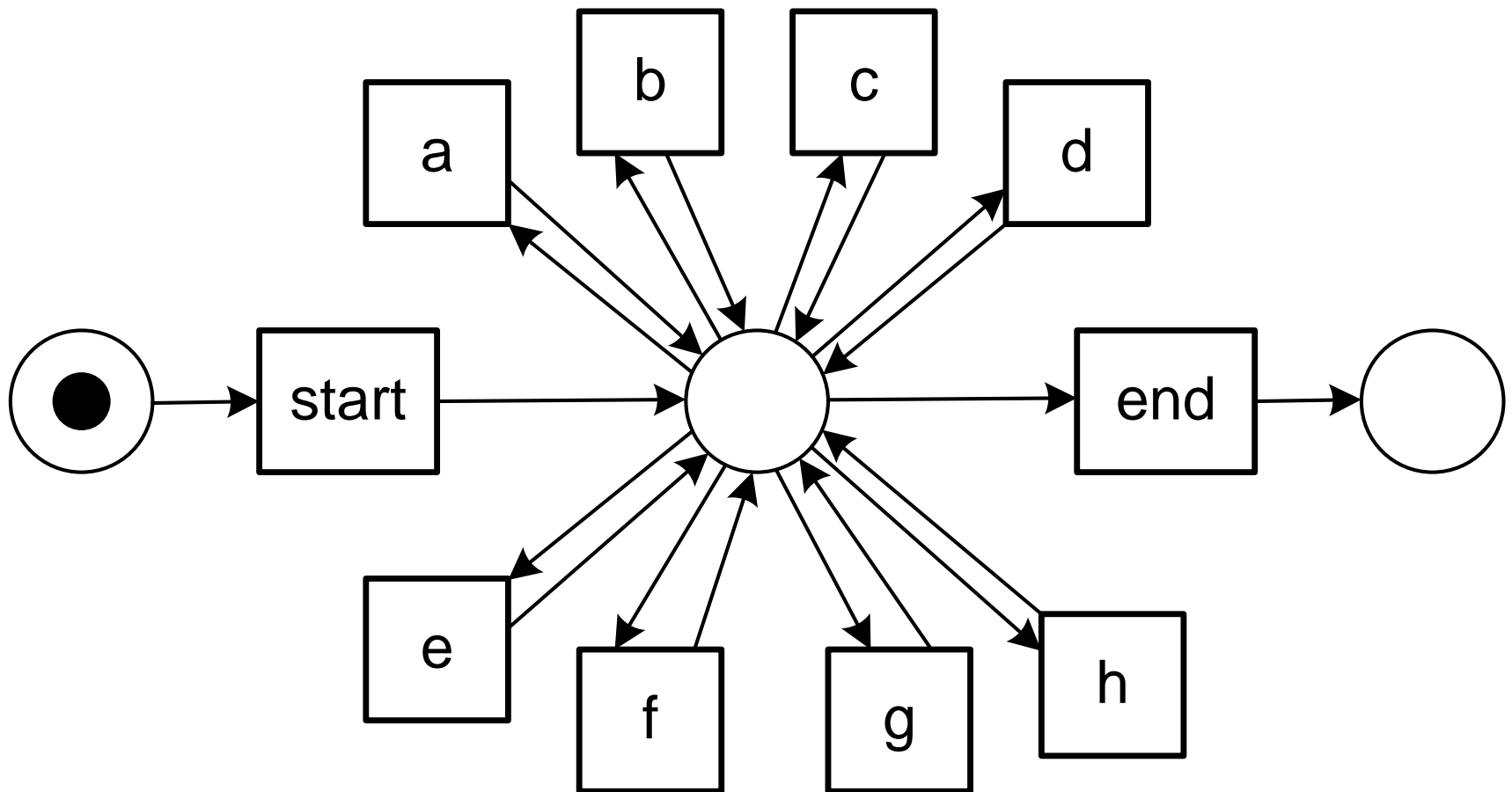


**Challenge: Balancing
Between Underfitting and
Overfitting**

Challenge: four competing quality criteria

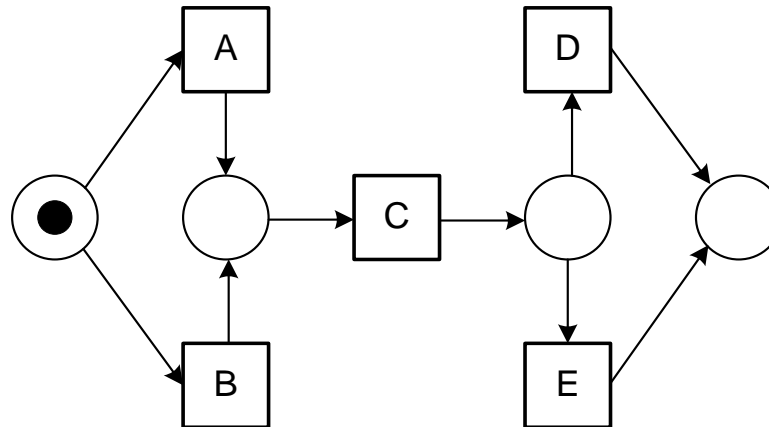
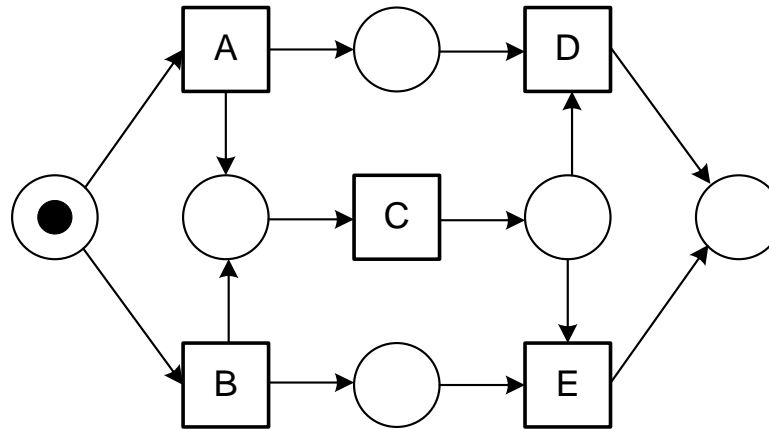


Flower model



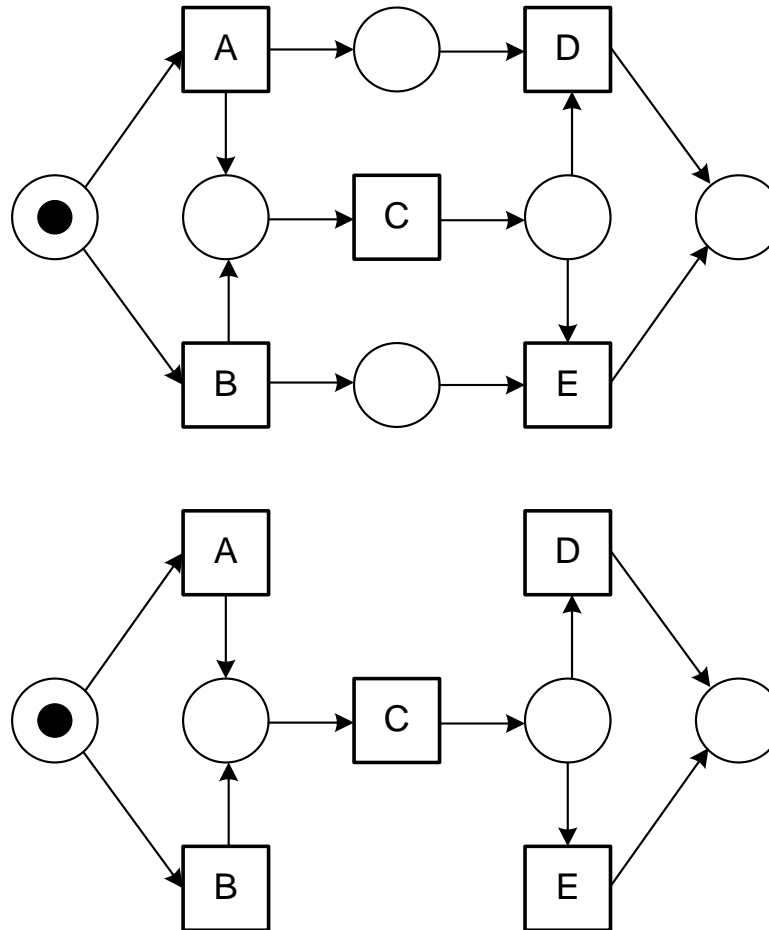
What is the best model?

ACD	99
ACE	0
BCE	85
BCD	0



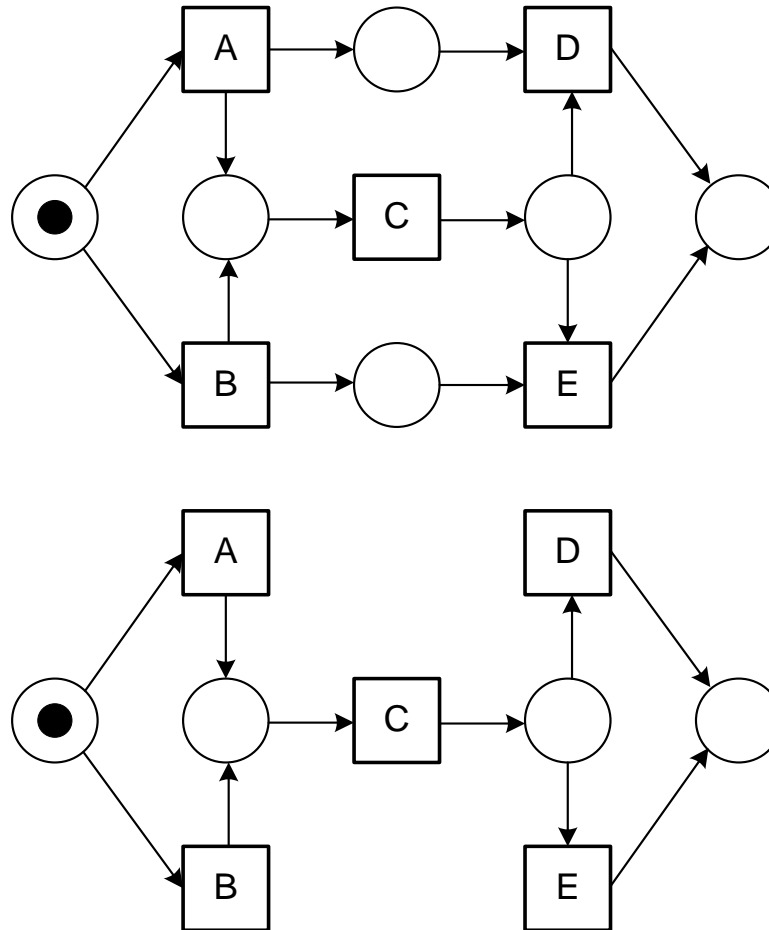
What is the best model?

ACD	99
ACE	88
BCE	85
BCD	78

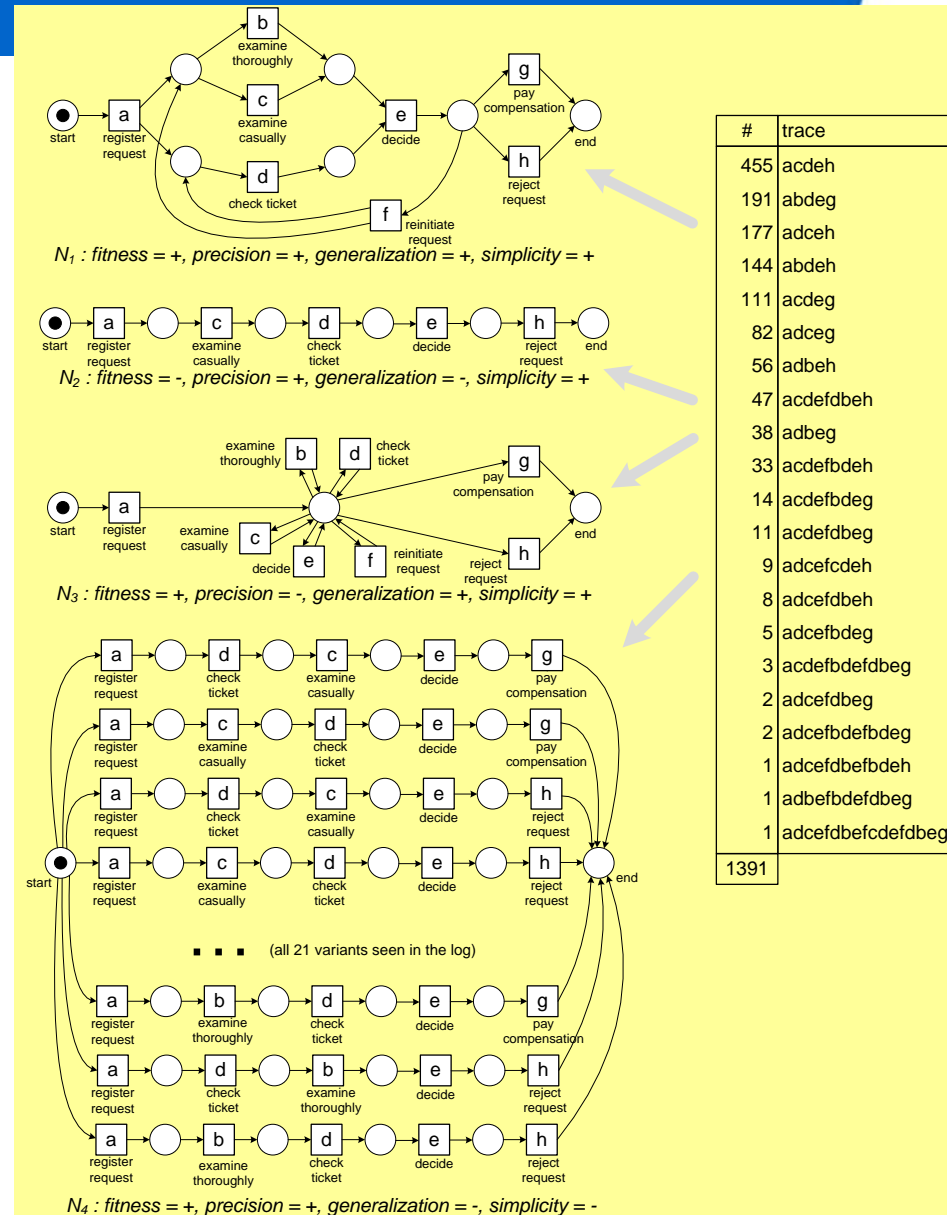
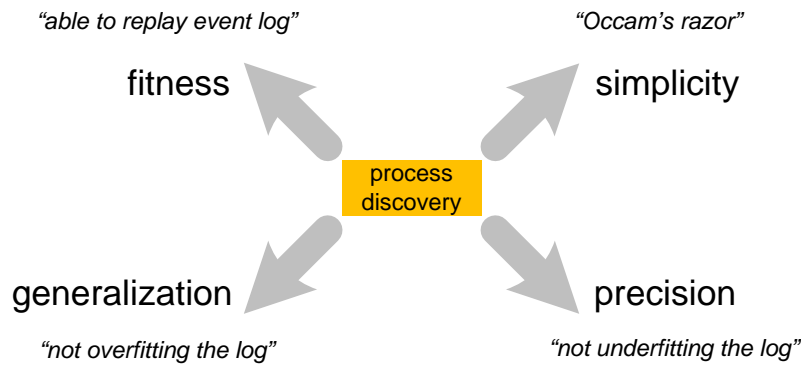


What is the best model?

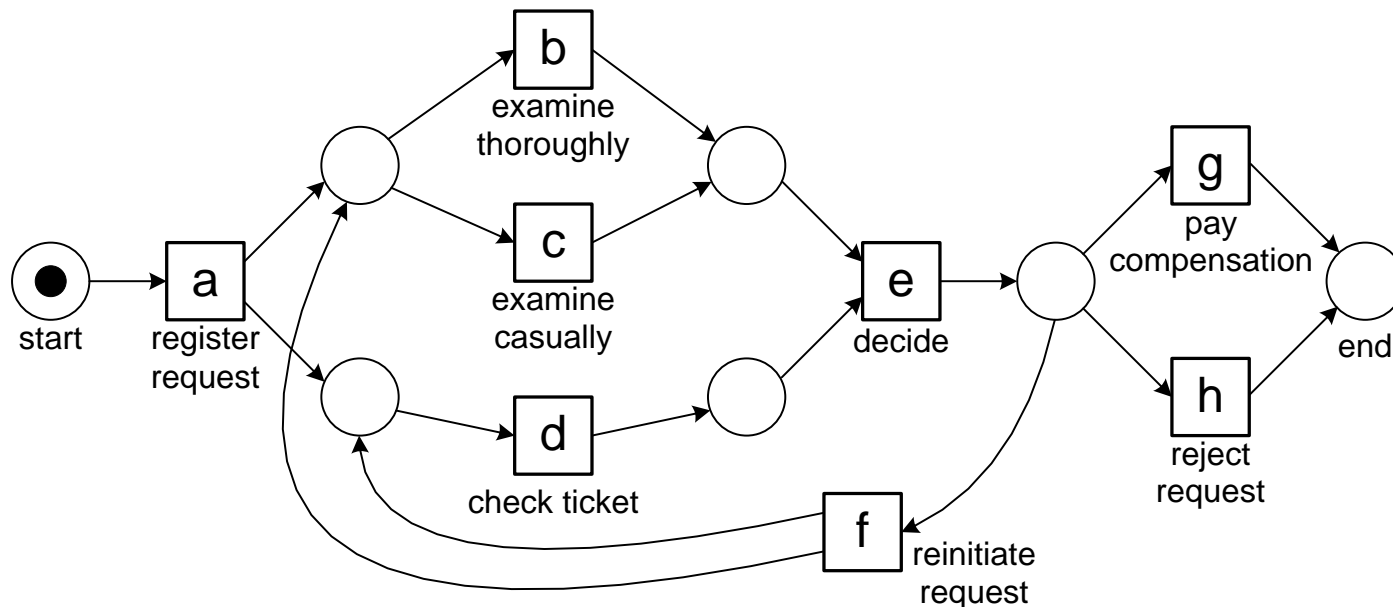
ACD	99
ACE	2
BCE	85
BCD	3



Example: one log four models



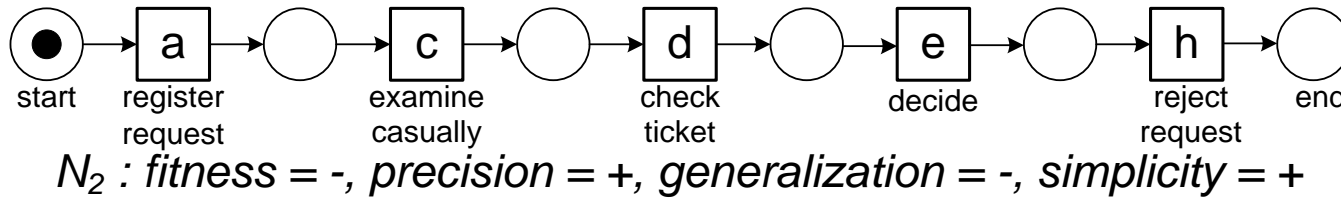
Model N₁



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

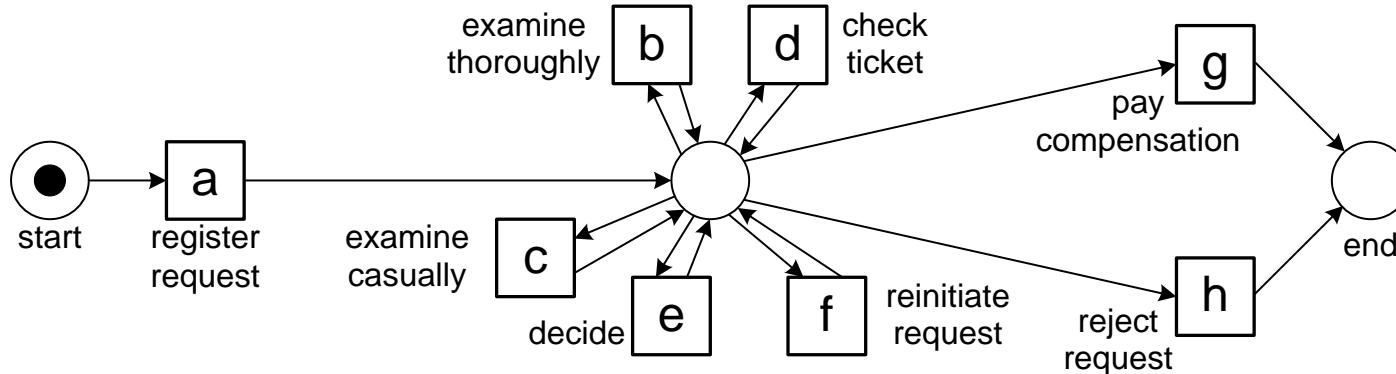
#	trace
455	acdeh
191	abdeg
177	adceh
144	abdeh
111	acdeg
82	adceg
56	adbeh
47	acdefdbeh
38	adbeg
33	acdefbdeh
14	acdefbdeg
11	acdefdbeg
9	adcefcdeh
8	adcefdbeh
5	adcefbdeg
3	acdefbdefdbeg
2	adcefdbeg
2	adcefbdefbdeg
1	adcefdbefbdeh
1	adbefbdefdbeg
1	adcefdbefcdefdbeg
1391	

Model N₂



#	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	

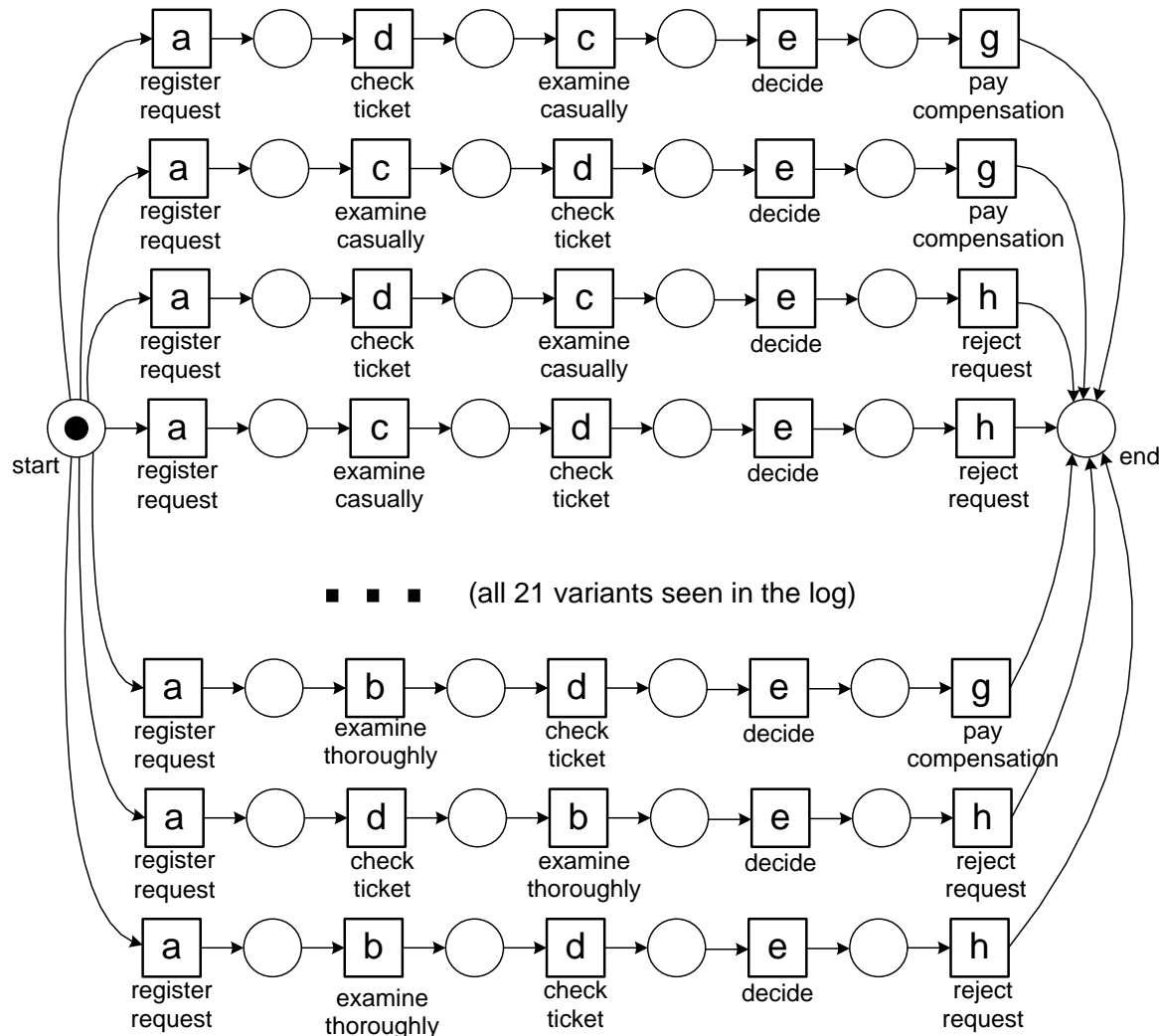
Model N₃



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).

Analyzing Lasagna and Spaghetti Processes



TU/e

Technische Universiteit
Eindhoven
University of Technology

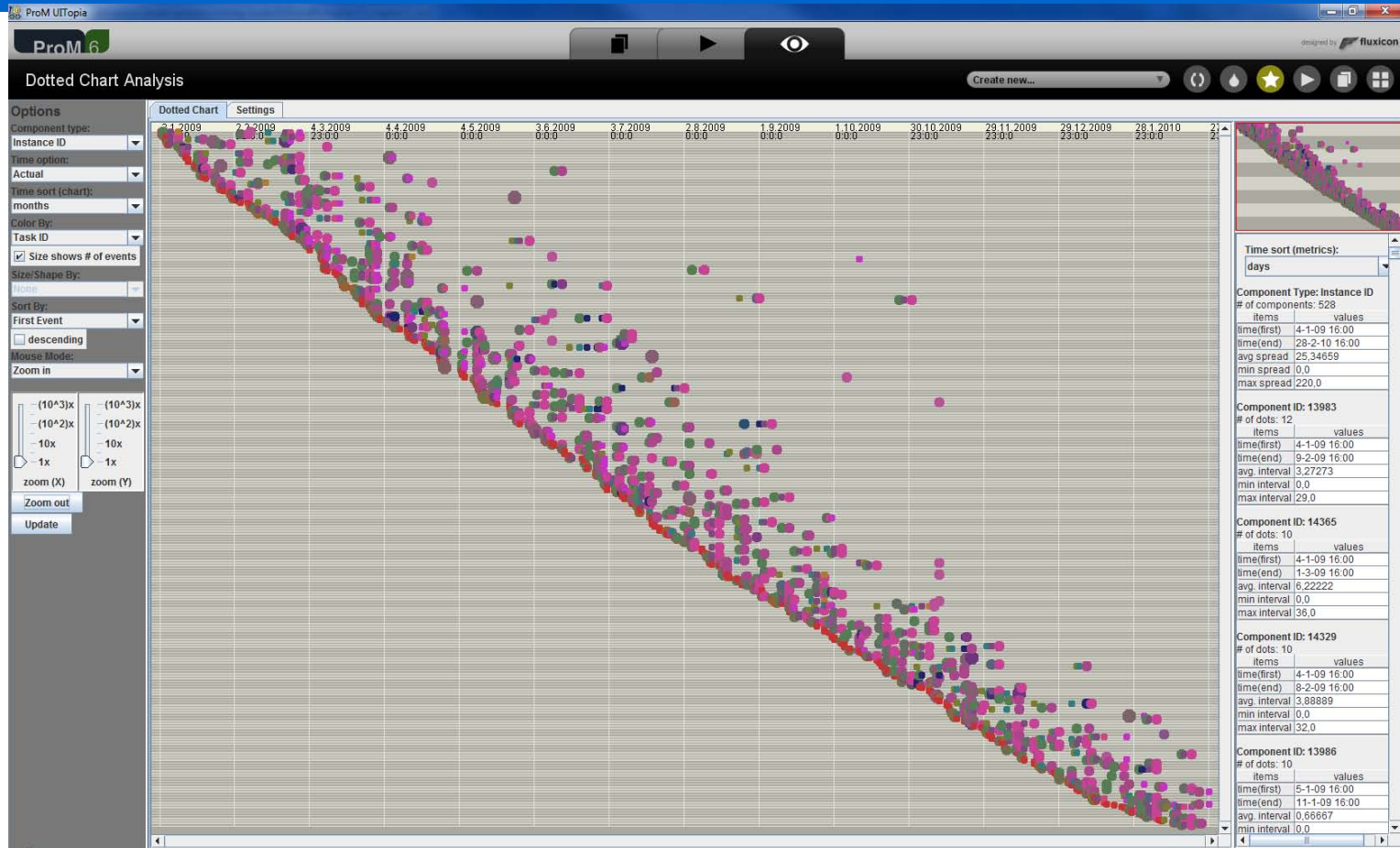
Where innovation starts

How can process mining help?

- Detect bottlenecks
- Detect deviations
- Performance measurement
- Suggest improvements
- Decision support (e.g., recommendation and prediction)

- Provide mirror
- Highlight important problems
- Avoid ICT failures
- Avoid management by PowerPoint
- From “politics” to “analytics”

Example of a Lasagna process: WMO process of a Dutch municipality



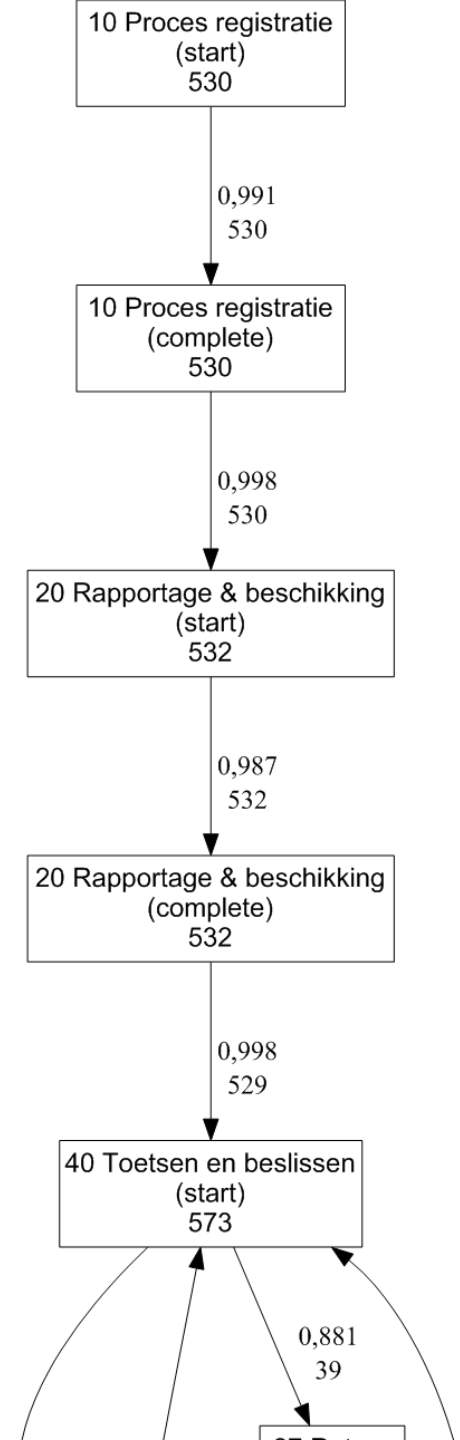
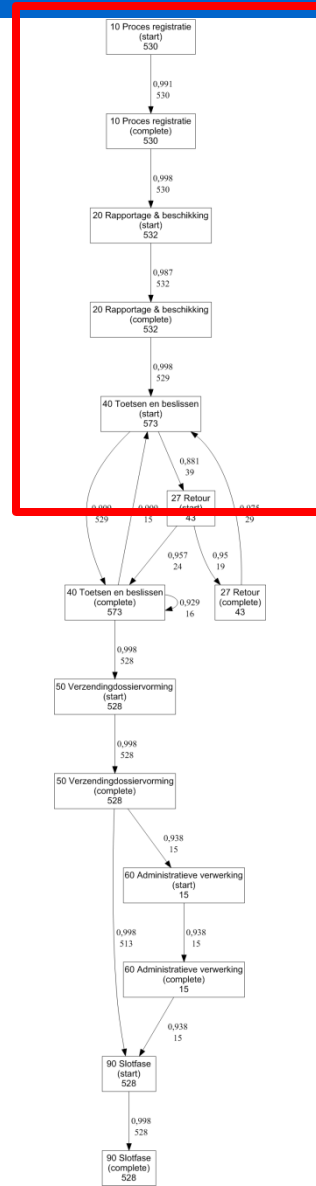
Each line corresponds to one of the 528 requests that were handled in the period from 4-1-2009 until 28-2-2010. In total there are 5498 events represented as dots. The mean time needed to handle a case is approximately 25 days.

WMO process

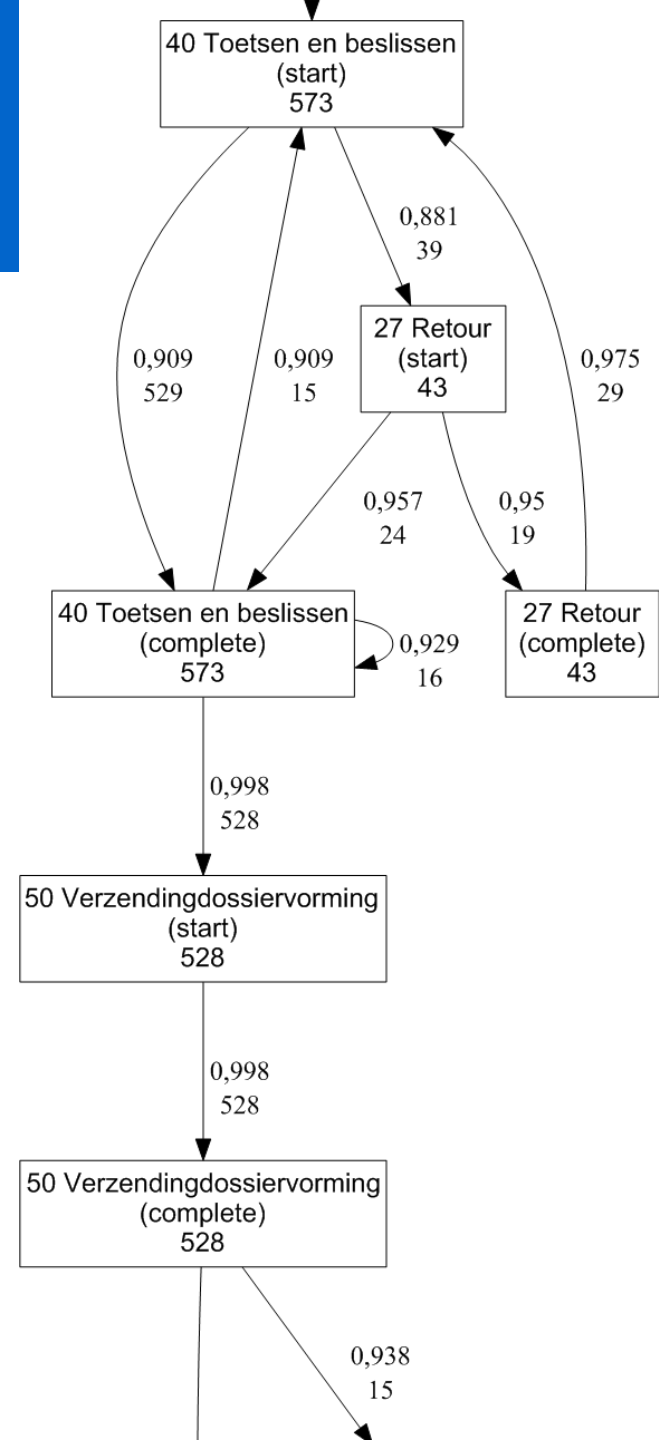
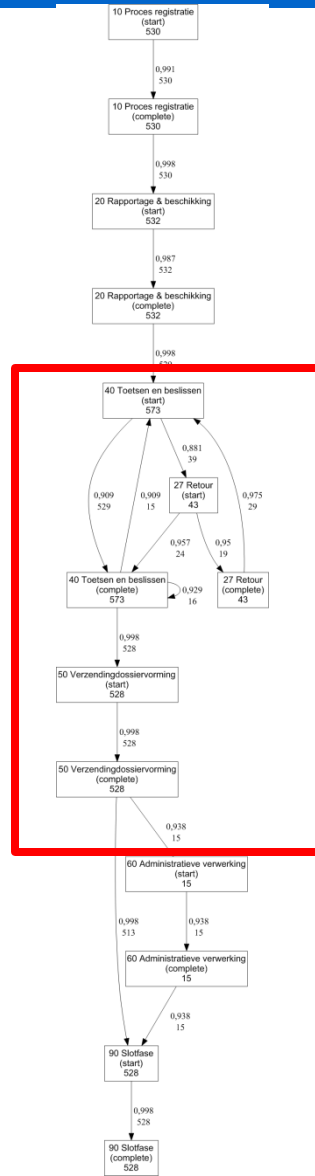
(Wet Maatschappelijke Ondersteuning)

- **WMO refers to the social support act that came into force in The Netherlands on January 1st, 2007.**
- **The aim of this act is to assist people with disabilities and impairments. Under the act, local authorities are required to give support to those who need it, e.g., household help, providing wheelchairs and scootmobiles, and adaptations to homes.**
- **There are different processes for the different kinds of help. We focus on the process for handling requests for household help.**
- **In a period of about one year, 528 requests for household WMO support were received.**
- **These 528 requests generated 5498 events.**

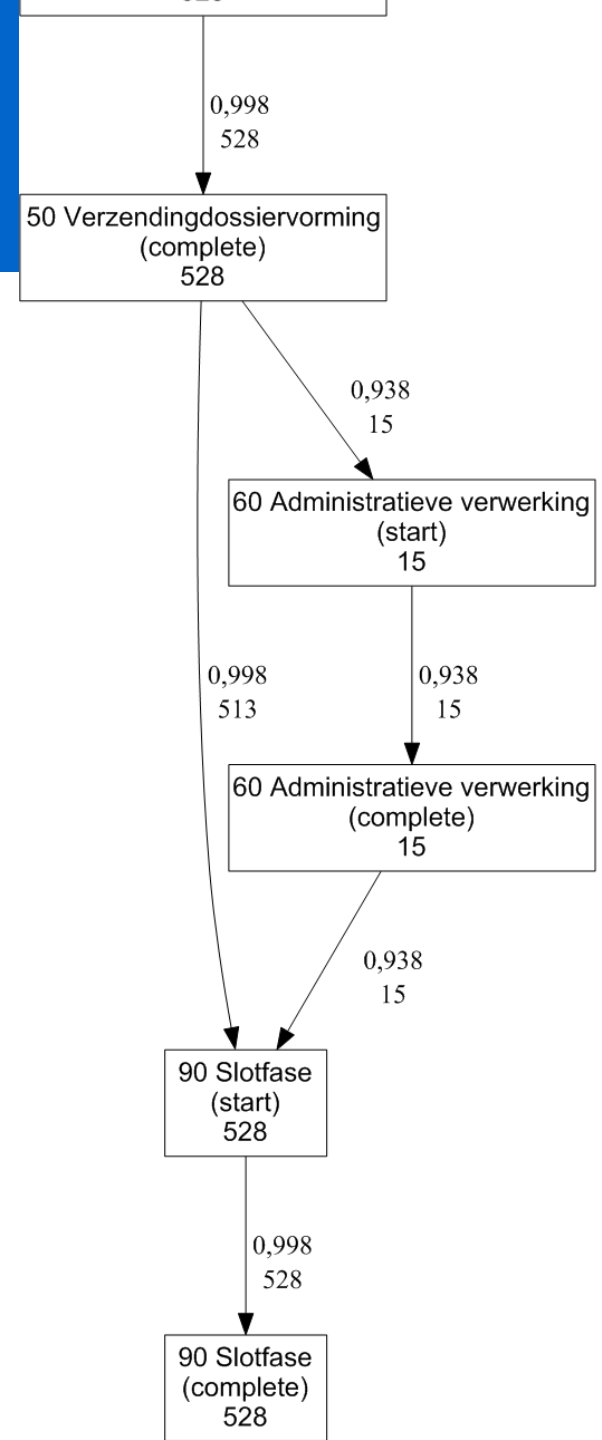
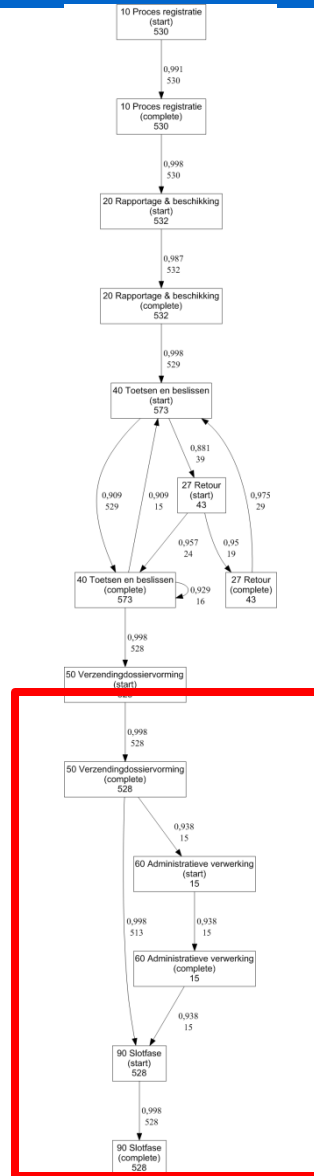
C-net discovered using heuristic miner (1/3)



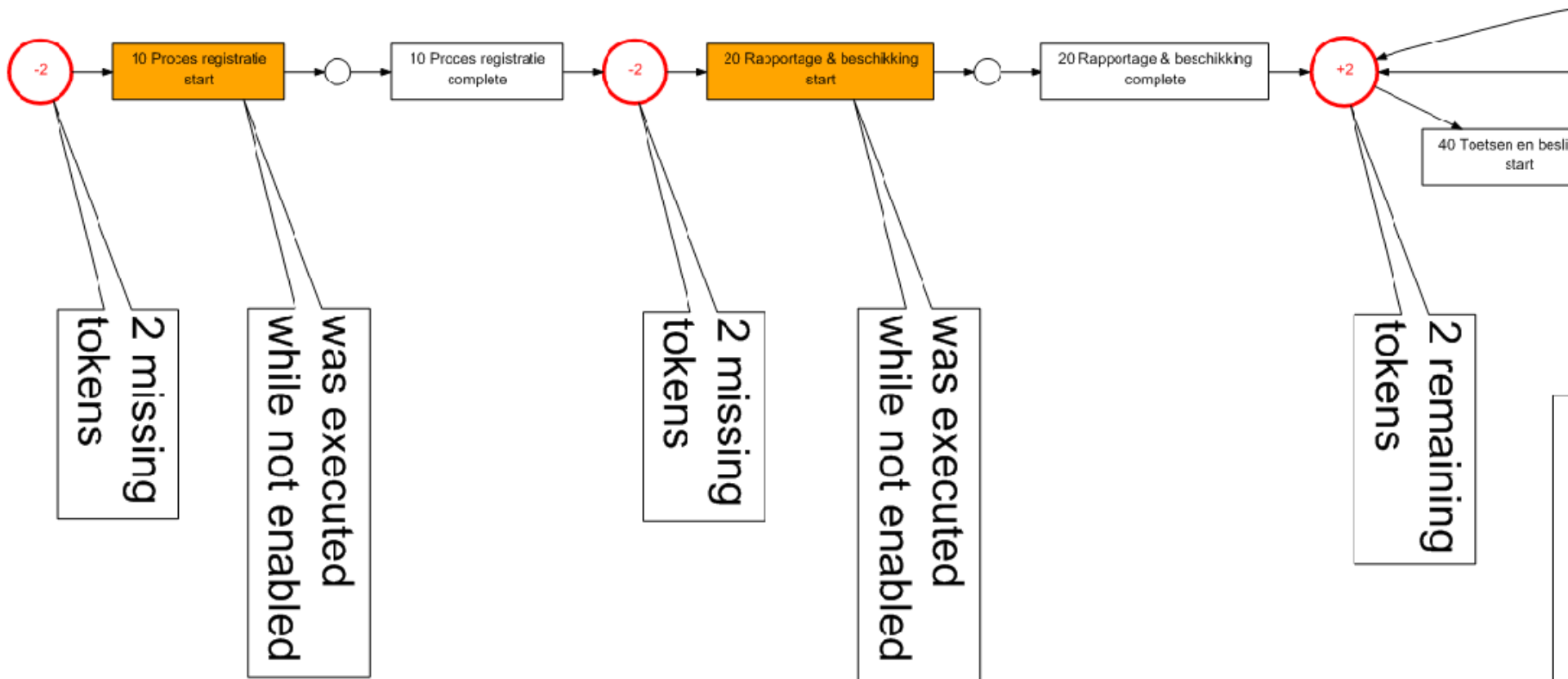
C-net discovered using heuristic miner (2/3)



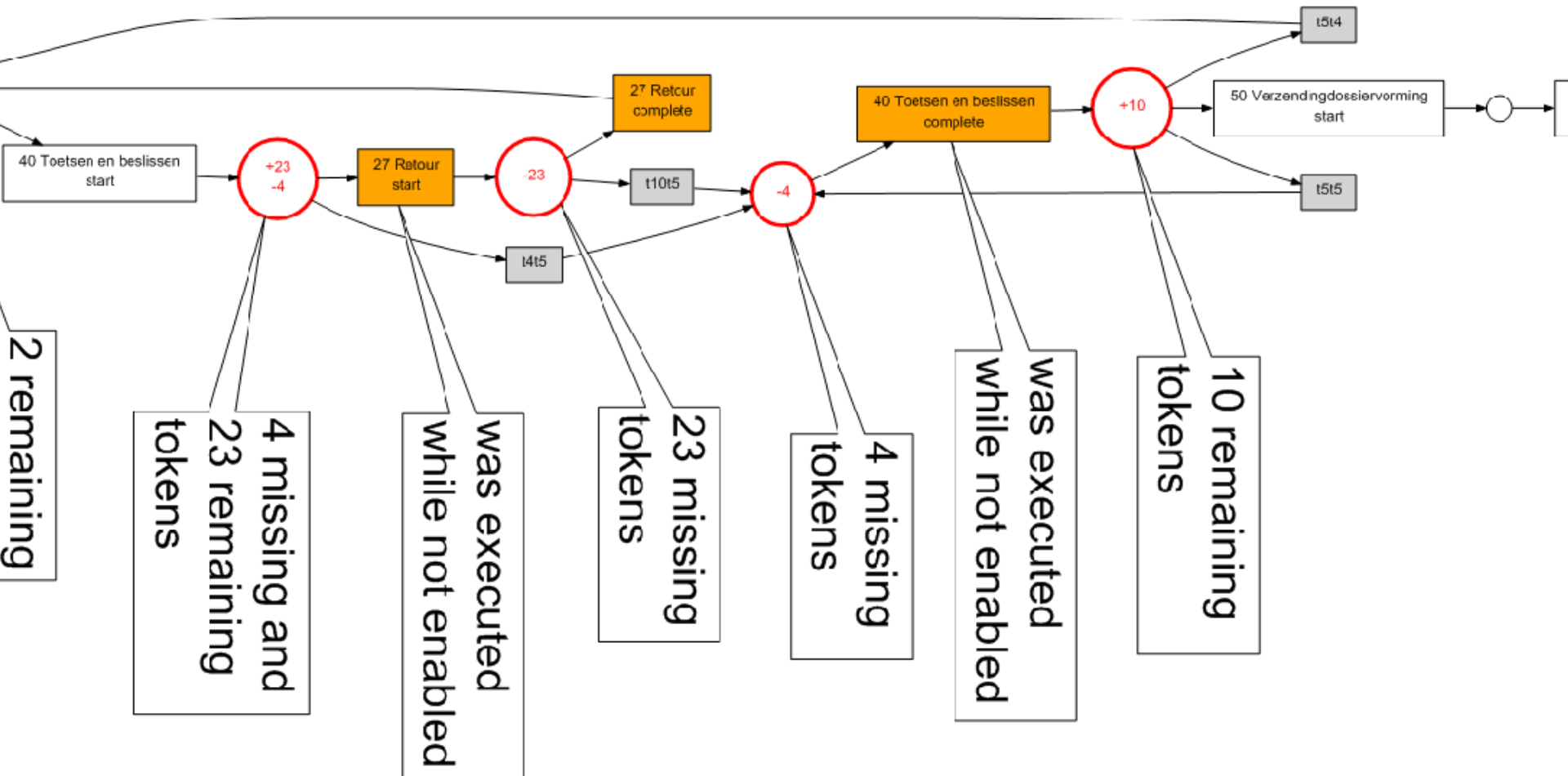
C-net discovered using heuristic miner (3/3)



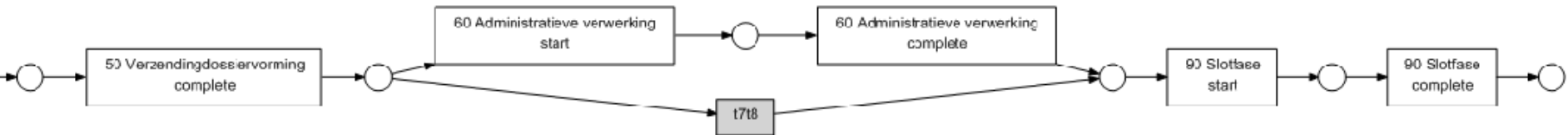
Conformance check WMO process (1/3)



Conformance check WMO process (2/3)

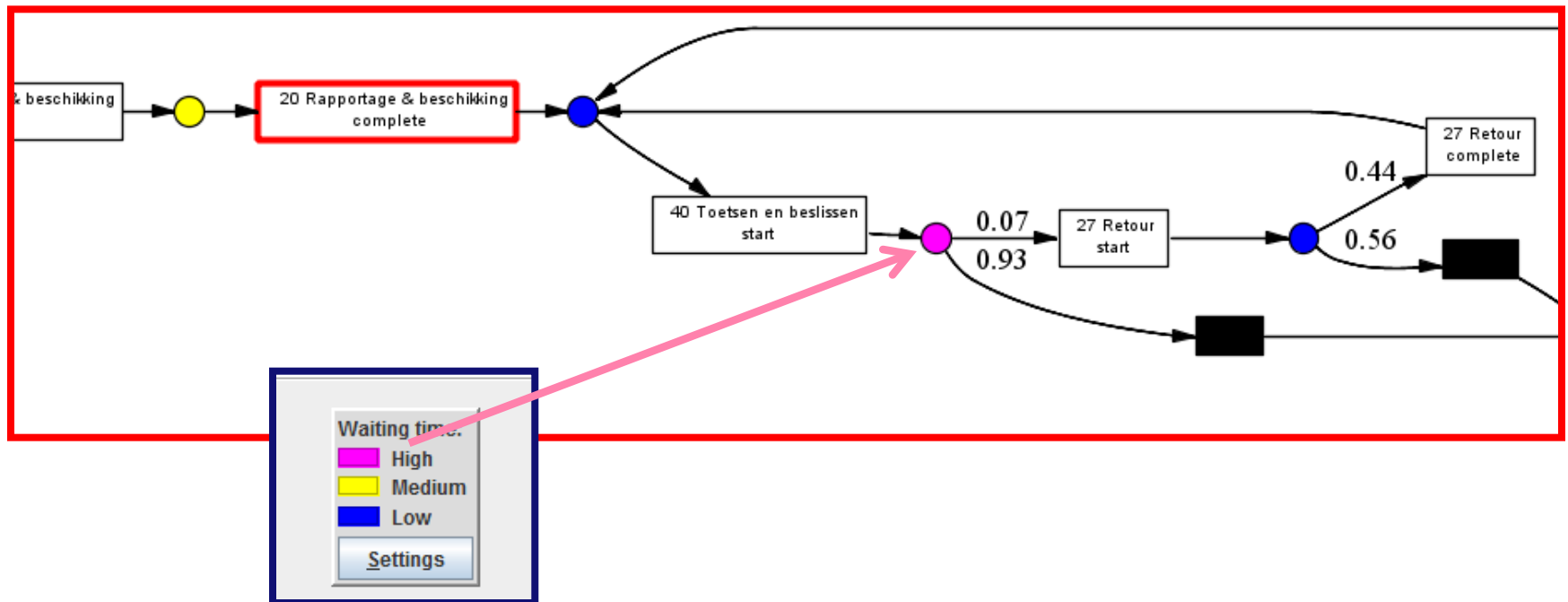
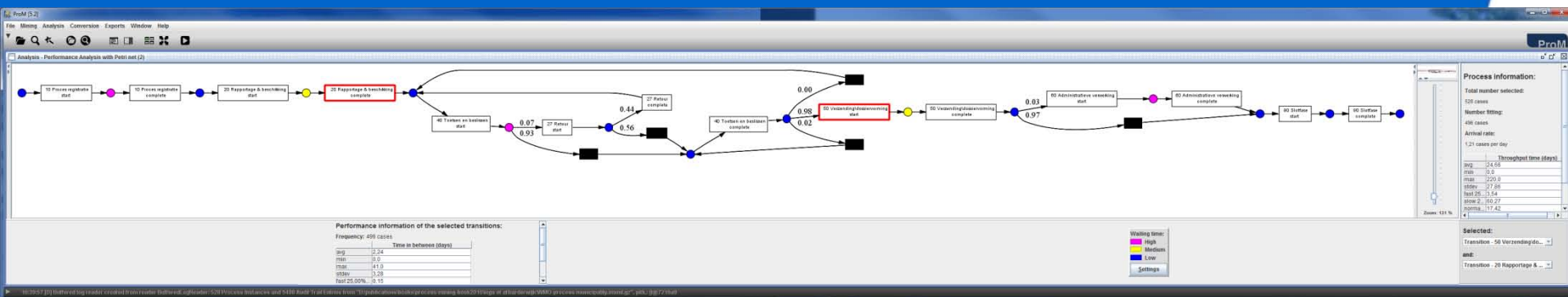


Conformance check WMO process (3/3)



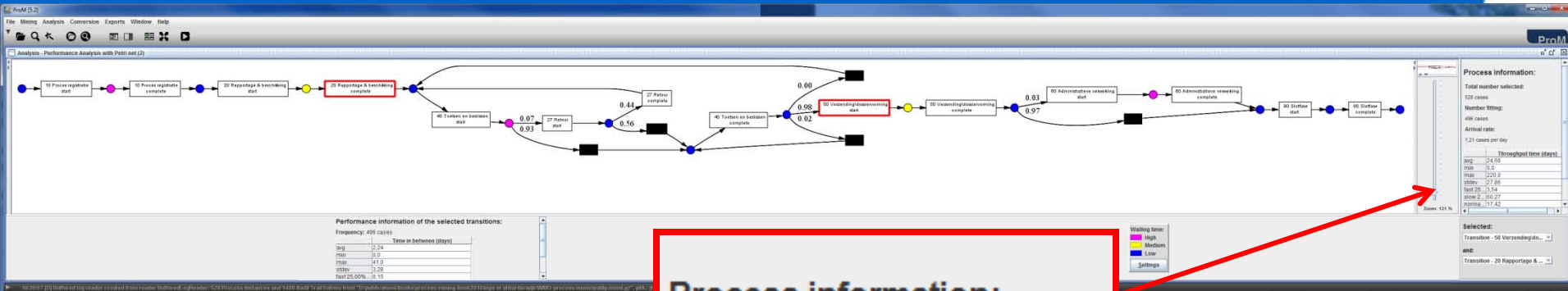
The fitness of the discovered process is 0.99521667. Of the 528 cases, 496 cases fit perfectly whereas for 32 cases there are missing or remaining tokens.

Bottleneck analysis WMO process (1/3)





Bottleneck analysis WMO process (3/3)



Process information:

Total number selected:

528 cases

Number fitting:

496 cases

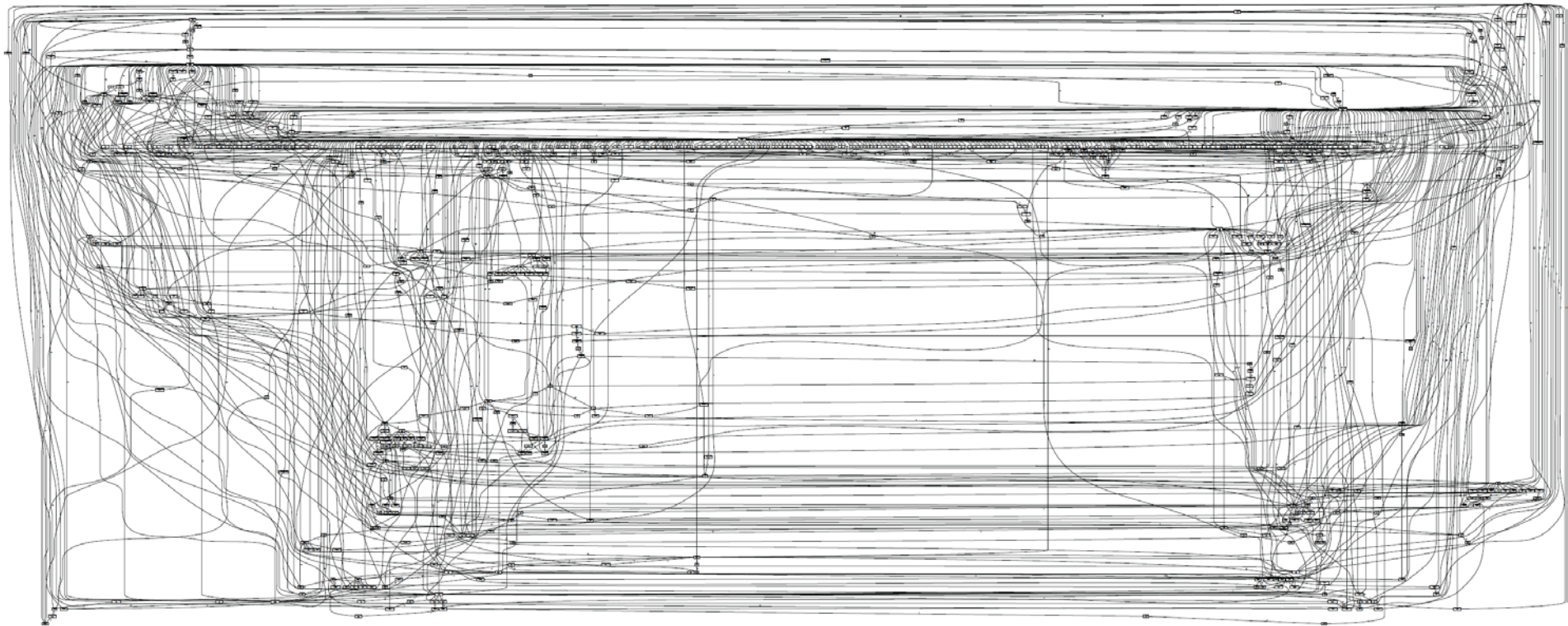
Arrival rate:

1,21 cases per day

	Throughput time (days)
avg	24,66
min	0,0
max	220,0
stdev	27,86
fast 25...	3,54
slow 2...	60,27
norma...	17,42

flow time of
approx. 25 days
with a standard
deviation of
approx. 28

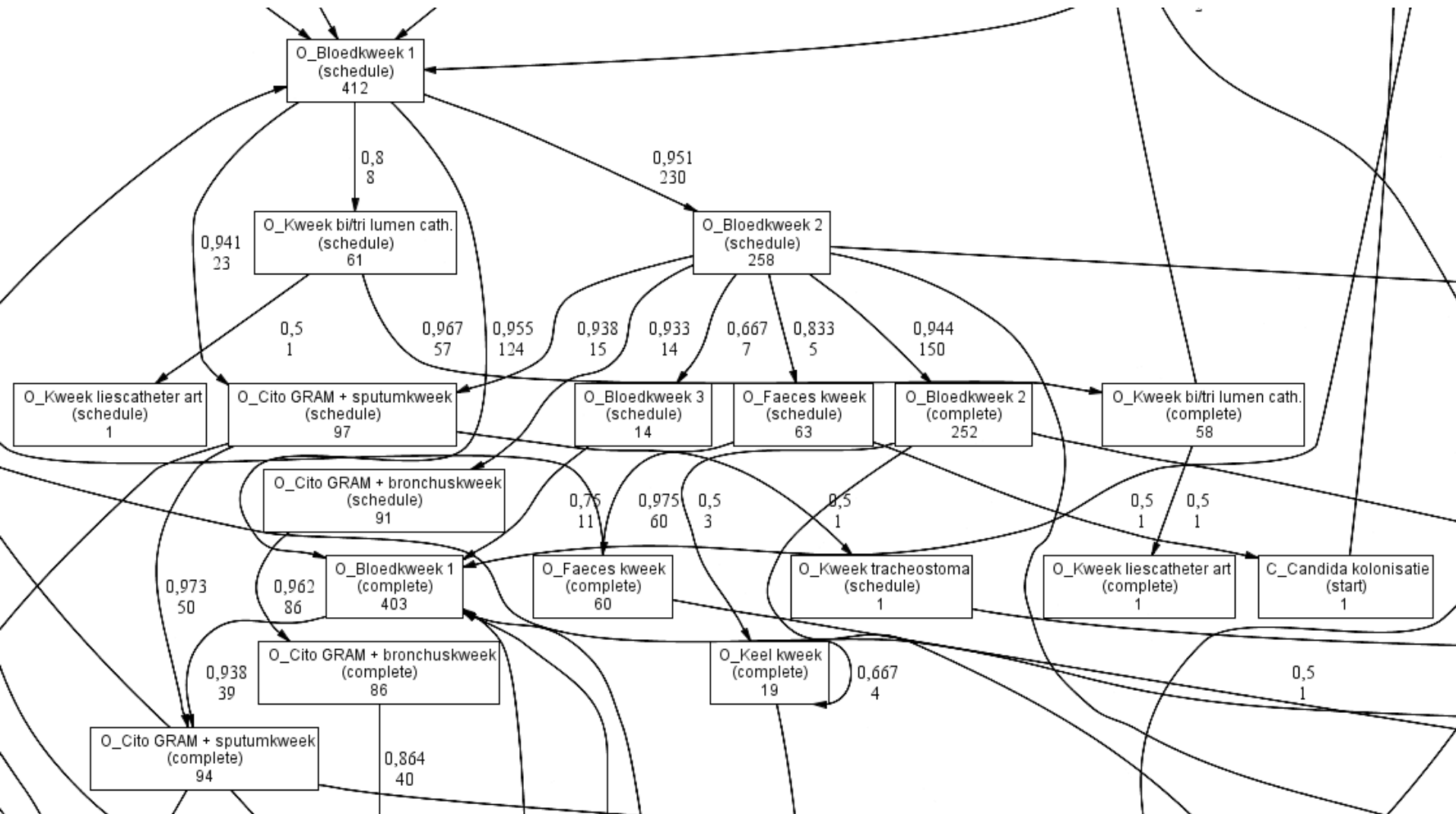
Example of a Spaghetti process



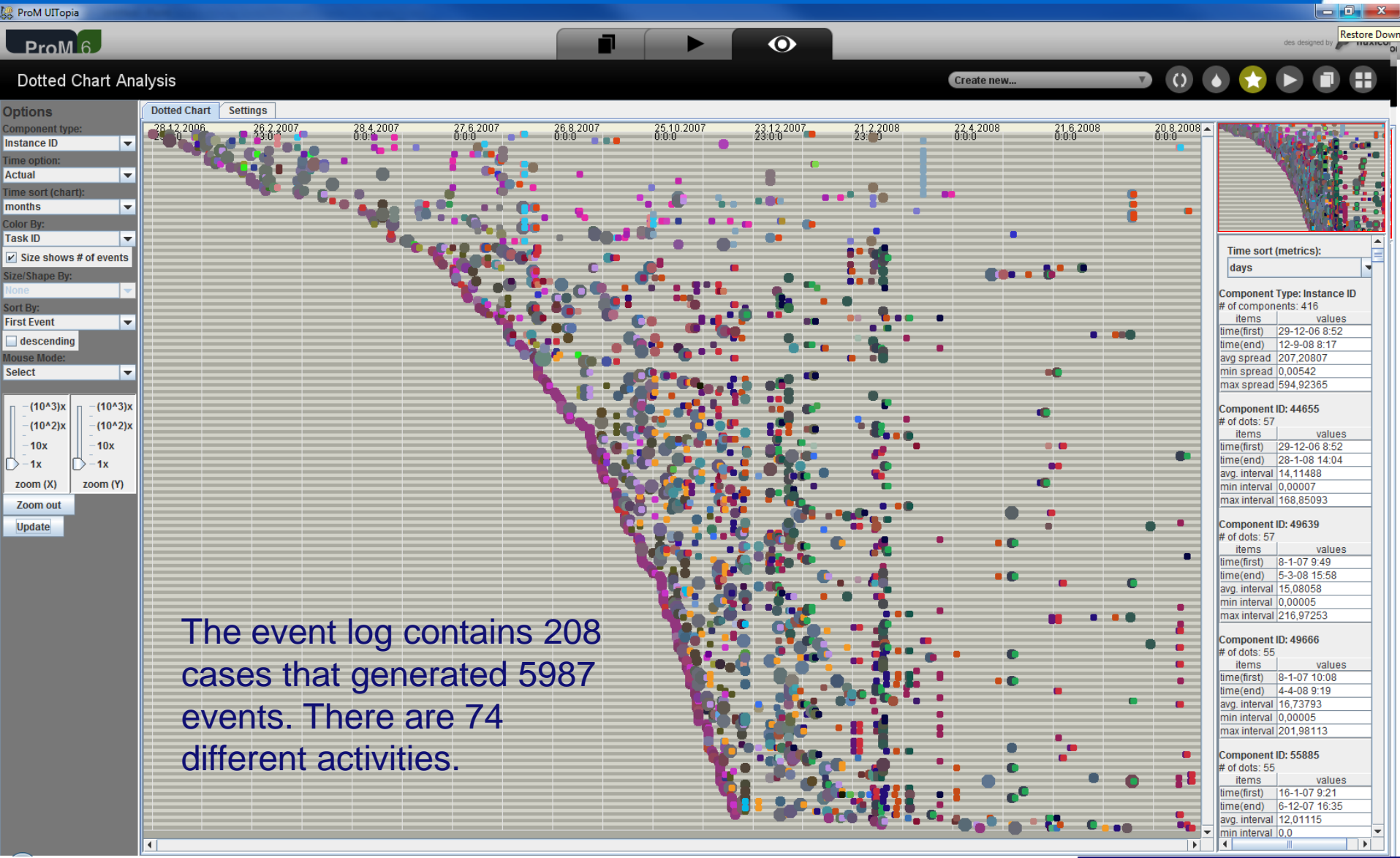
Spaghetti process describing the diagnosis and treatment of 2765 patients in a Dutch hospital. The process model was constructed based on an event log containing 114,592 events. There are 619 different activities (taking event types into account) executed by 266 different individuals (doctors, nurses, etc.).

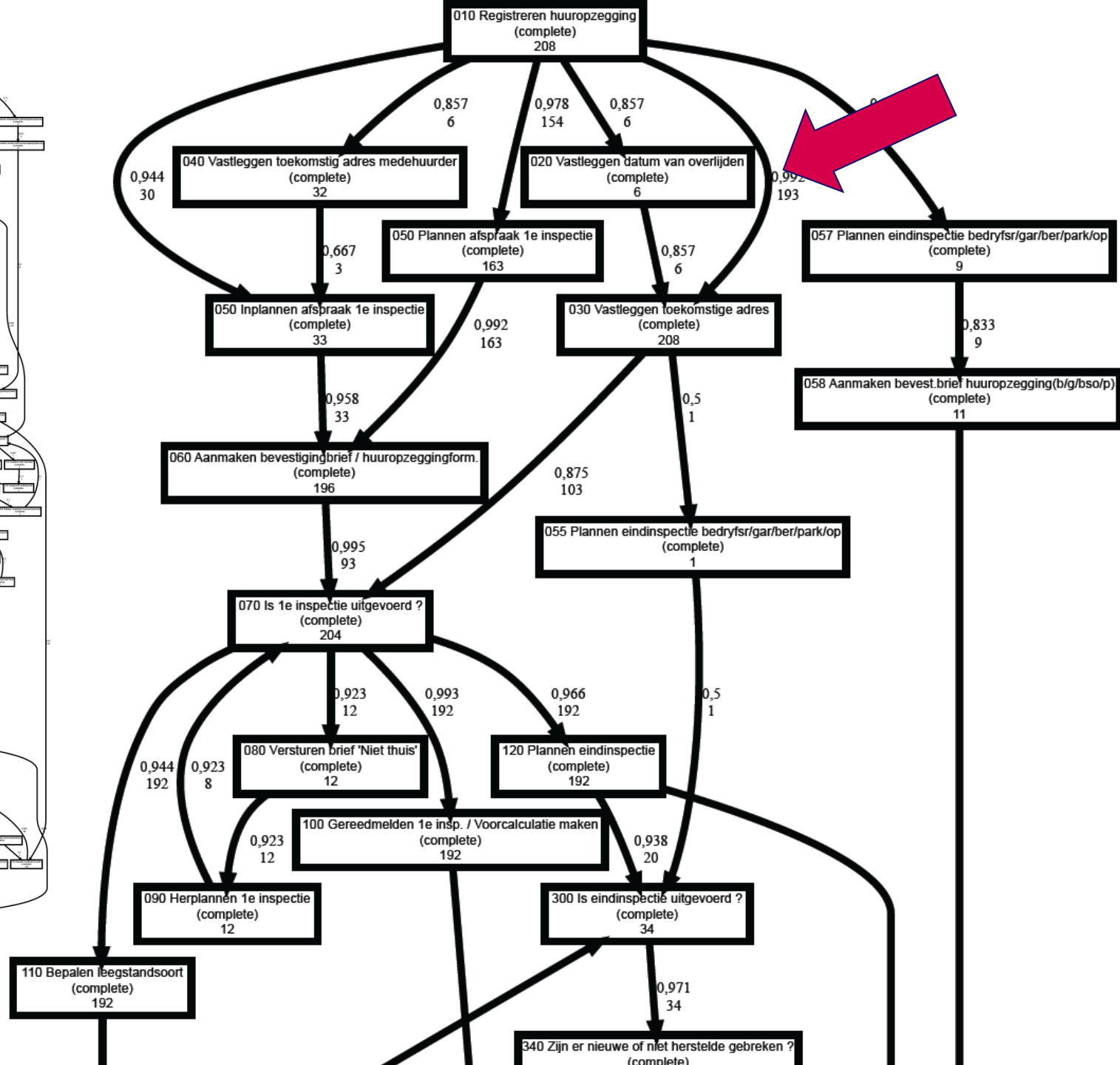
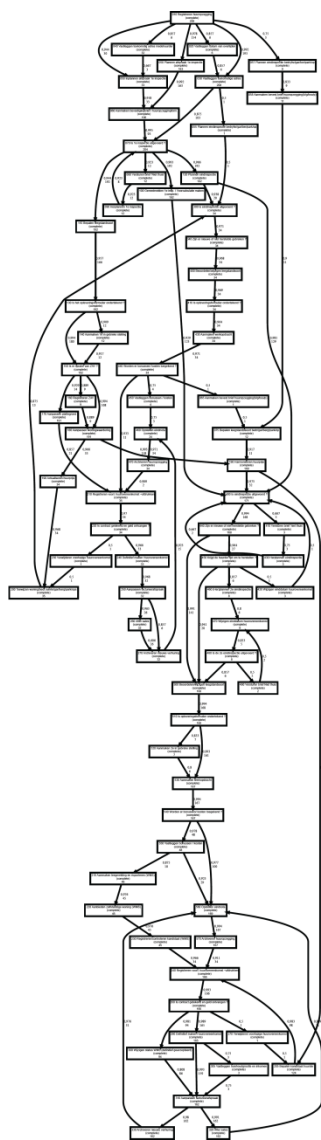
Fragment

18 activities of the 619 activities (2.9%)



Another example (event log of Dutch housing agency)





How can process mining help?

- Detect bottlenecks
- Detect deviations
- Performance measurement
- Suggest improvements
- Decision support (e.g., recommendation and prediction)

- Provide mirror
- Highlight important problems
- Avoid ICT failures
- Avoid management by PowerPoint
- From “politics” to “analytics”

After this lecture you should be able to:

- Provide an overview of process mining and ProM's functionality.
- Discover a Petri net based on a concrete event log using the α algorithm.
- Tell about the limitations of the α algorithm.
- Construct event logs (or targeted Petri nets) for which the α algorithm produces an incorrect result.
- Explain the delicate balance between overfitting and underfitting.