Process Mining

Making Sense of Processes Hidden in Big Event Data

EIS Colloquium, 7-12-2012, TU/e, Eindhoven

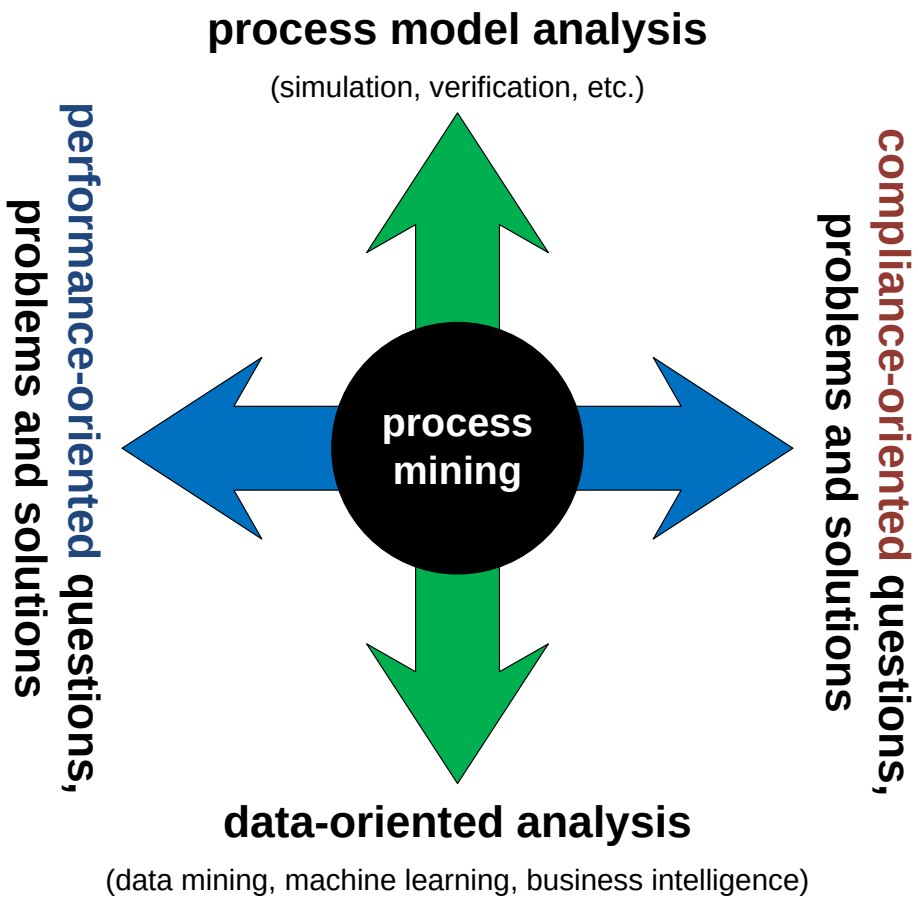
Wil van der Aalst

www.processmining.org

Technische Universiteit
Eindhoven
University of Technology

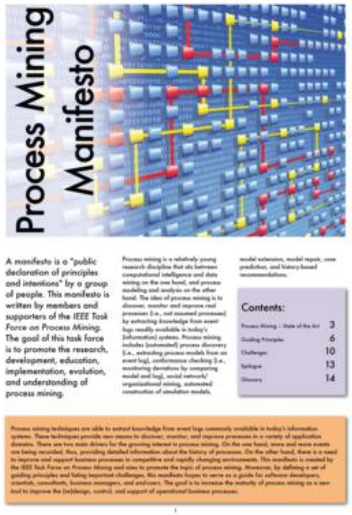
Where innovation starts

Positioning Process Mining



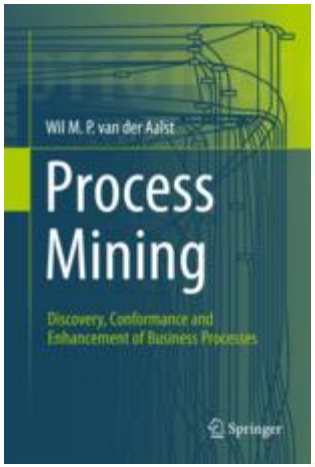
Advances in Process Mining

- **Many process discovery and conformance checking algorithms and tools are available (cf. the various ProM packages).**
- **Also commercial software based on these ideas:**
  Disco (Fluxicon), Reflect (Futura/Percentive), BPMOne (Pallas Athena/Perceptive), ARIS Process Performance Manager (Software AG), Futura Reflect (Futura Technology), Interstage Automated Process Discovery (Fujitsu), QPR ProcessAnalyzer/Analysis (QPR Software), flow (fourspark), Discovery Analyst (StereoLOGIC), etc.
- **We applied process mining in over 100 organizations.**



More than 75 people involving more than 50 organizations created the Process Mining Manifesto in the context of the IEEE Task Force on Process Mining.

Available in 13 languages



What Happens in an **Internet Minute?**

And **Future Growth** is **Staggering**

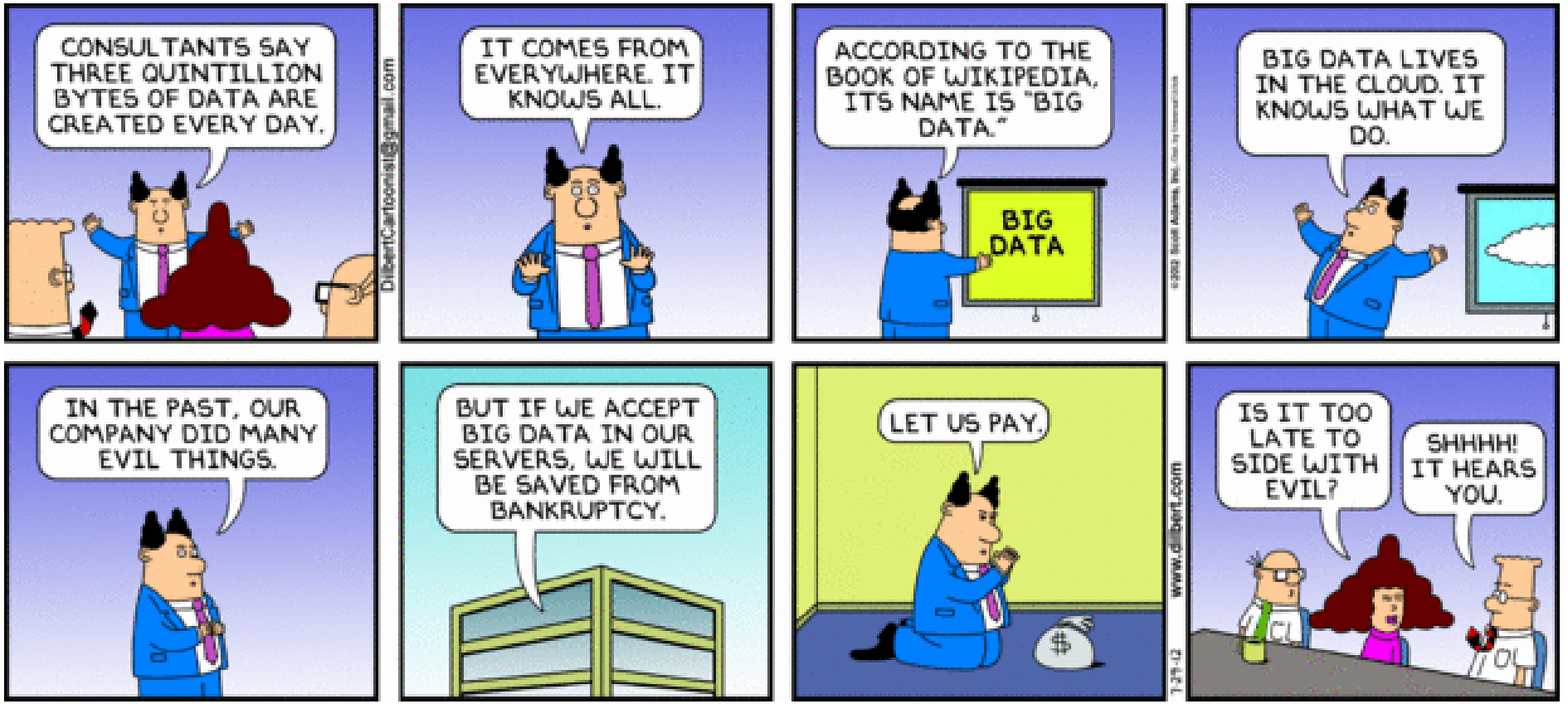
Today, the number of **networked devices** = the global population

By **2015**, the number of **networked devices** = **2x** the global population

In **2015**, it would take you **5 years**

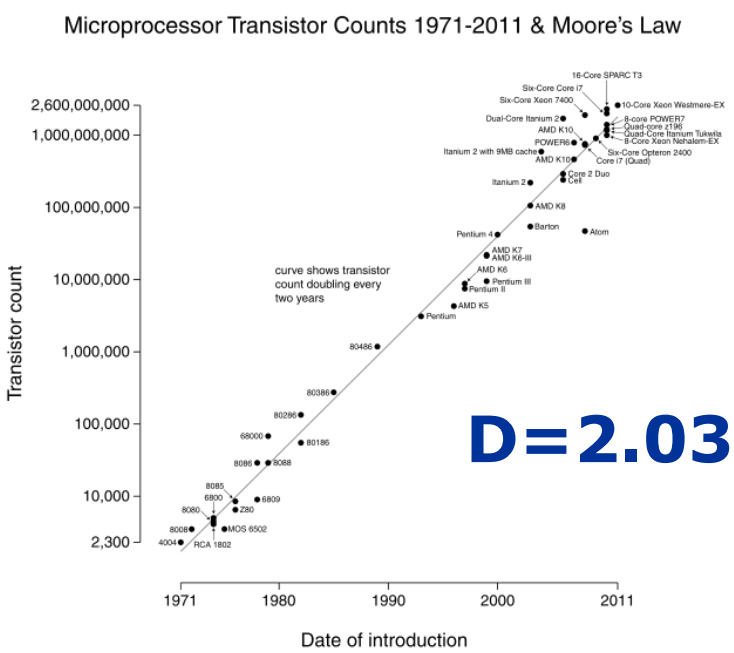
to view all video crossing IP networks each **second**

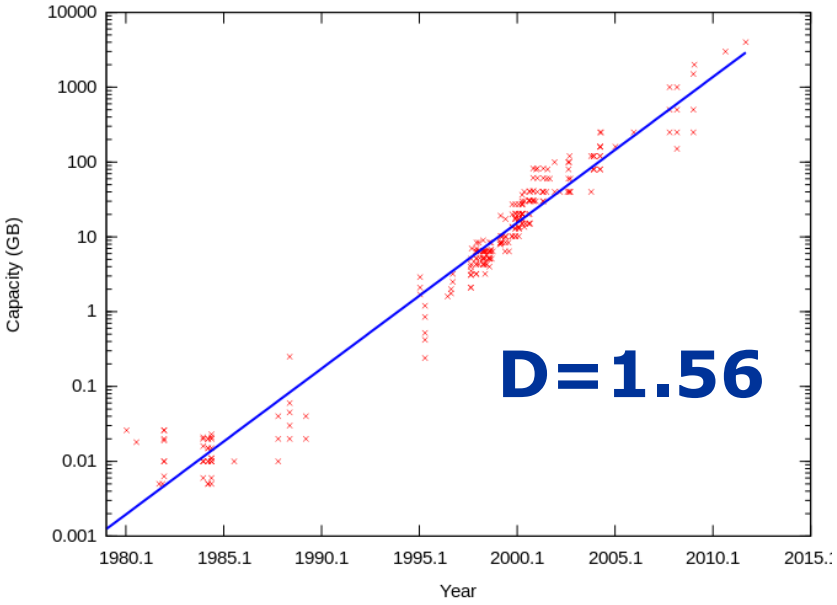
Big Data: Even Dilbert and the "pointy-haired boss" know about it …

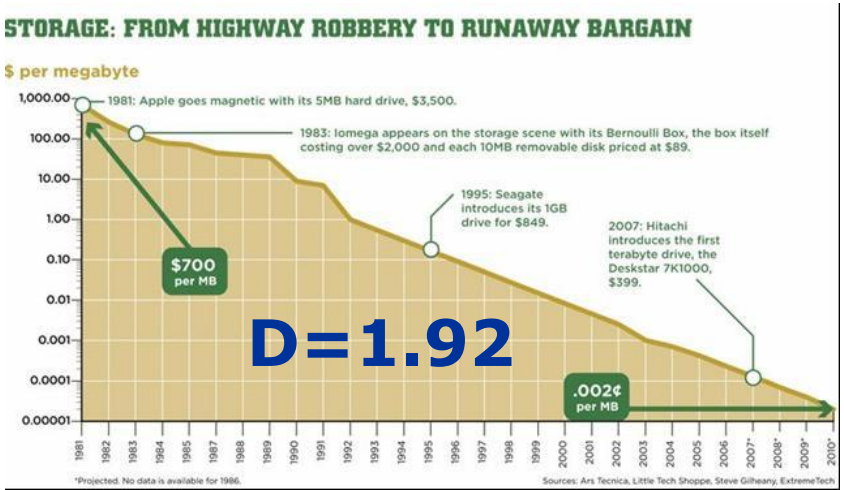


http://dilbert.com/strips/comic/2012-07-29/

Moore's Law


Microprocessor Transistor Counts 1971-2011 & Moore's Law
Transistor count
2,600,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
2,300
curve shows transistor count doubling every two years
16-Core SPARC T3
Six-Core Core i7
Six-Core Xeon 7400
10-Core Xeon Westmere-EX
Dual-Core Itanium 2
8-core POWER7
AMD K10
Quad-core z196
Quad-Core Itanium Tukwila
POWER6
8-Core Xeon Nehalem-EX
Itanium 2 with 9MB cache
Six-Core Opteron 2400
AMD K10
Core i7 (Quad)
Core 2 Duo
Cell
Itanium 2
AMD K8
Pentium 4
Barton
Atom
AMD K7
AMD K6-III
AMD K6
Pentium III
Pentium II
AMD K5
Pentium
80486
80386
80286
68000
80186
8086
8088
8085
6800
8809
8080
Z80
8008
MOS 6502
4004
RCA 1802
1971
1980
1990
2000
2011
Date of introduction
D=2.03


Capacity (GB)
10000
1000
100
10
1
0.1
0.01
0.001
1980.1
1985.1
1990.1
1995.1
2000.1
2005.1
2010.1
2015.
Year
D=1.56


STORAGE: FROM HIGHWAY ROBBERY TO RUNAWAY BARGAIN
$ per megabyte
1,000.00
100.00
10.00
1.00
0.10
0.01
0.001
0.0001
0.00001
1981: Apple goes magnetic with its 5MB hard drive, $3,500.
1983: Iomega appears on the storage scene with its Bernoulli Box, the box itself costing over $2,000 and each 10MB removable disk priced at $89.
1995: Seagate introduces its 1GB drive for $849.
2007: Hitachi introduces the first terabyte drive, the Deskstar 7K1000, $399.
$700 per MB
.002¢ per MB
D=1.92
1981
1982
1983
1984
1985
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007*
2008*
2009*
2010*
*Projected. No data is available for 1986.
Sources: Ars Tecnica, Little Tech Shoppe, Steve Gilheany, ExtremeTech

A simple calculation

- **Starting point 2010:**
  - **Harddisk 1 Terabyte = 10^{12} bytes**
  - **Digital Universe 1.2 Zettabyte = $1.2*10^{21}$ bytes** (estimate in IDC's annual report, "The Digital Universe Decade – Are You Ready?" May 2010)
- **Disk needs to grow $2^{30.16} = 1.2* 10^{9} = 1.2*10^{21}/ 10^{12}$ times its current size.**
- **Assuming D=1.56 this takes 30.16*1.56 = 47.05 years.**
- **Hence, in 2060 your laptop can contain all of today's digital universe (internet, computer files, transaction logs, movies, photos, music, books, databases, etc.)!**

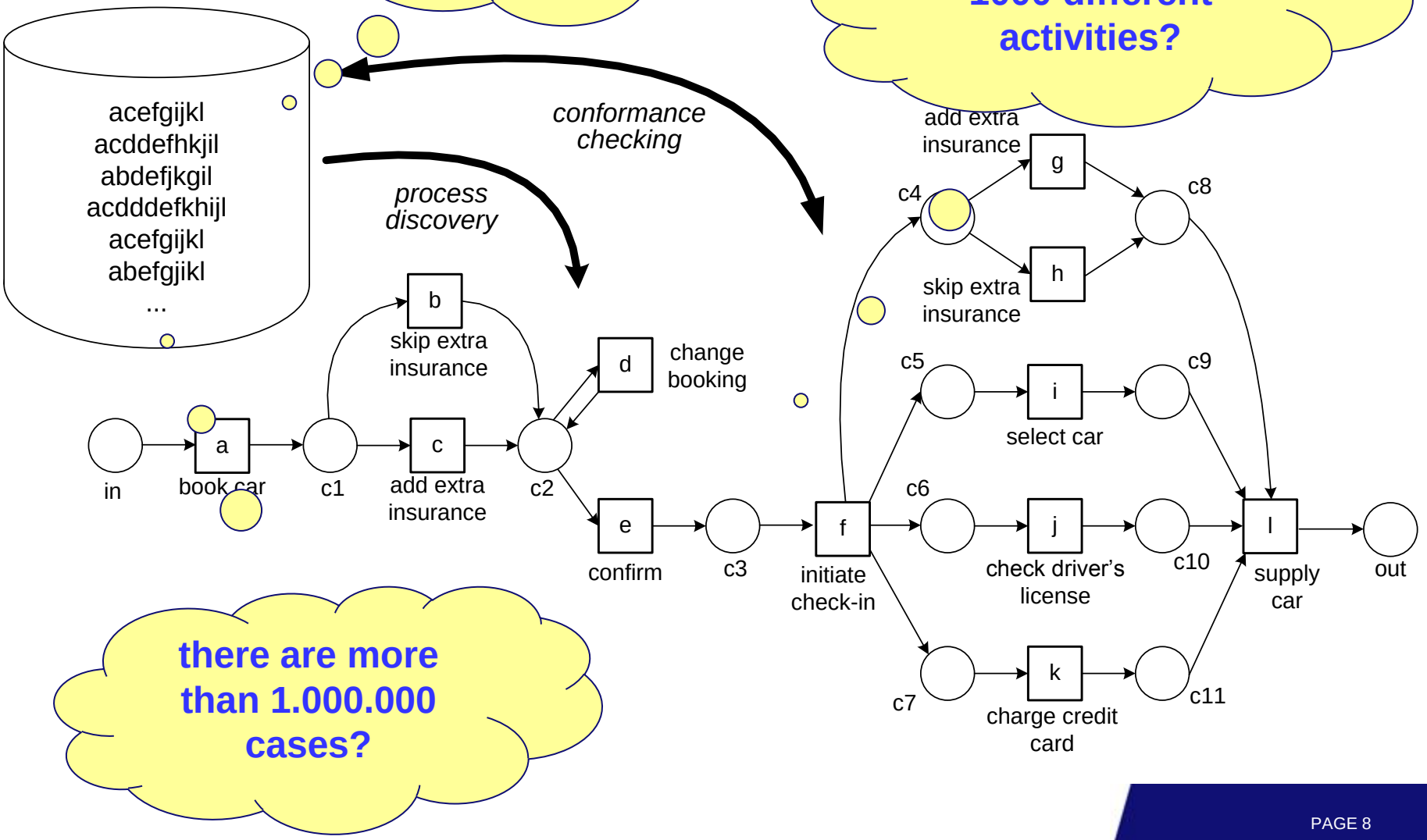
Big Data: Opportunities and Challenges

What if?

there are more than 100.000.000 events?

there are more than 1000 different activities?

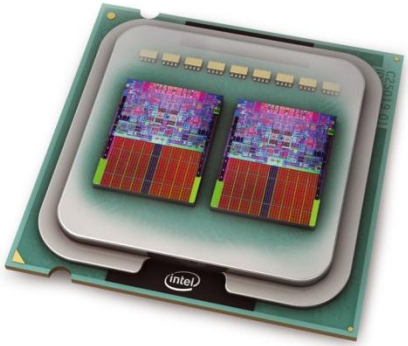
there are more than 1.000.000 cases?

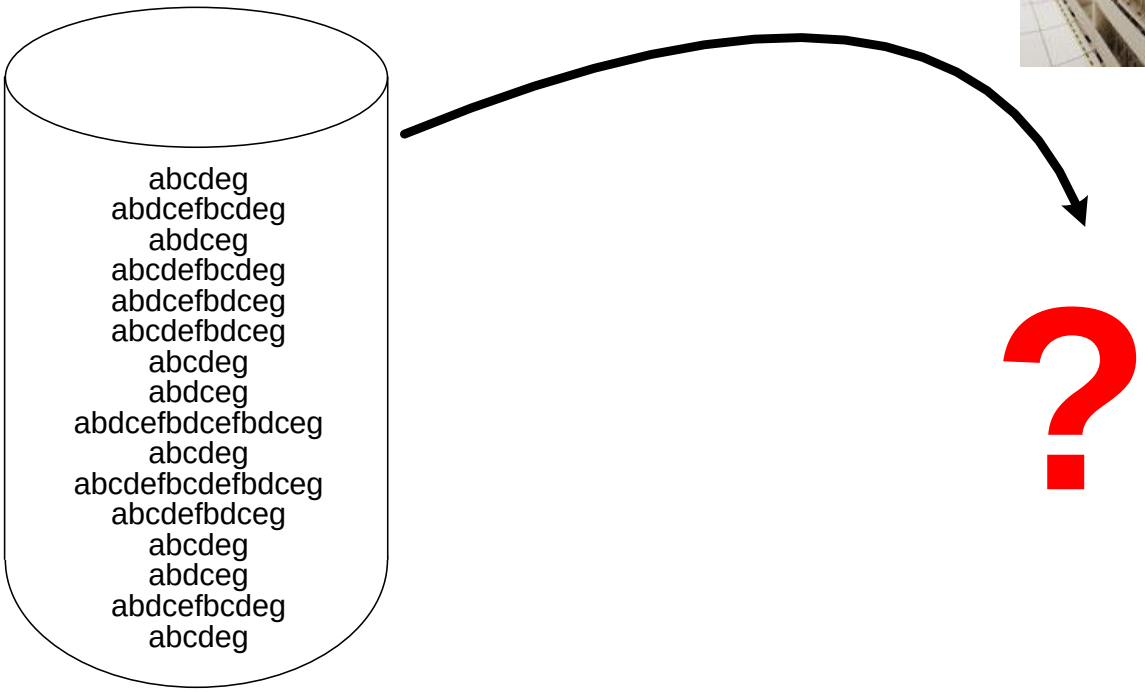


Decomposing/Distributing Process Mining Problems

This talk is not about a new algorithm!

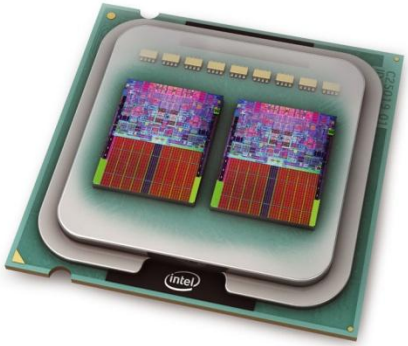
How to decompose/distribute process discovery?

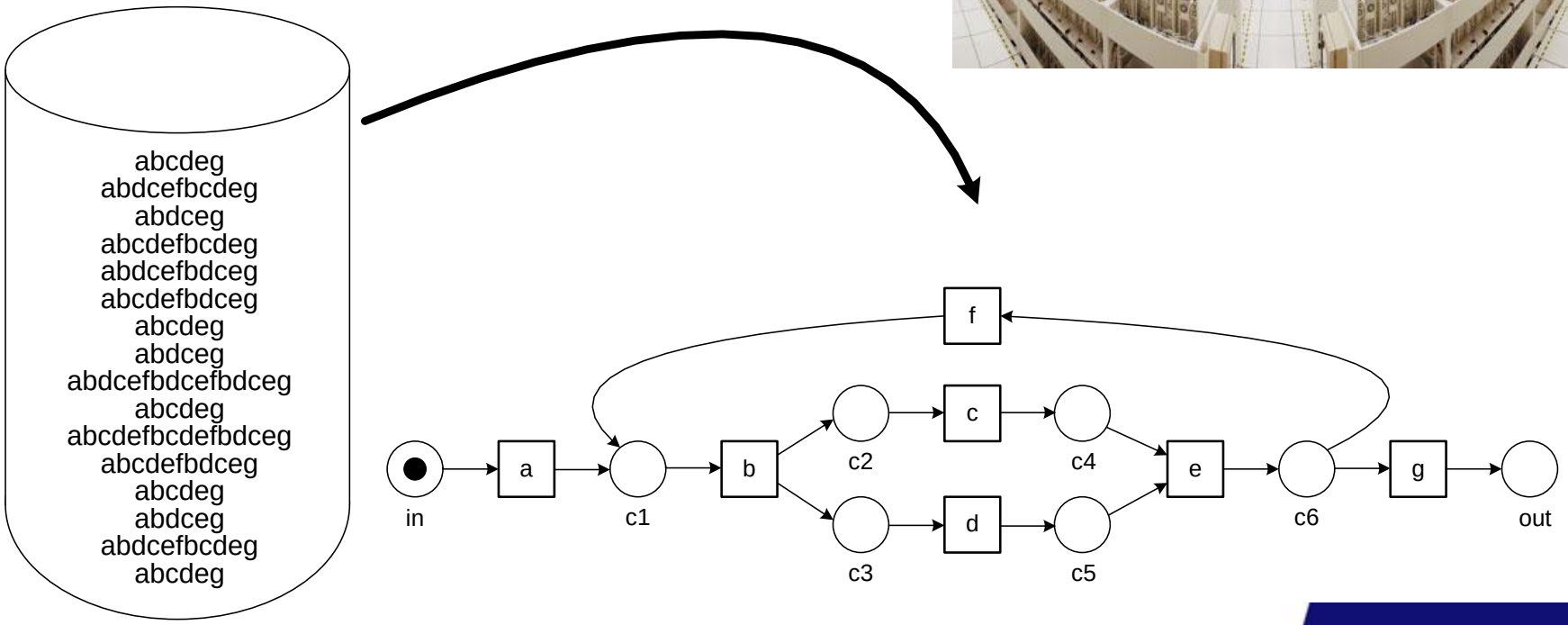




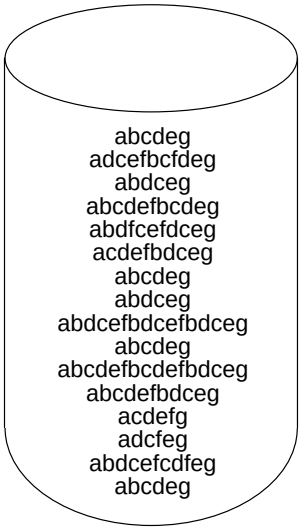
?

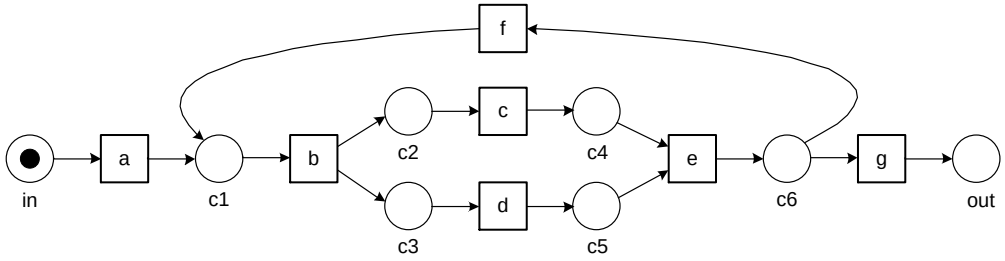
How to decompose/distribute process discovery?

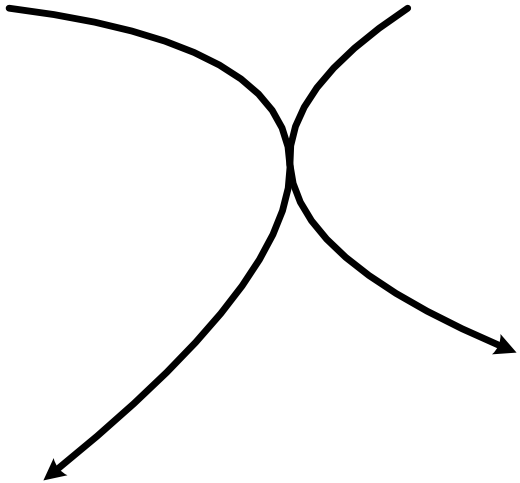




How to decompose/distribute conformance checking?

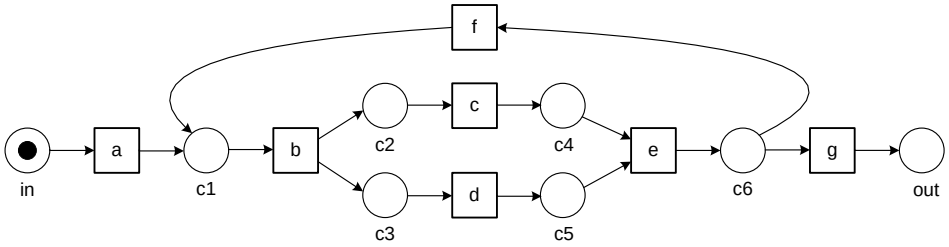


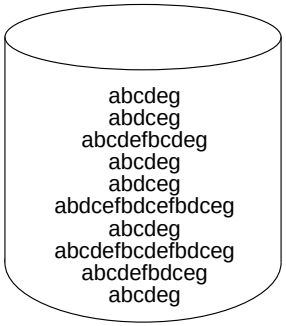


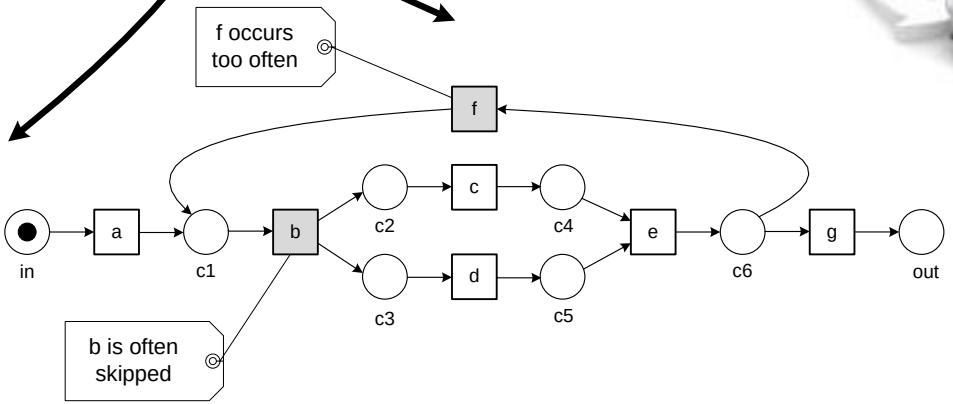


?

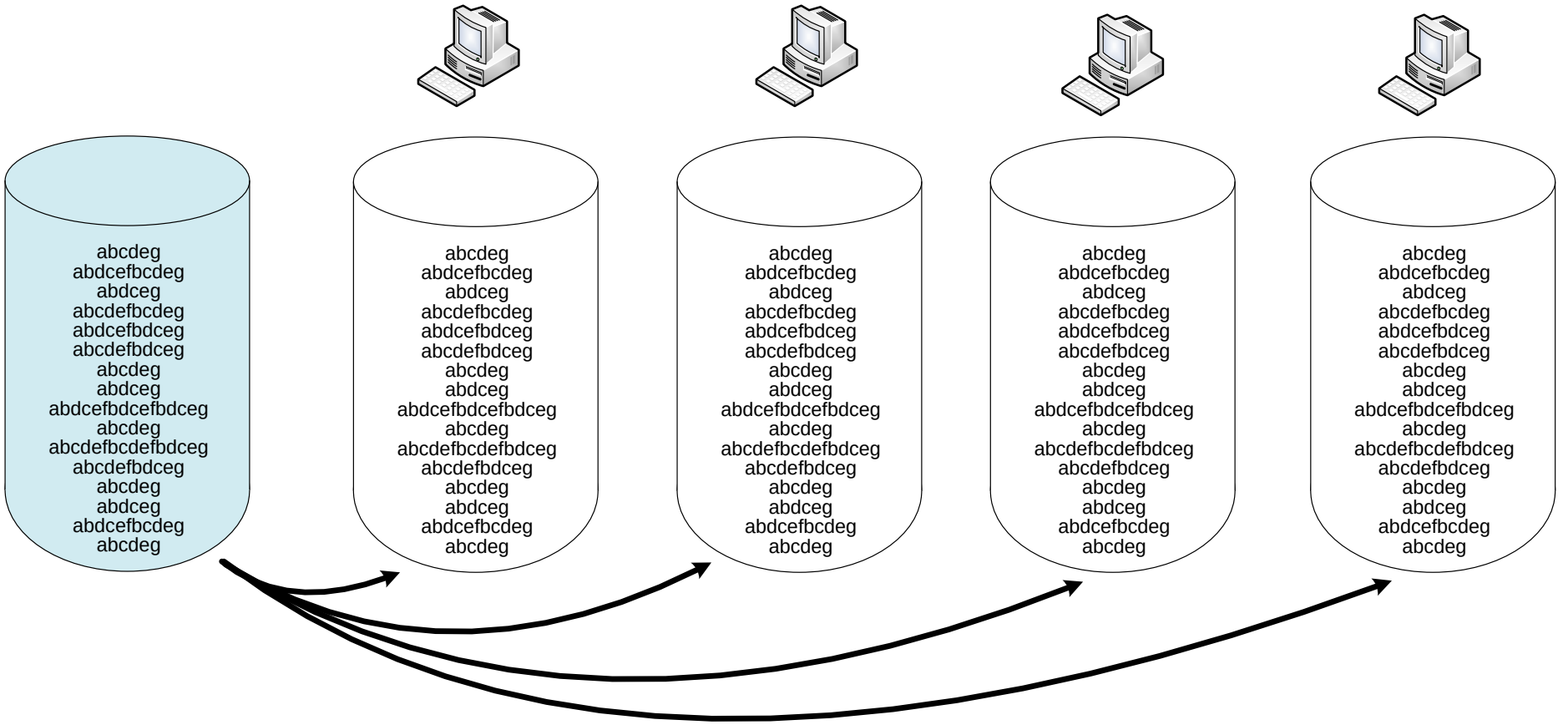
How to decompose/distribute conformance checking?



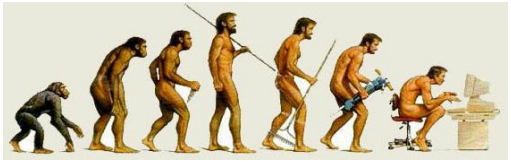




Replication: Same event log on all computing nodes



Only makes sense if random elements, e.g., genetic process mining.



Classification based on partitioning of event log: vertical and horizontal

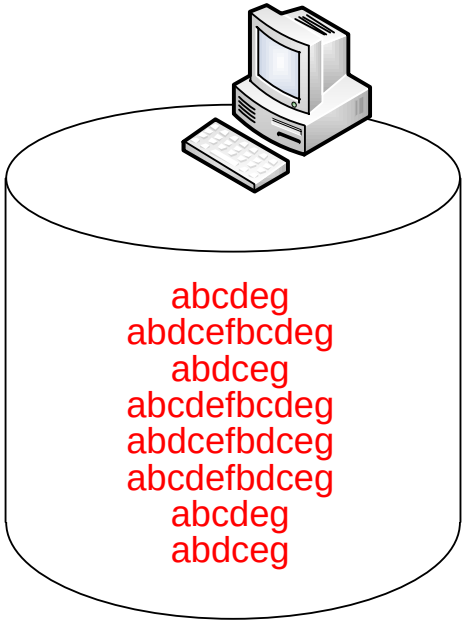
sets of cases

sets of activities

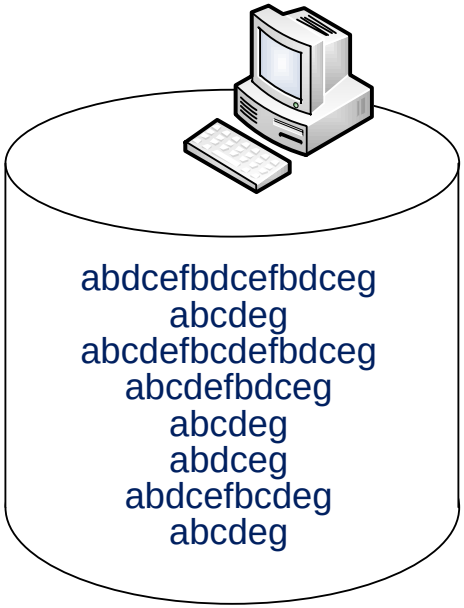
Vertical distribution I: Split cases arbitrarily

sets of cases

abcdeg
abdcefbcdeg
abdceg
abcdefbcdeg
abdcefbdceg
abcdefbdceg
abcdeg
abdceg
abdcefbdcefbdceg
abcdeg
abcdefbcdefbdceg
abcdefbdceg
abcdeg
abdceg
abdcefbcdeg
abcdeg

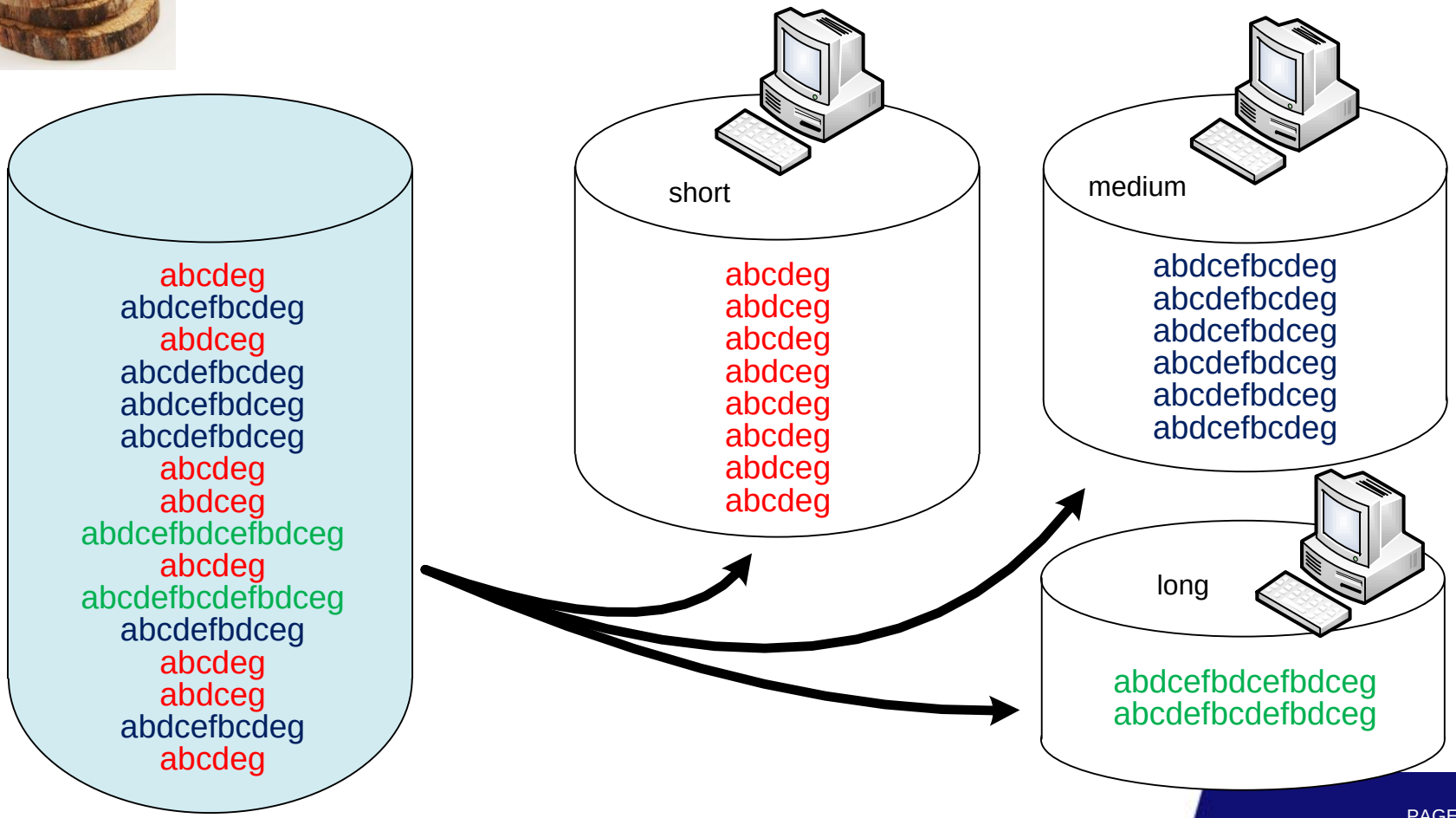


abcdeg
abdcefbcdeg
abdceg
abcdefbcdeg
abdcefbdceg
abcdefbdceg
abcdeg
abdceg



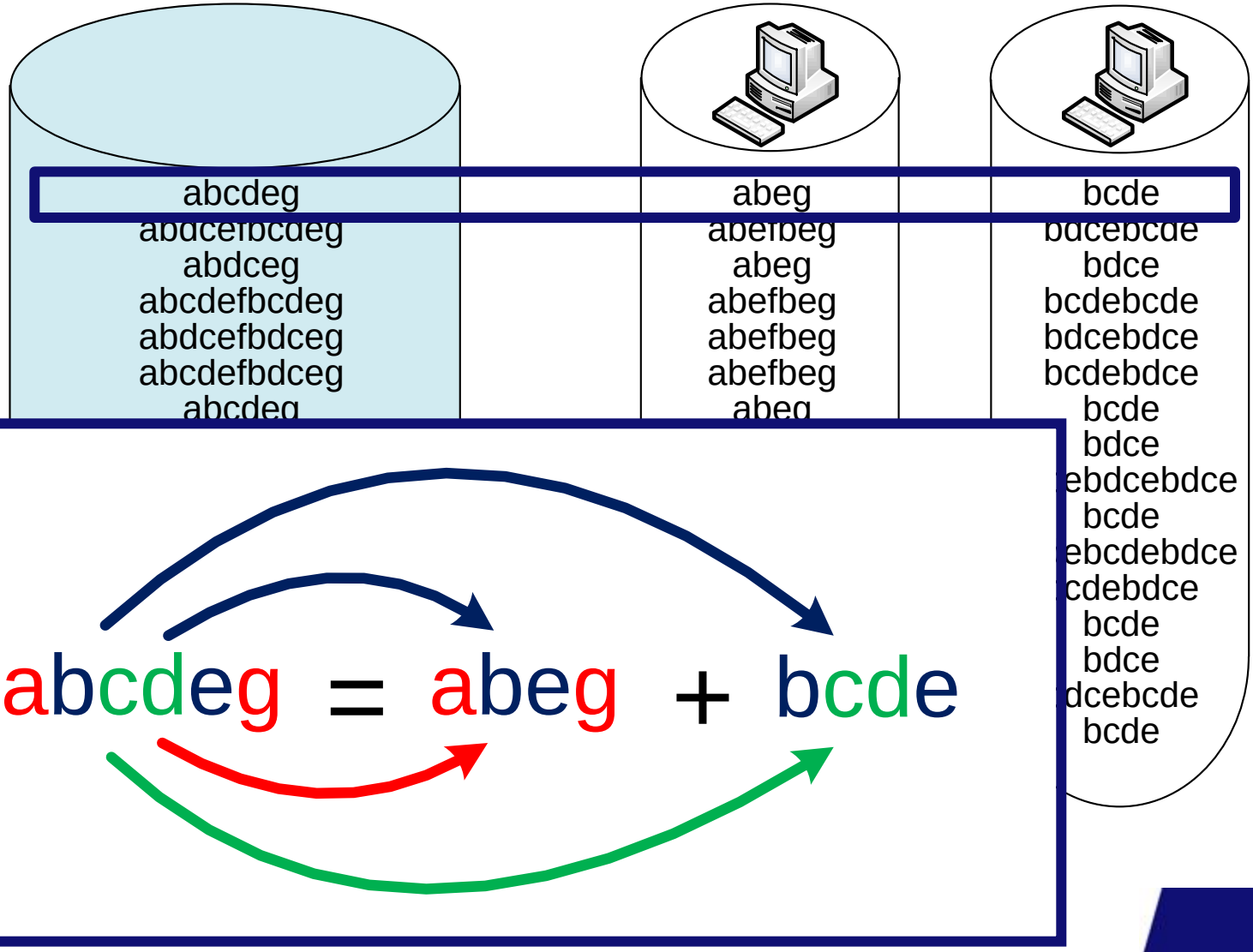
abdcefbdcefbdceg
abcdeg
abcdefbcdefbdceg
abcdefbdceg
abcdeg
abdceg
abdcefbcdeg
abcdeg

Vertical distribution II: Split cases based on a specific feature



Horizontal distribution

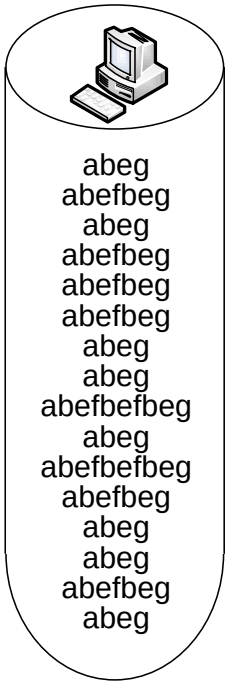
sets of activities

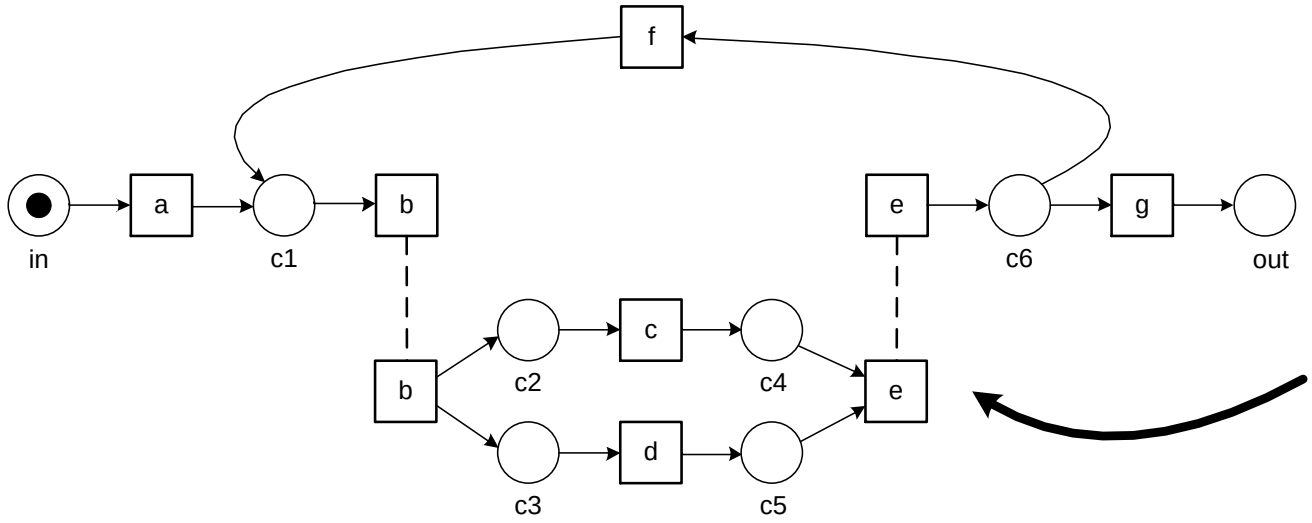


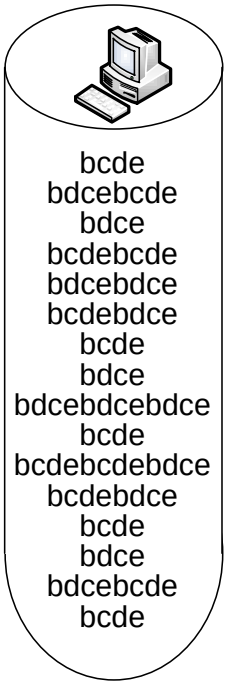
Horizontal distribution: The key idea

projected on a,b,e,f,g

projected on b,c,d,e

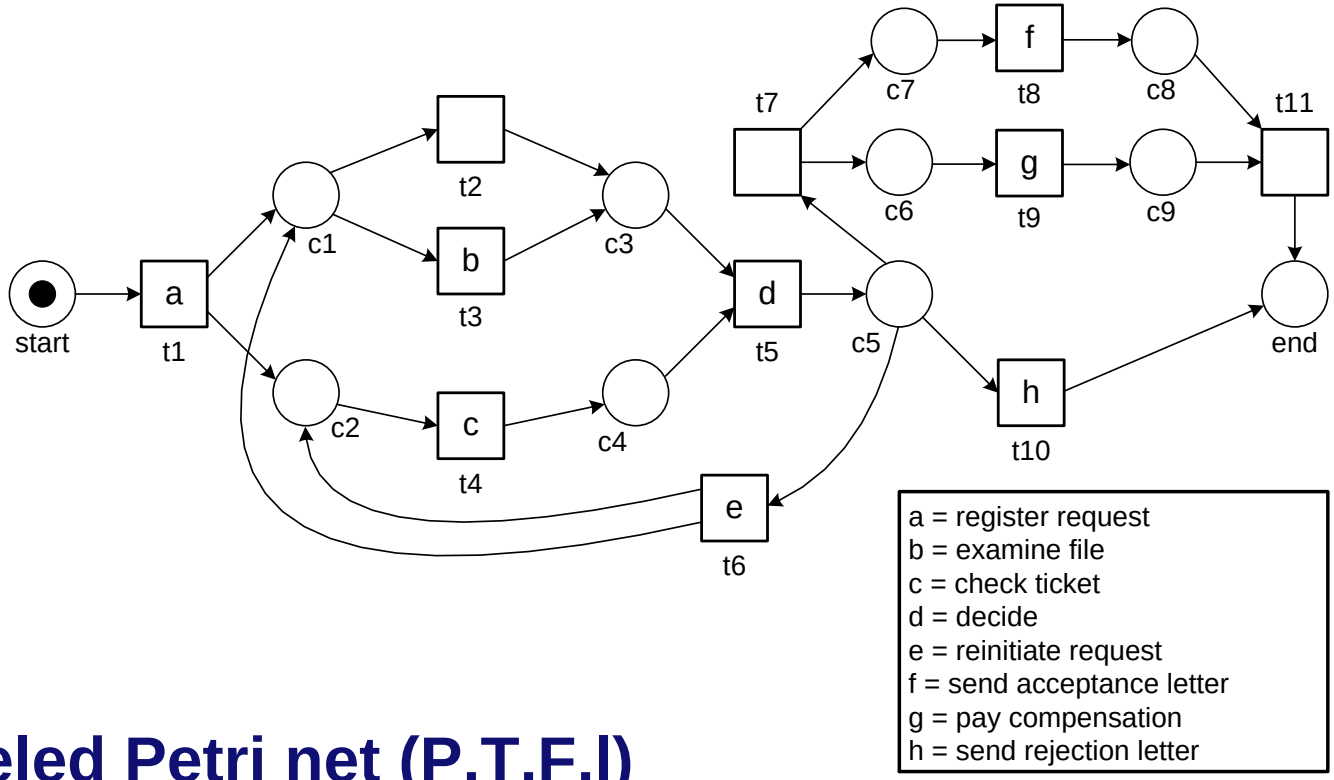






General Case

System Net

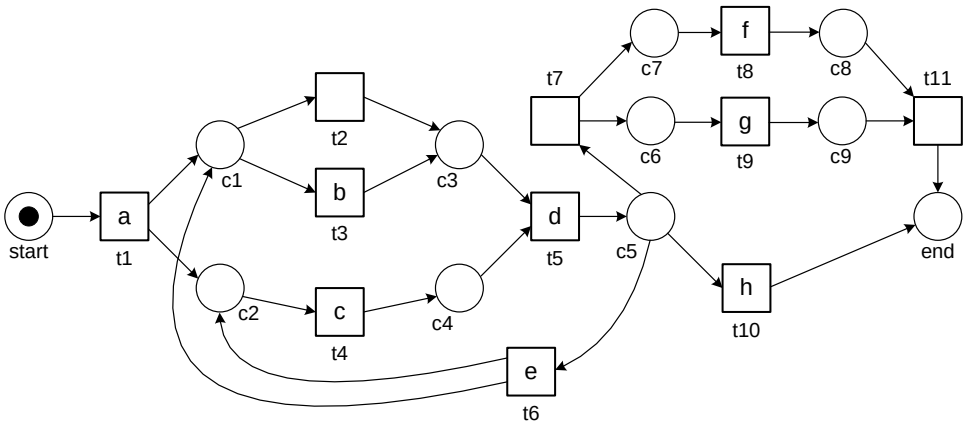


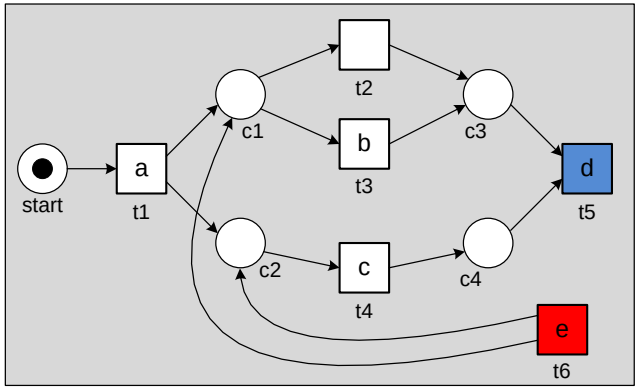
- **Labeled Petri net (P,T,F,l)**
- **Silent transitions and visible transitions (unique or not),**
- **One initial marking M_{init}, one final marking M_{final}**

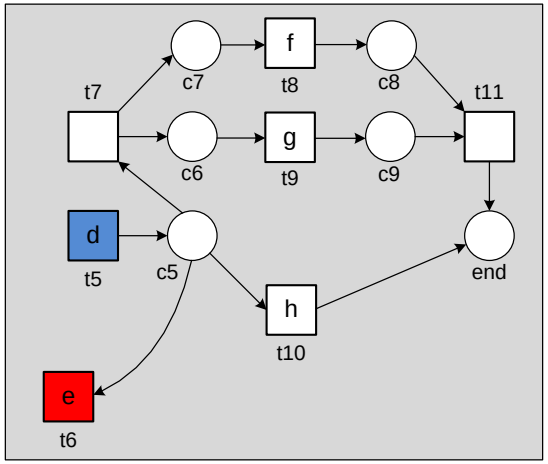
Traces and Logs

- A system net SN has a set of possible visible traces $\phi(SN)$ starting in M_{init} and ending M_{final} only showing the visible steps.
- An event log L is a multiset of traces.
- Two main process mining problems:
  1. Conformance checking: Given L and SN, evaluate the "conformance" (e.g., fitness, precision, generalization, etc.) of L and $\phi(SN)$
  2. Process discovery: Given L, create SN such that the conformance of L and $\phi(SN)$ is "as good as possible"

Valid Decomposition

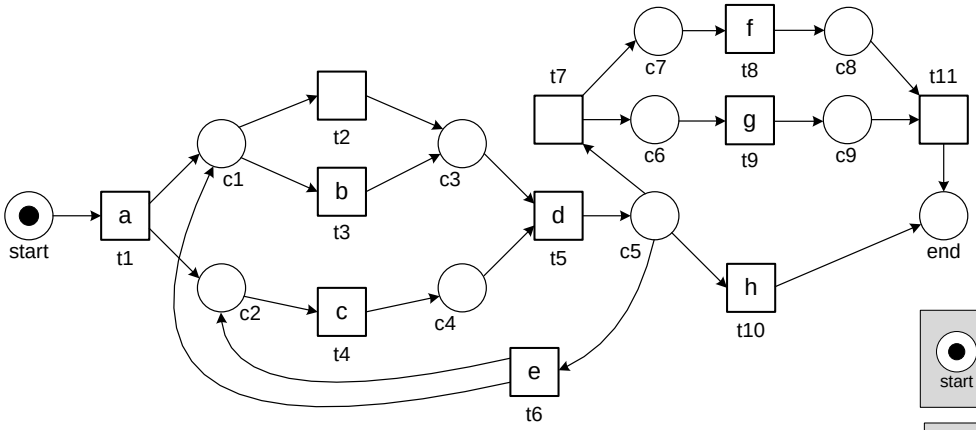


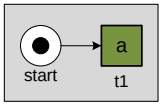


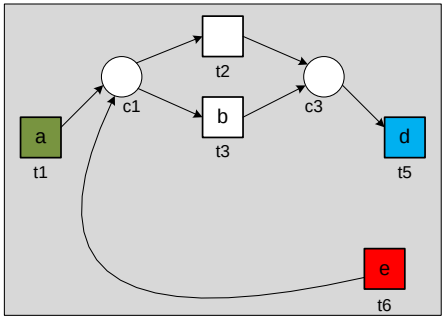


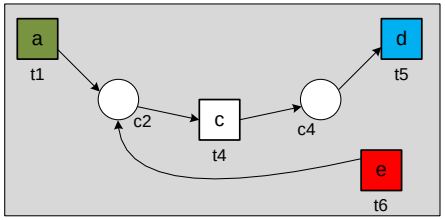
- **Union of subnets is original net**
- **No shared places**
- **Shared transitions are visible and have unique label**

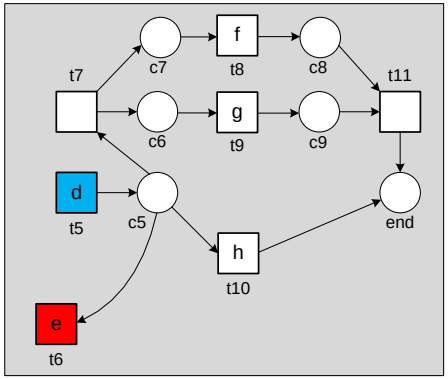
Another Decomposition







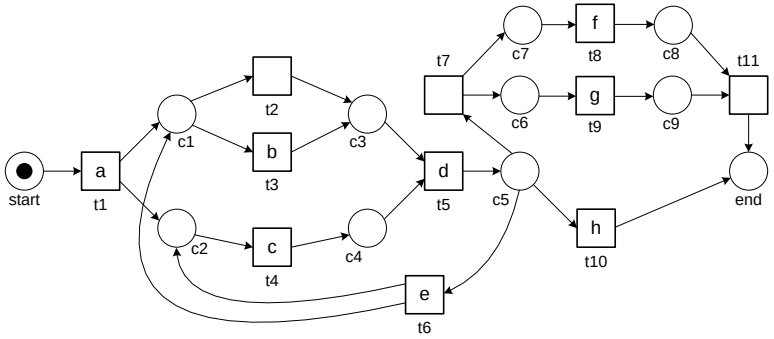


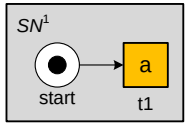


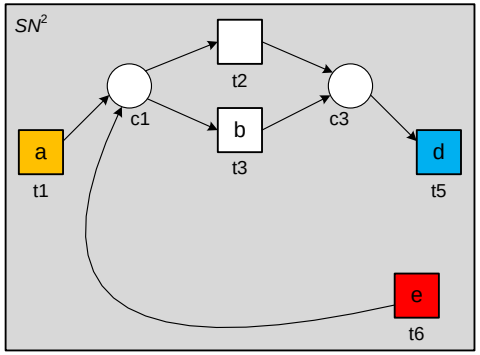
Requirement

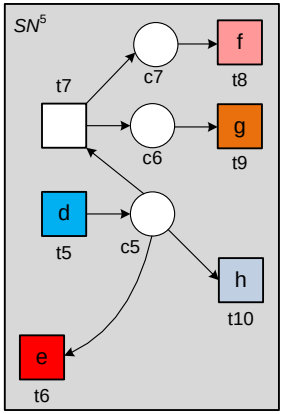
- **Union of subnets is original net**
- **No shared places**
- **Shared transitions are visible and unique**

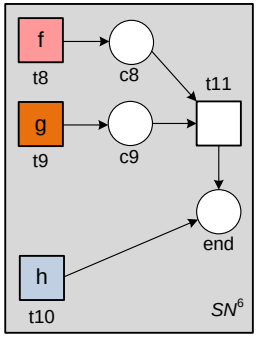
Maximal Valid Decomposition

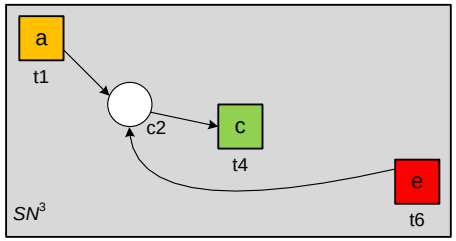


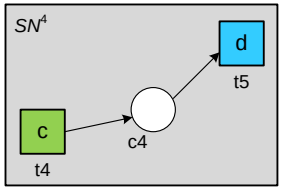




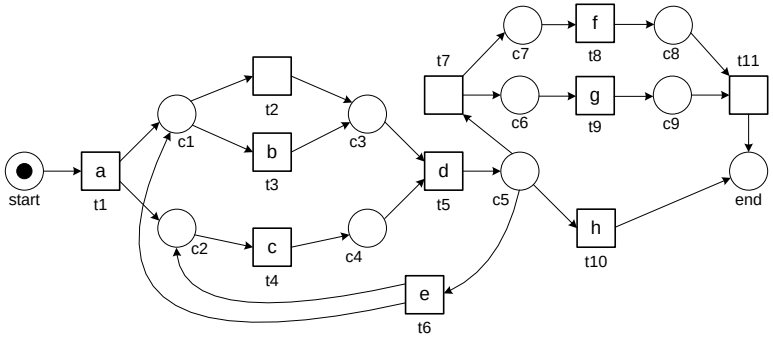


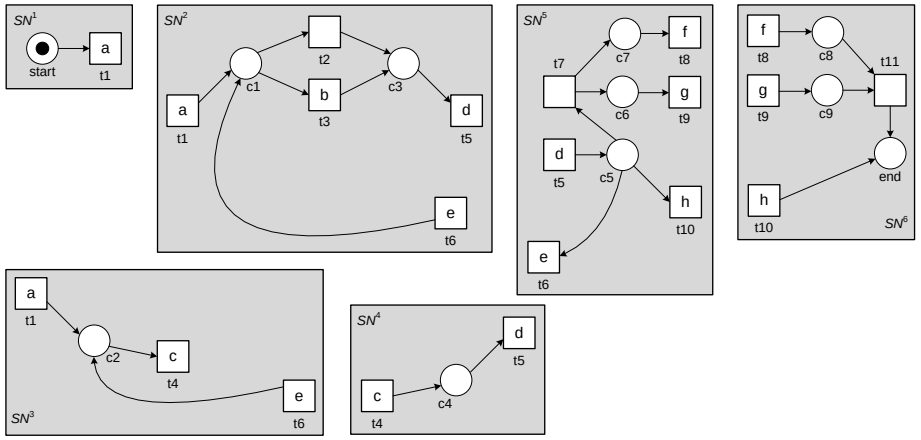


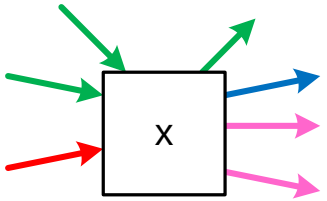




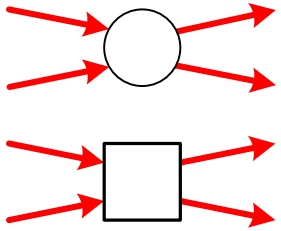
Maximal Decomposition

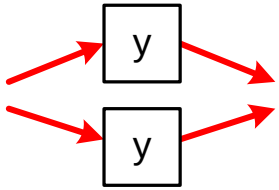




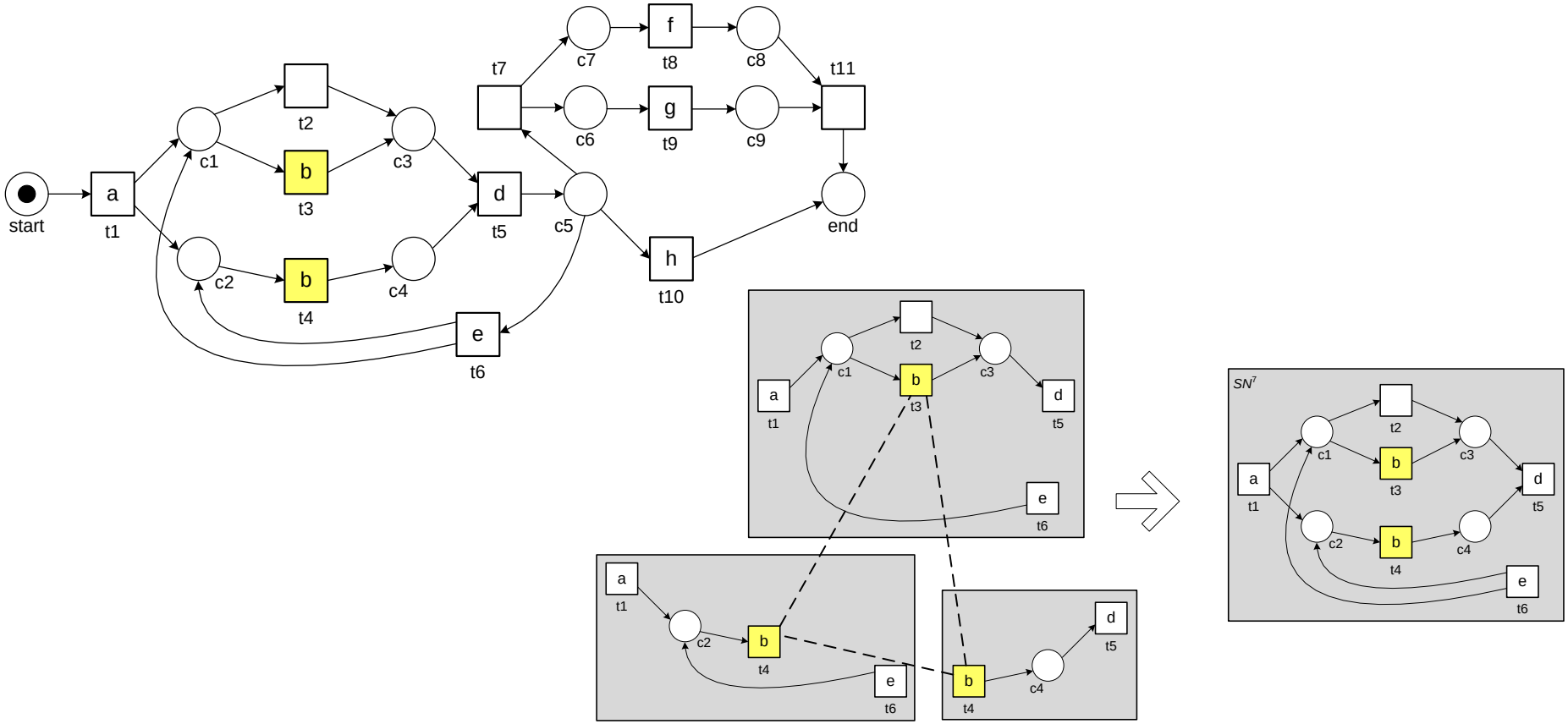


- **Construction: group arcs iteratively**
- **Maximal decomposition is unique and always a valid decomposition**





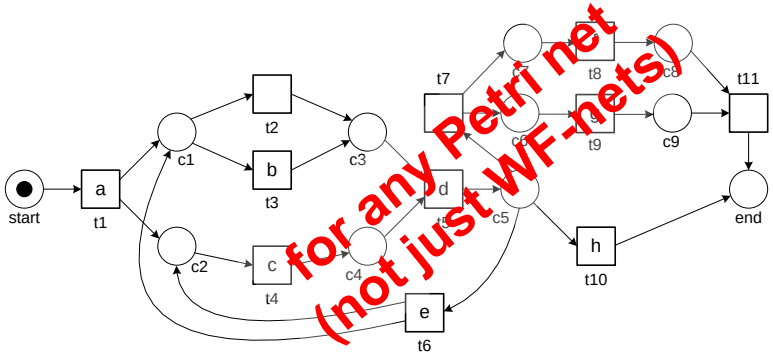
Non-unique visible labels

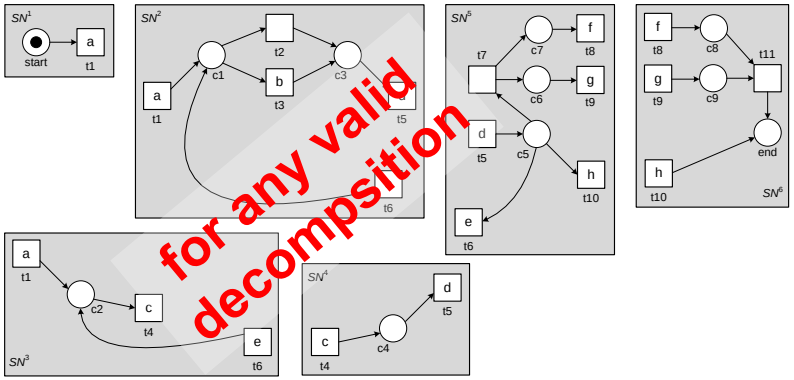


- **Union of subnets is original net**
- **No shared places**
- **Shared transitions are visible and unique**

Conformance checking can be decomposed !!!

- Let L be an event log, SN a system net, and $D=\{SN^1, SN^2, \ldots SN^n\}$ a valid decomposition
- L^i is the sublog of SN^i (L projected onto visible transitions of SN^i)

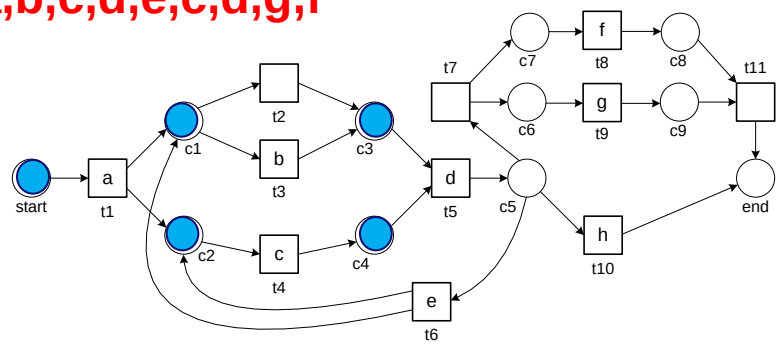


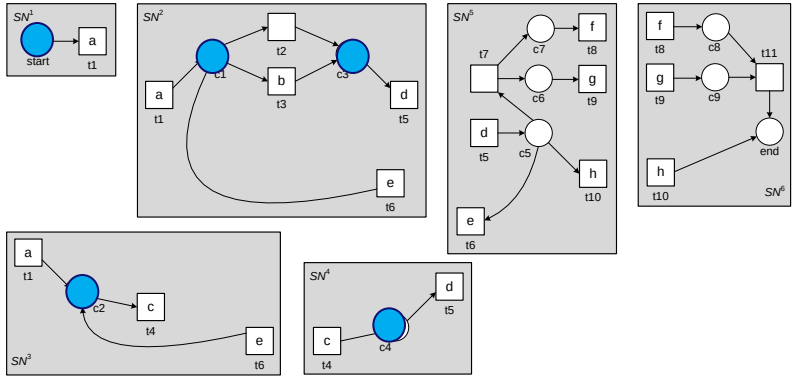


L is perfectly fitting SN
if and only if
each projected log is L^i is perfectly fitting SN^i

Example of alignment for observed trace a,b,c,d,e,c,d,g,f

a,b,c,d,e,c,d,g,f





$$\gamma_3 = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ \hline a & b & c & d & e & c & \gg & d & \gg & g & f & \gg \\ \hline a & b & c & d & e & c & \tau & d & \tau & g & f & \tau \\ t1 & t3 & t4 & t5 & t6 & t4 & t2 & t5 & t7 & t9 & t8 & t11 \\ \hline \end{array}$$

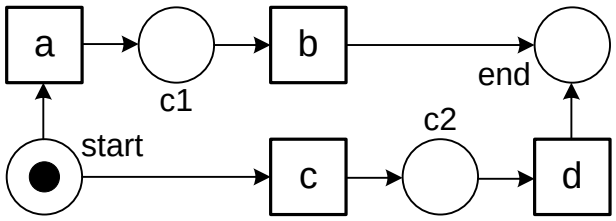
Etc.

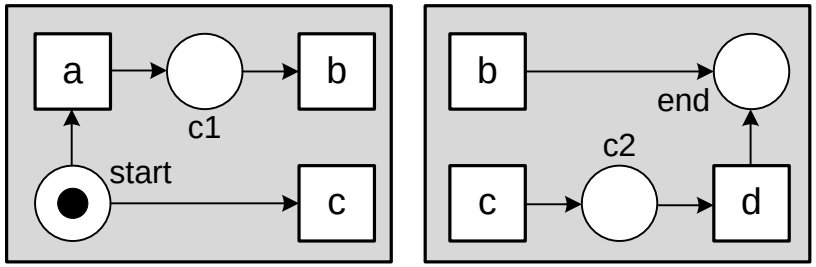
$$\gamma_3^1 = \begin{array}{|c|} \hline 1 \\ \hline a \\ \hline a \\ t1 \\ \hline \end{array} \qquad \gamma_3^2 = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 2 & 4 & 5 & 7 & 8 \\ \hline a & b & d & e & \gg & d \\ \hline a & b & d & e & \tau & d \\ t1 & t3 & t5 & t6 & t2 & t5 \\ \hline \end{array} \qquad \gamma_3^3 = \begin{array}{|c|c|c|c|} \hline 1 & 3 & 5 & 6 \\ \hline a & c & e & c \\ \hline a & c & e & c \\ t1 & t4 & t6 & t4 \\ \hline \end{array}$$

$$\gamma_3^4 = \begin{array}{|c|c|c|c|} \hline 3 & 4 & 6 & 8 \\ \hline c & d & c & d \\ \hline c & d & c & d \\ t4 & t5 & t4 & t5 \\ \hline \end{array} \qquad \gamma_3^5 = \begin{array}{|c|c|c|c|c|c|} \hline 4 & 5 & 8 & 9 & 10 & 11 \\ \hline d & e & d & \gg & g & f \\ \hline d & e & d & \tau & g & f \\ t5 & t6 & t5 & t7 & t9 & t8 \\ \hline \end{array} \qquad \gamma_3^6 = \begin{array}{|c|c|c|} \hline 10 & 11 & 12 \\ \hline g & f & \gg \\ \hline g & f & \tau \\ t9 & t8 & t11 \\ \hline \end{array}$$

Quantifying Conformance

- **Fraction of cases perfectly fitting SN equals the fraction of cases for which each projected log is L^i is perfectly fitting SN^i**
- **For fitness at the event level (costs based alignment) it is possible to compute an optimistic value.**





a,b  **c,d** 

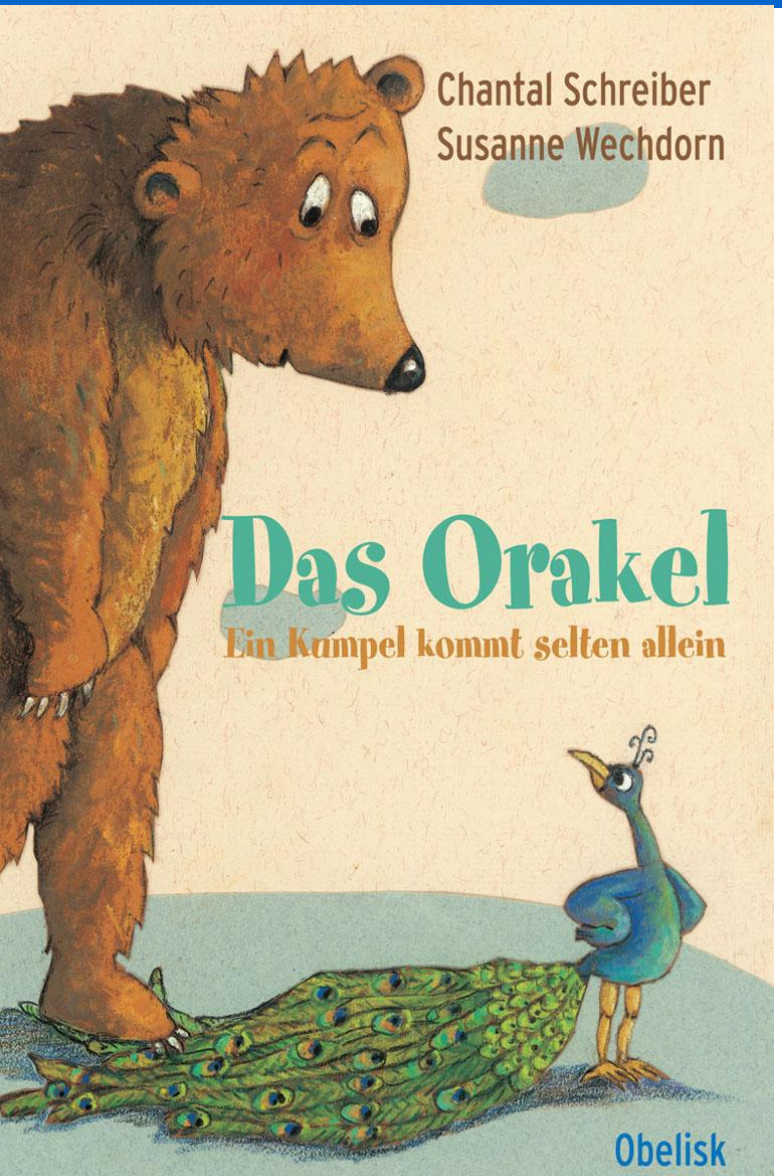
a,~~c~~,b 

~~a,b~~,~~c~~,d 

Deviations may be underestimated but are always detected!

How about discovery?

Discovery can also be distributed!



- **Assume the set of activities is split in overlapping sets.**
- **Split log L in sublogs L^i based on these sets.**
- **Discover a model SN^i per sublog.**
- **Merge the models SN^i into SN and return result.**

All the earlier guarantees hold, e.g., L is perfectly fitting SN if and only if each projected log is L^i is perfectly fitting SN^i

Example

$$L = [\langle a,b,c,d,h\rangle^{30}, \langle a,c,b,d,e,b,c,d,j,g,f,k\rangle^{28}, \langle a,c,b,d,e,c,b,d,h\rangle^{28}, \langle a,c,b,d,e,c,b,d,j,f,g,k\rangle^{27}, \langle a,b,c,d,e,c,i,d,j,g,f,k\rangle^{26}, \langle a,b,c,d,e,i,c,d,h\rangle^{26}, \langle a,i,c,d,e,c,b,d,j,g,f,k\rangle^{25}, \langle a,i,c,d,j,f,g,k\rangle^{24}, \langle a,c,b,d,e,b,c,d,h\rangle^{24}, \langle a,b,c,d,e,b,c,d,j,g,f,k\rangle^{24}, \langle a,c,b,d,e,c,i,d,h\rangle^{22}, \langle a,b,c,d,e,c,i,d,h\rangle^{22}, \langle a,c,i,d,j,f,g,k\rangle^{21}, \langle a,c,i,d,j,g,f,k\rangle^{20}, \langle a,c,i,d,e,b,c,d,h\rangle^{18}, \langle a,c,i,d,e,c,i,d,h\rangle^{17}, \langle a,i,c,d,h\rangle^{17}, \langle a,c,b,d,j,g,f,k\rangle^{17}, \langle a,c,b,d,h\rangle^{15}, \langle a,i,c,d,e,i,c,d,j,f,g,k\rangle^{14}, \langle a,c,i,d,e,c,b,d,h\rangle^{14}, \langle a,c,i,d,e,c,b,d,j,f,g,k\rangle^{12}, \langle a,b,c,d,e,i,c,d,j,f,g,k\rangle^{12}, \langle a,b,c,d,e,c,b,d,h\rangle^{11}, \langle a,c,b,d,j,f,g,k\rangle^{10}, \langle a,i,c,d,j,g,f,k\rangle^{9}, \langle a,i,c,d,e,b,c,d,h\rangle^{9}, \langle a,i,c,d,e,c,b,d,h\rangle^{9}, \langle a,i,c,d,e,c,b,d,j,f,g,k\rangle^{8}, \langle a,c,b,d,e,b,c,d,j,f,g,k\rangle^{8}, \langle a,b,c,d,e,b,c,d,h\rangle^{7}, \langle a,c,b,d,e,i,c,d,j,g,f,k\rangle^{7}, \langle a,i,c,d,e,c,i,d,j,g,f,k\rangle^{6}, \langle a,c,i,d,h\rangle^{6}, \langle a,c,b,d,e,i,c,d,h\rangle^{6}, \langle a,i,c,d,e,b,c,d,j,g,f,k\rangle^{6}, \langle a,b,c,d,j,f,g,k\rangle^{5}, \langle a,i,c,d,e,i,c,d,h\rangle^{5}, \langle a,i,c,d,e,c,i,d,h\rangle^{4}, \langle a,c,i,d,e,i,c,d,h\rangle^{4}, \langle a,b,c,d,e,c,i,d,j,f,g,k\rangle^{4}, \langle a,b,c,d,e,c,b,d,j,g,f,k\rangle^{4}, \langle a,b,c,d,j,g,f,k\rangle^{3}, \langle a,c,i,d,e,b,c,d,j,f,g,k\rangle^{3}, \langle a,b,c,d,e,b,c,d,j,f,g,k\rangle^{3}, \langle a,i,c,d,e,b,c,d,j,f,g,k\rangle^{3}, \langle a,b,c,d,e,c,b,d,j,f,g,k\rangle^{3}, \langle a,c,i,d,e,i,c,d,j,f,g,k\rangle^{2}, \langle a,c,i,d,e,i,c,d,e,b,c,d,e,b,c,d,j,g,f,k\rangle^{2}, \langle a,b,c,d,e,i,c,d,j,g,f,k\rangle^{2}, \langle a,c,b,d,e,c,i,d,j,g,f,k\rangle^{2}, \langle a,c,i,d,e,c,i,d,j,f,g,k\rangle^{2}, \langle a,c,b,d,e,b,c,d,e,c,b,d,j,g,f,k\rangle^{2}, \langle a,c,i,d,e,c,b,d,j,g,f,k\rangle^{2}, \langle a,c,b,d,e,i,c,d,j,f,g,k\rangle^{2}, \langle a,c,i,d,e,i,c,d,e,i,c,d,j,g,f,k\rangle^{1}, \langle a,c,b,d,e,c,i,d,j,f,g,k\rangle^{1}, \langle a,i,c,d,e,c,i,d,e,c,i,d,e,i,c,d,e,i,c,d,j,g,f,k\rangle^{1}, \langle a,i,c,d,e,c,i,d,j,f,g,k\rangle^{1}, \langle a,c,i,d,e,c,i,d,j,g,f,k\rangle^{1}]$$

$$A^1 = \{a, b, d, e, i\}$$

$$A^2 = \{a, c, d, e\}$$

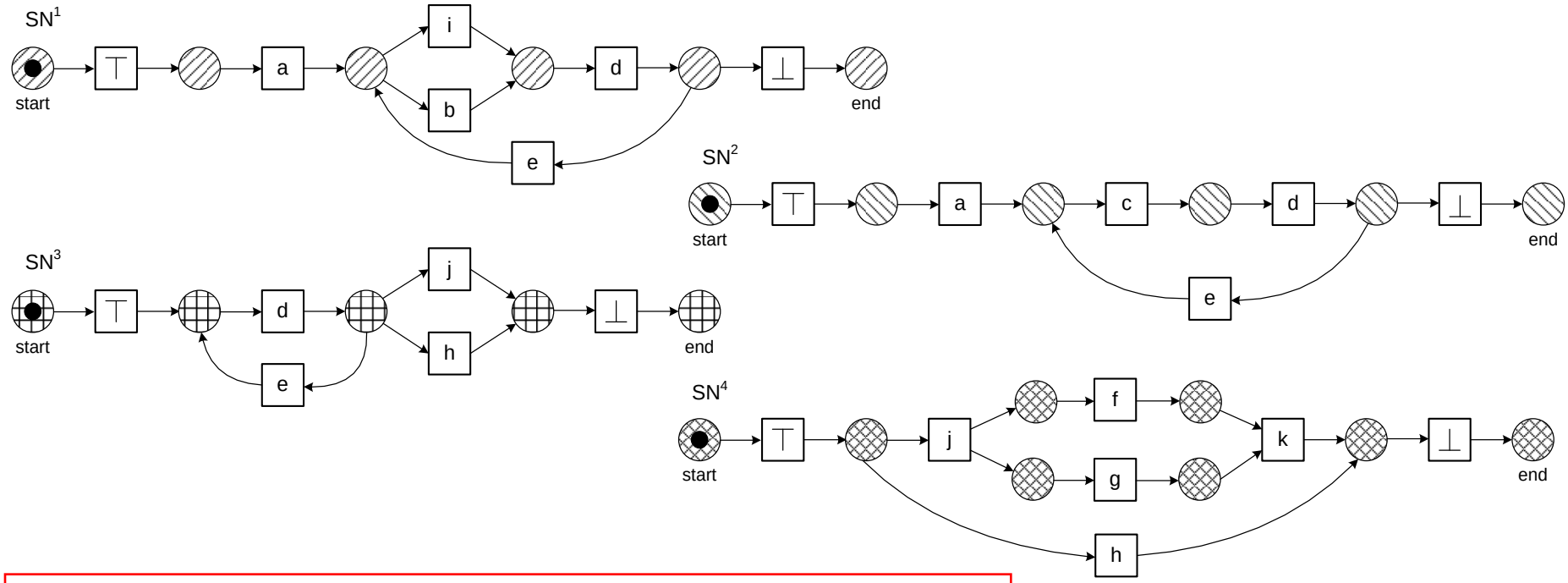
$$A^3 = \{d, e, h, j\}$$

$$A^4 = \{f, g, h, j, k\}$$

- **Let's use the Alpha algorithm**
- **Alpha algorithm is not very suitable for discovering transition bordered subnets.**
- **By adding start and end activities we get (in this case) perfectly fitting subnets.**

$$L' = [\langle\top\rangle \cdot \sigma \cdot \langle\bot\rangle \mid \sigma \in L_o] = [\langle \top, a, b, c, d, h, \bot\rangle^{30}, \langle \top, a, c, b, d, e, b, c, d, j, g, f, k, \bot\rangle^{28}, \langle \top, a, c, b, d, e, c, b, d, h, \bot\rangle^{28}, \langle \top, a, c, b, d, e, c, b, d, j, f, g, k, \bot\rangle^{27}, \langle \top, a, b, c, d, e, c, i, d, j, g, f, k, \bot\rangle^{26}, \ldots].$$

Discover model per sublog (Alpha algorithm)



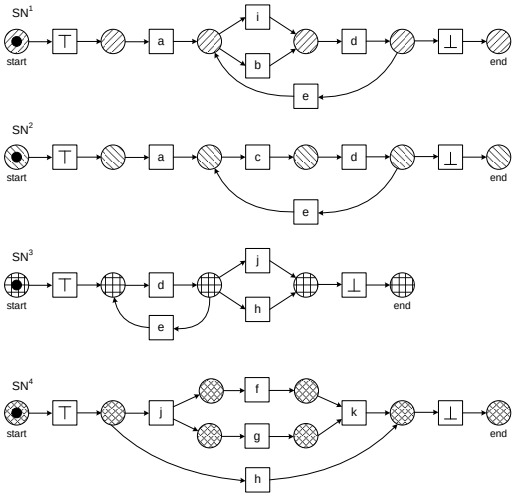
$L^1 = [\langle \top, a, b, d, \bot \rangle^{30}, \langle \top, a, b, d, e, b, d, \bot \rangle^{28}, \langle \top, a, b, d, e, b, d, \bot \rangle^{28}, \langle \top, a, b, d, e, b, d, \bot \rangle^{27}, \langle \top, a, b, d, e, i, d, \bot \rangle^{26}, \ldots]$

$L^2 = [\langle \top, a, c, d, \bot \rangle^{30}, \langle \top, a, c, d, e, c, d, \bot \rangle^{28}, \langle \top, a, c, d, e, c, d, \bot \rangle^{28}, \langle \top, a, c, d, e, c, d, \bot \rangle^{27}, \langle \top, a, c, d, e, c, d, \bot \rangle^{26}, \ldots]$

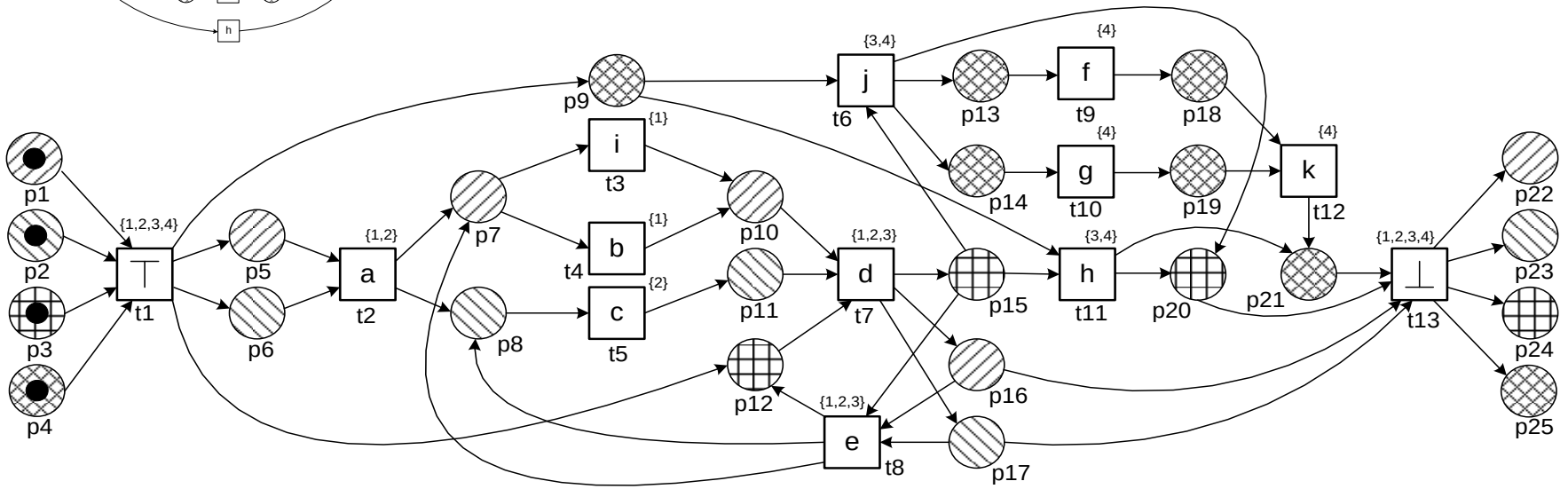
$L_3 = [\langle \top, d, h, \bot \rangle^{30}, \langle \top, d, e, d, j, \bot \rangle^{28}, \langle \top, d, e, d, h, \bot \rangle^{28}, \langle \top, d, e, d, j, \bot \rangle^{27}, \langle \top, d, e, d, j, \bot \rangle^{26}, \ldots]$

$L_4 = [\langle \top, h, \bot \rangle^{30}, \langle \top, j, g, f, k, \bot \rangle^{28}, \langle \top, h, \bot \rangle^{28}, \langle \top, j, f, g, k, \bot \rangle^{27}, \langle \top, j, g, f, k, \bot \rangle^{26}, \ldots]$

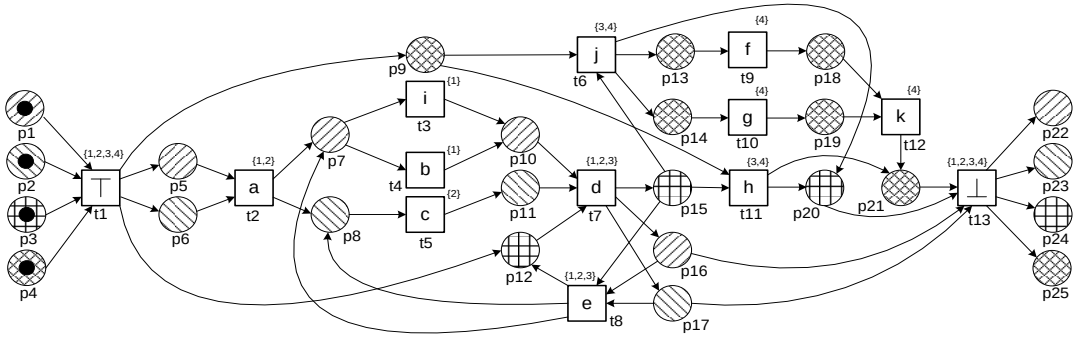
Merge models

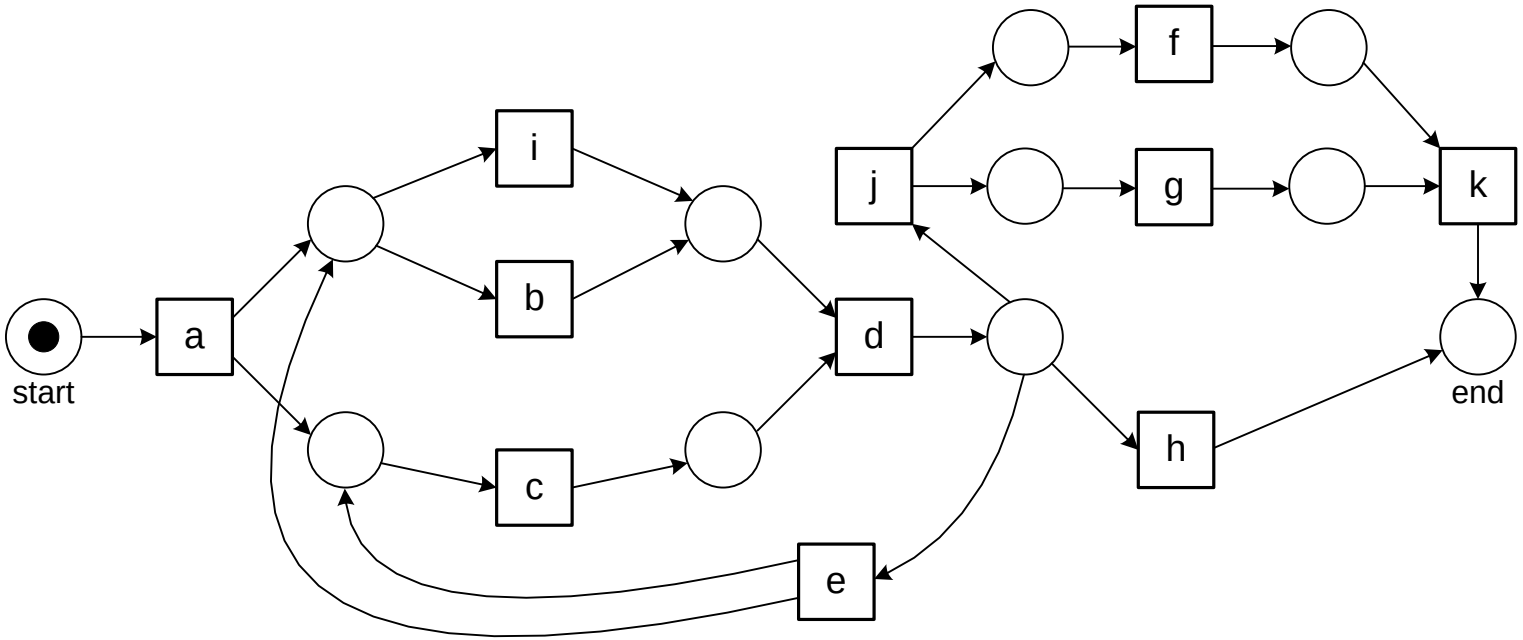


L is perfectly fitting SN because each projected log is L^i is perfectly fitting SN^i



Simplify by removing redundant places and initialization





Log is perfectly fitting this model !

Passages

Wil van der Aalst. Decomposing Process Mining Problems Using Passages. In Petri Nets 2012, LNCS 7347 , pages 72-91. Springer, 2012.

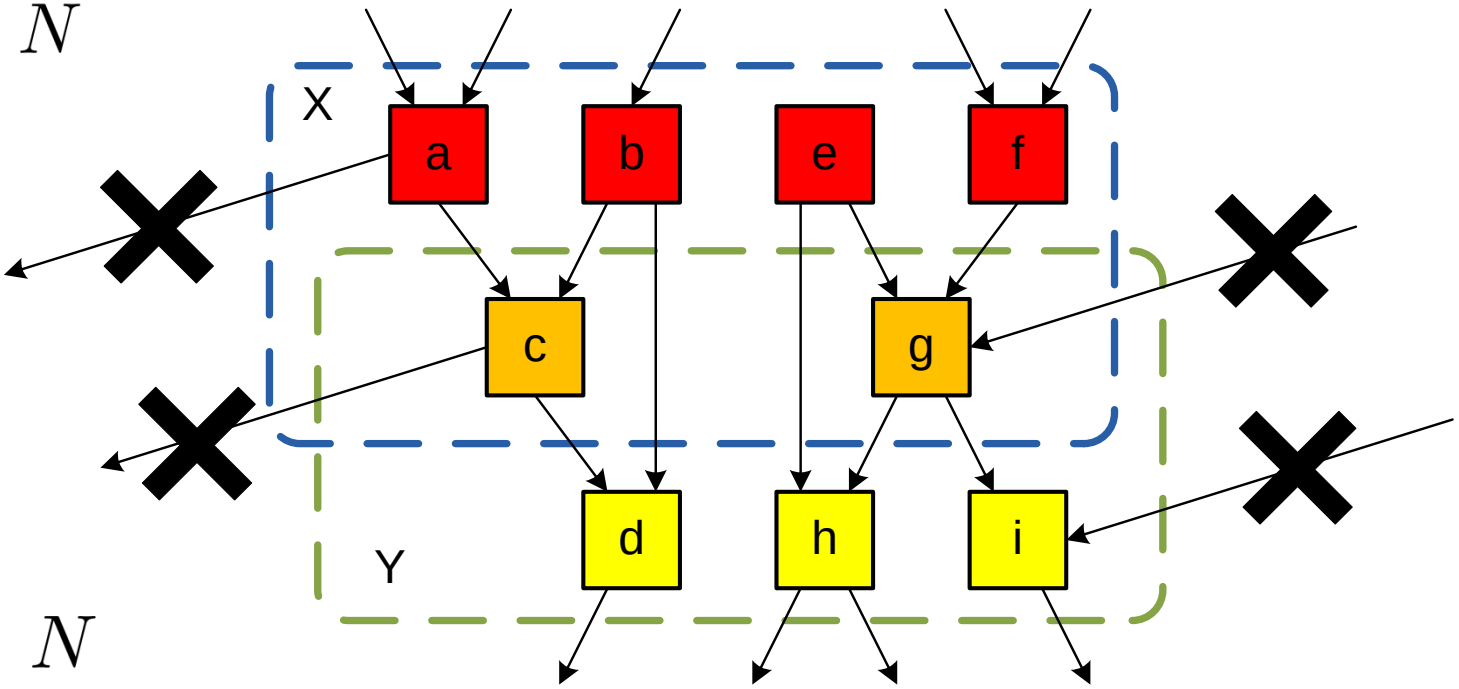
Passages

Moscow GUM

Passage *P=(X,Y)*

causal dependency:
may trigger or enable

$$\emptyset \neq X \subseteq N$$

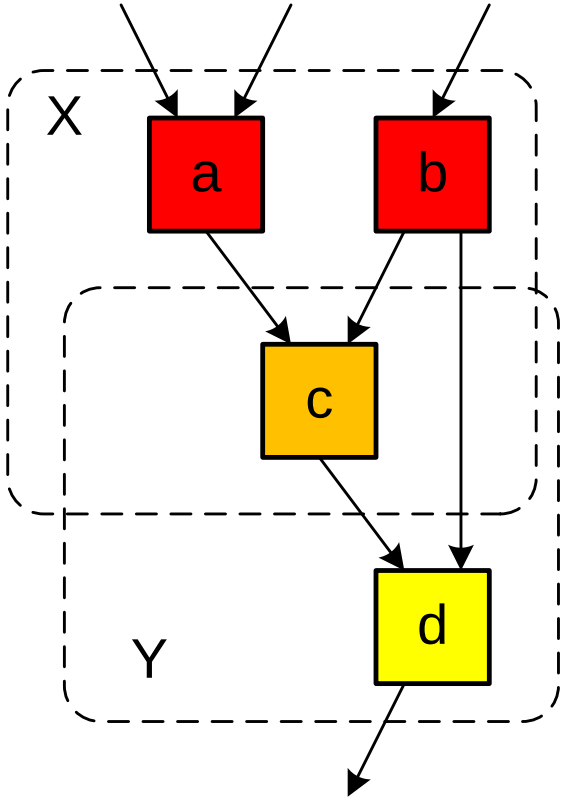


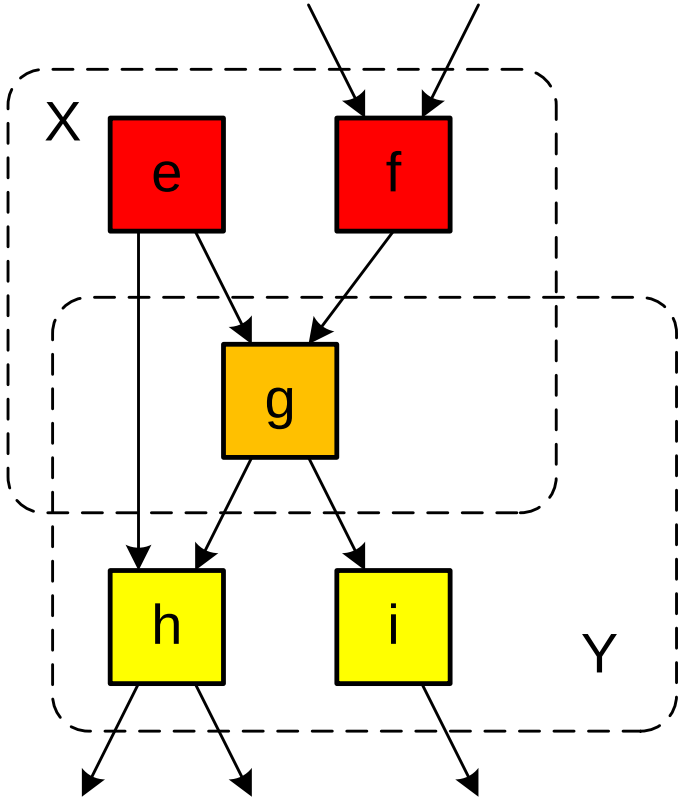
$$\emptyset \neq Y \subseteq N$$

$$X \overset{G}{\bullet} = Y$$

$$X = \overset{G}{\bullet} Y$$

Minimal passages



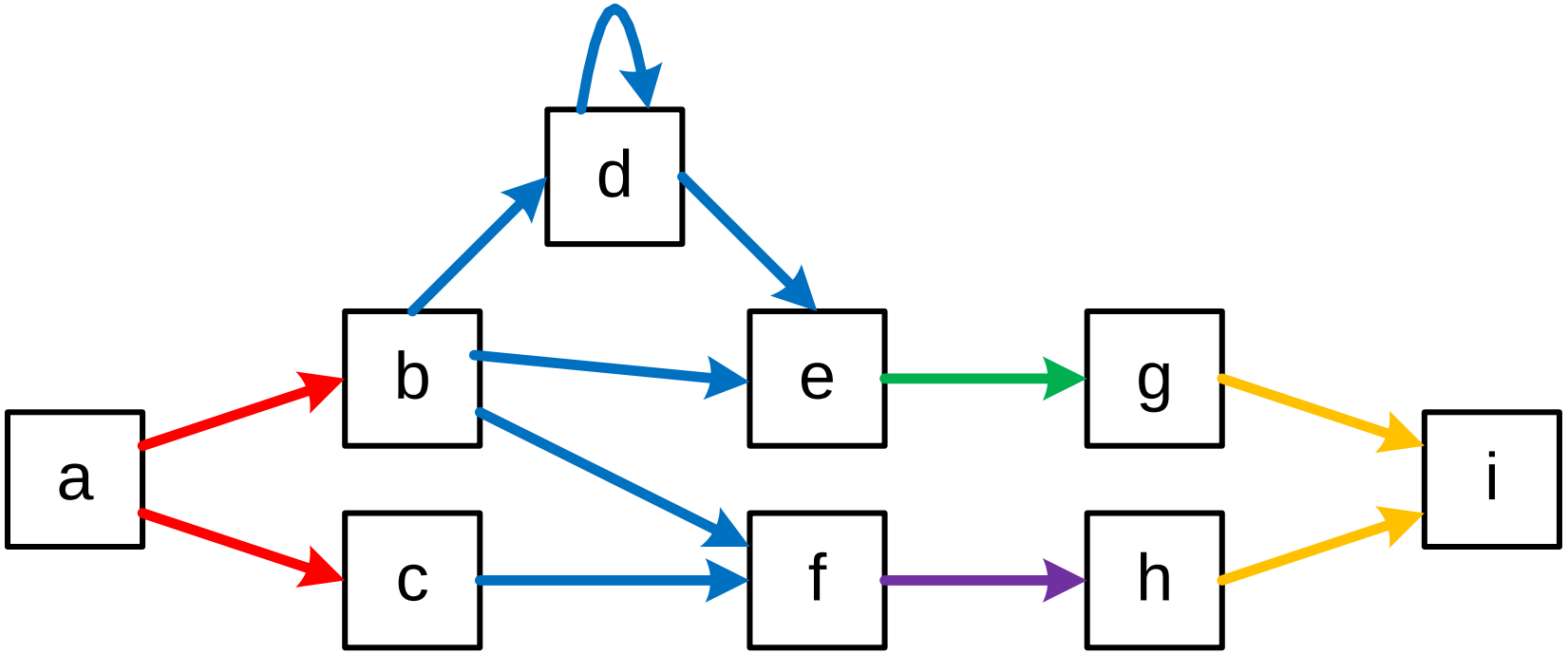


$$X \overset{G}{\bullet} = Y$$

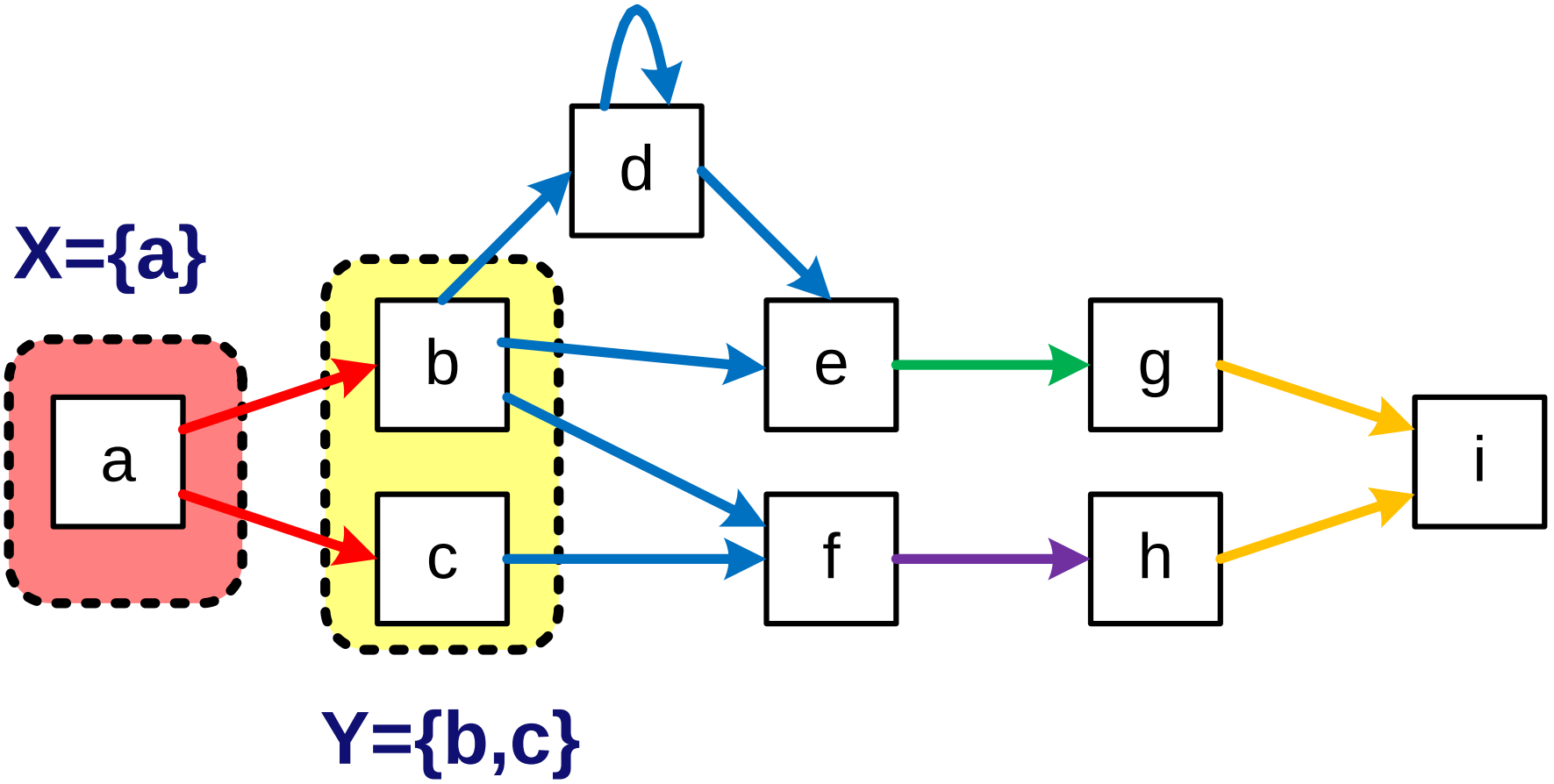
$$X = \overset{G}{\bullet} Y$$

a passage is minimal if it does not contain smaller passages

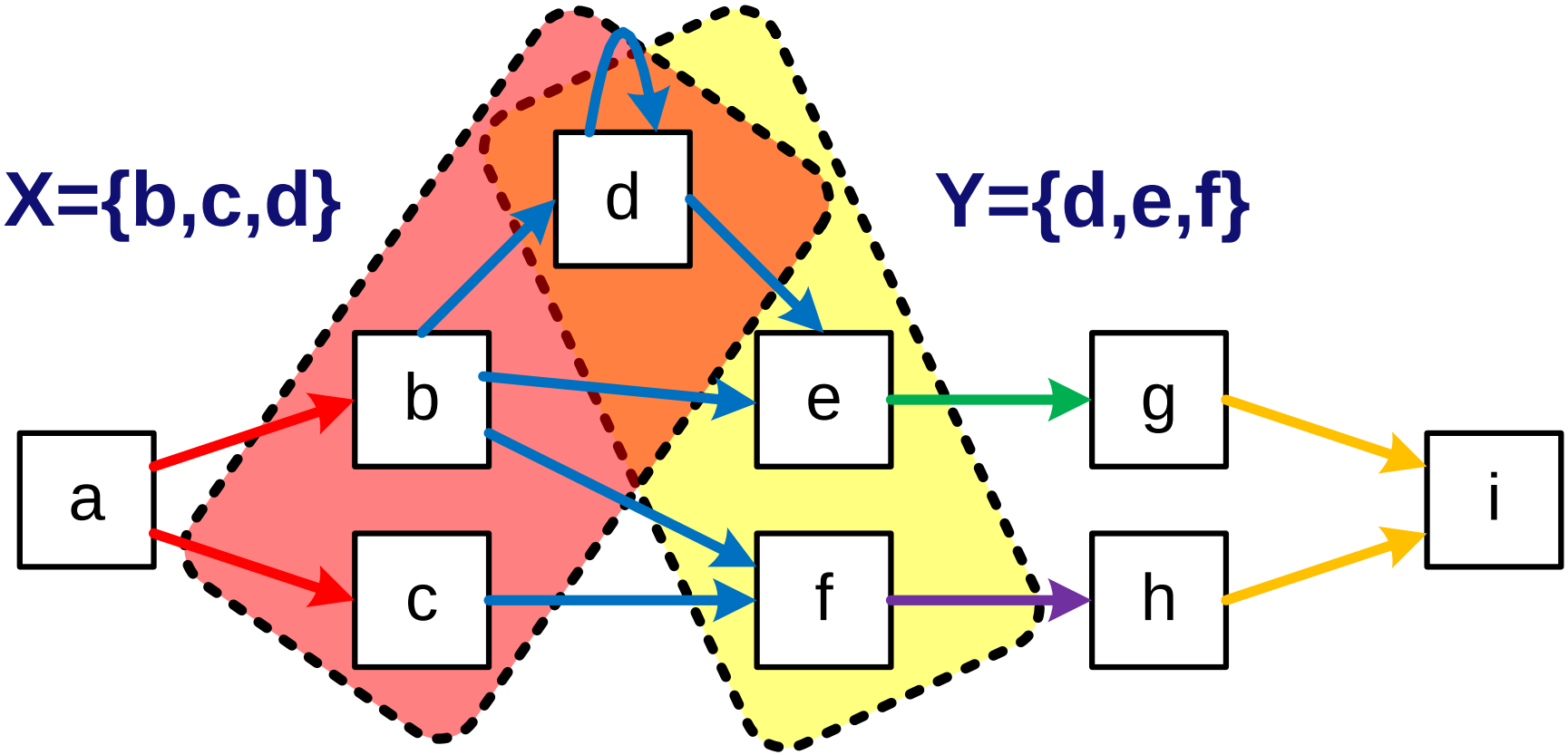
Passages define an equivalence relation on the edges in the graph



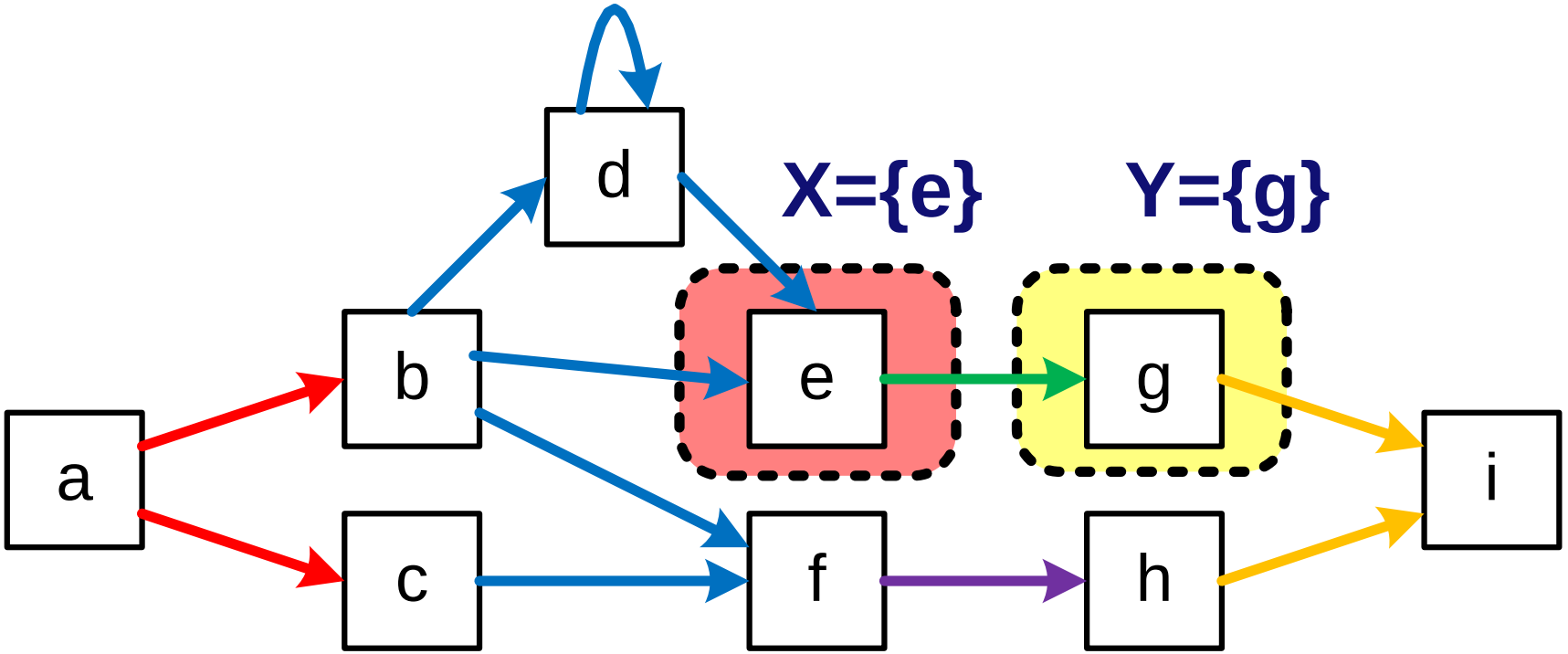
Minimal passage 1: (X,Y) = ({a},{b,c})



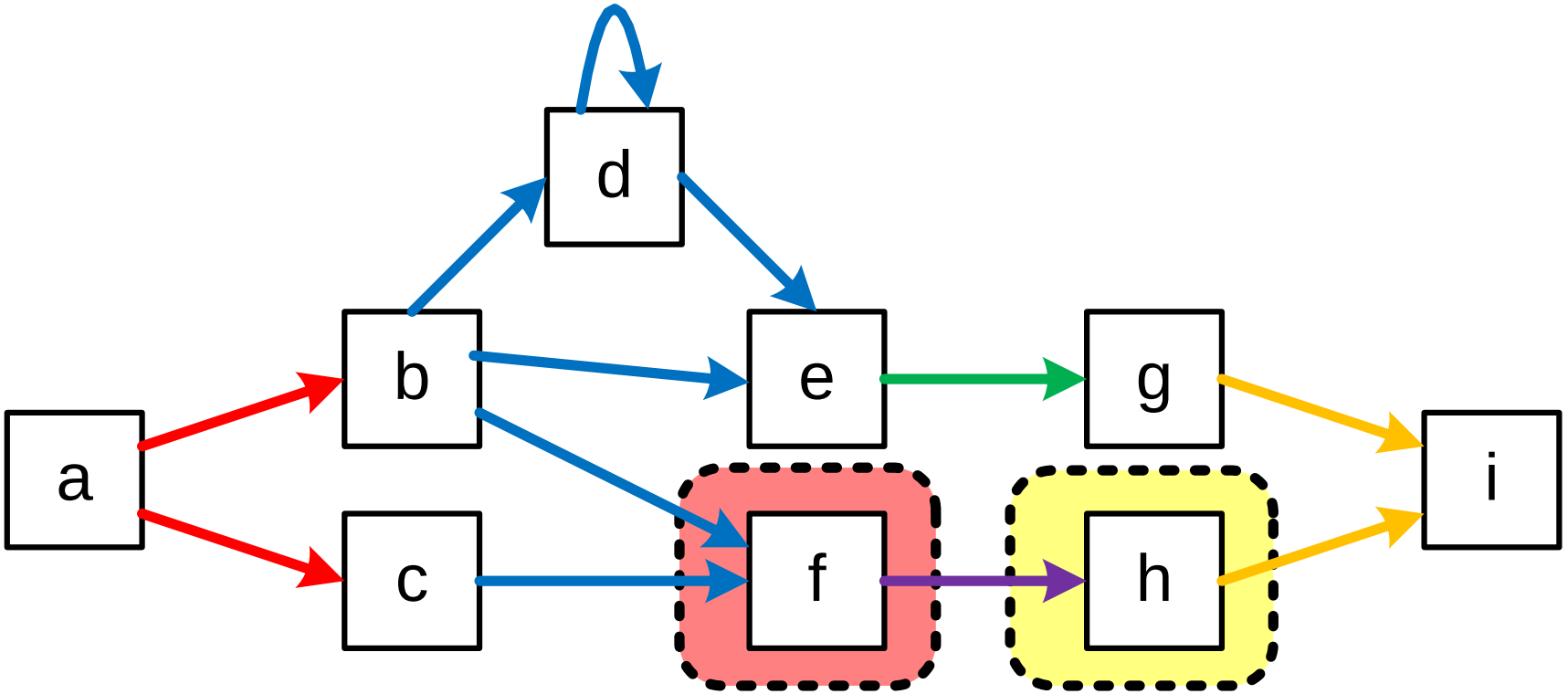
Minimal passage 2: (X,Y) = ({b,c,d},{d,e,f})



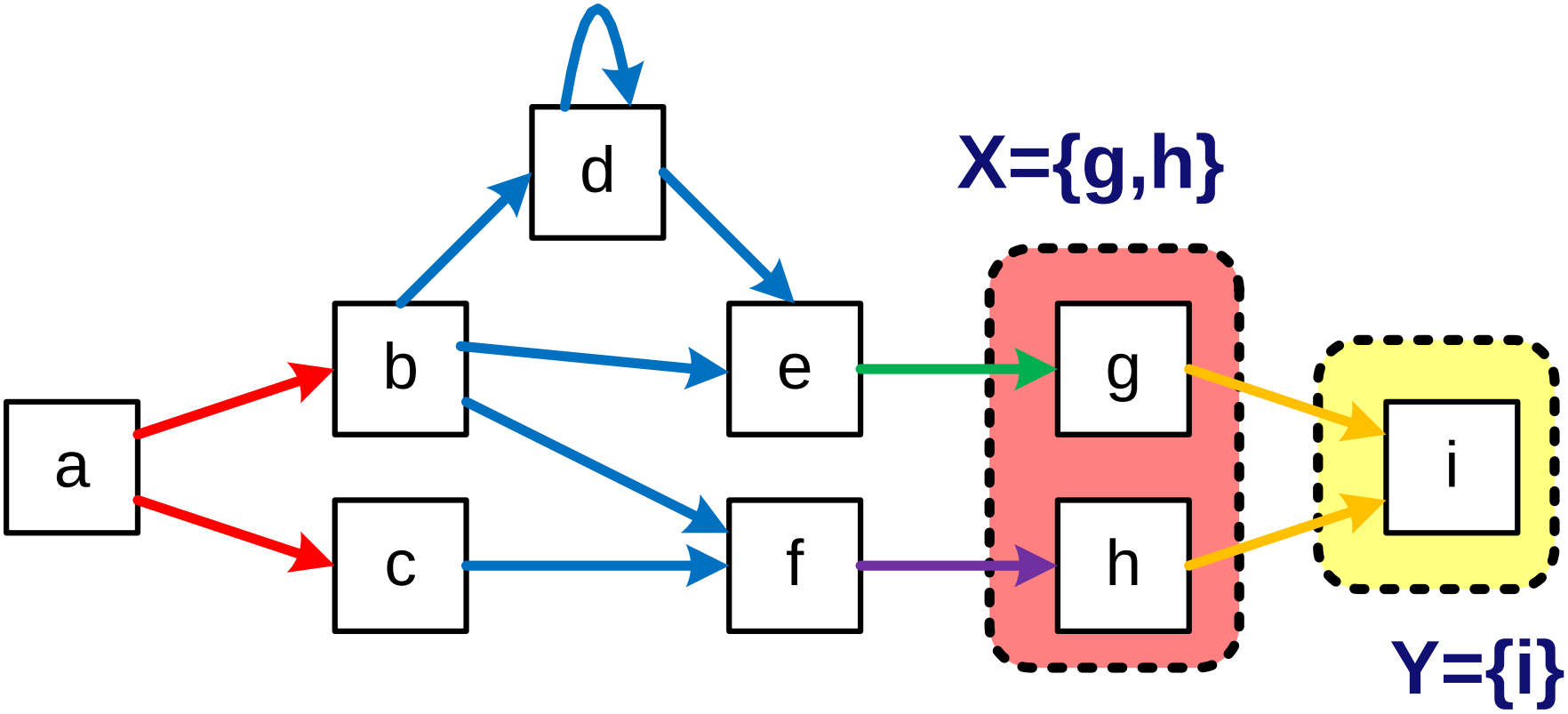
Minimal passage 3: (X,Y) = ({e},{g})



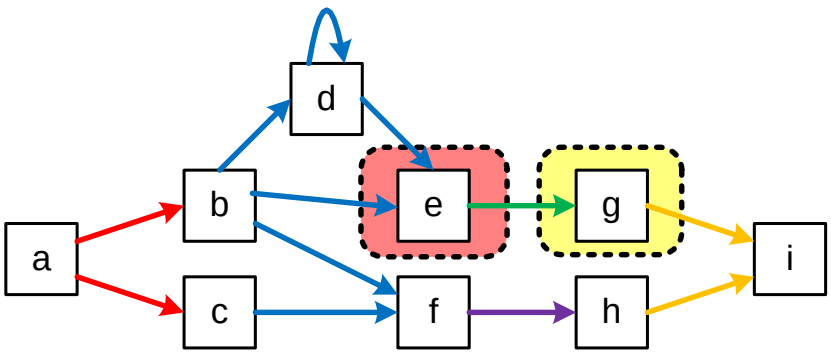
Minimal passage 4: (X,Y) = ({f},{h})



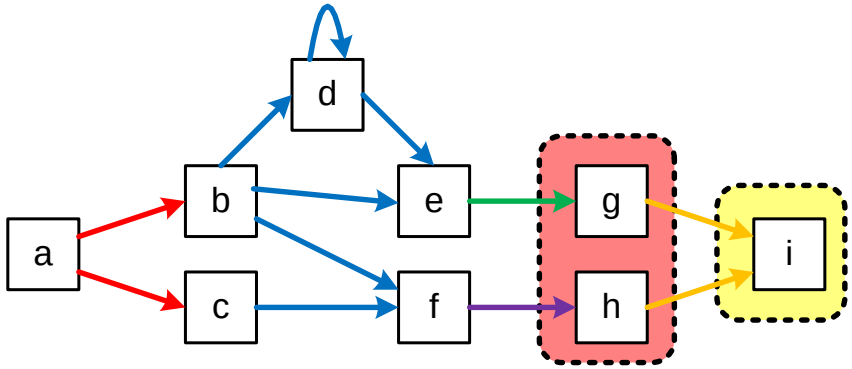
Minimal passage 5: (X,Y) = ({g,h},{i})



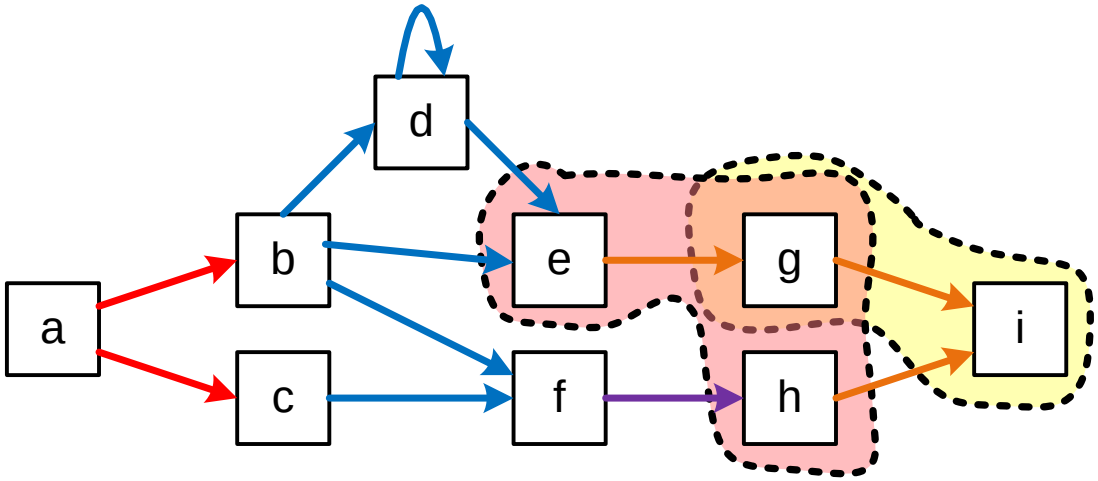
Union of two passages



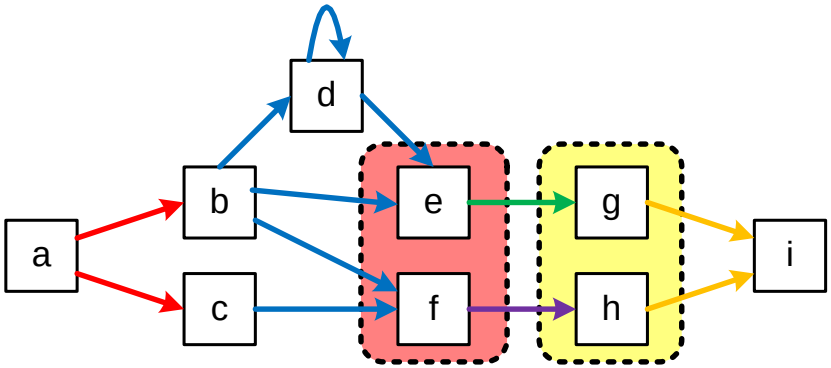
$\cup$



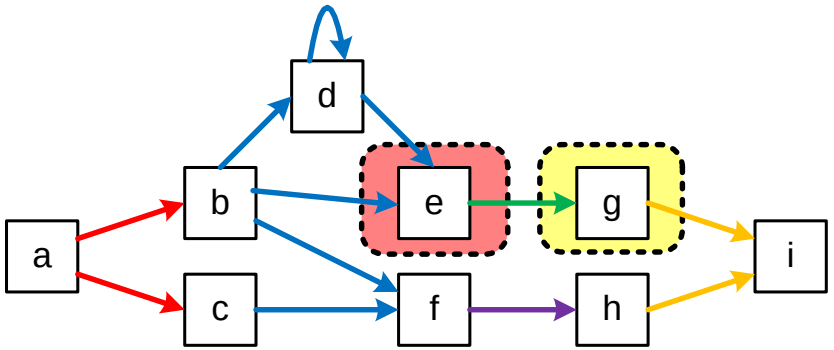
$$P_1 \cup P_2 = (X_1 \cup X_2, Y_1 \cup Y_2)$$



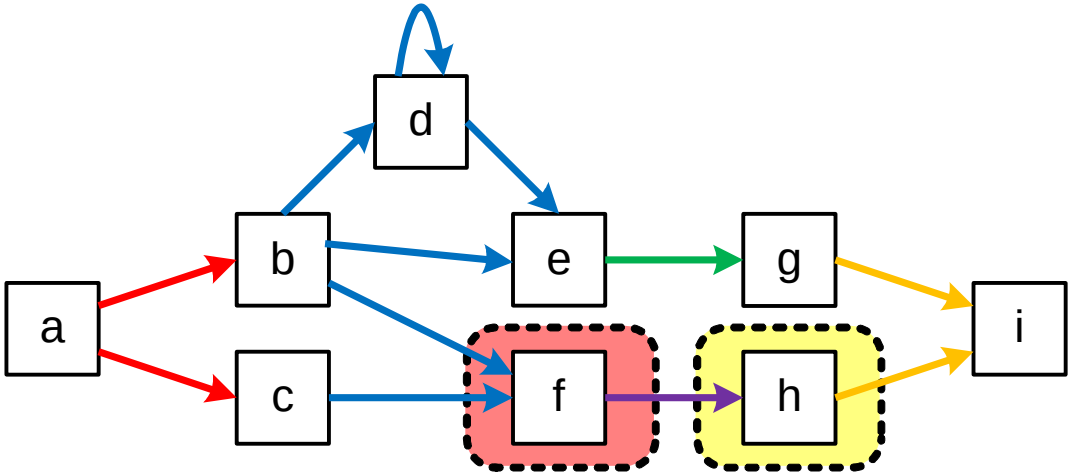
Difference of two passages



\



$$P_1 \setminus P_2 = (X_1 \setminus X_2, Y_1 \setminus Y_2)$$

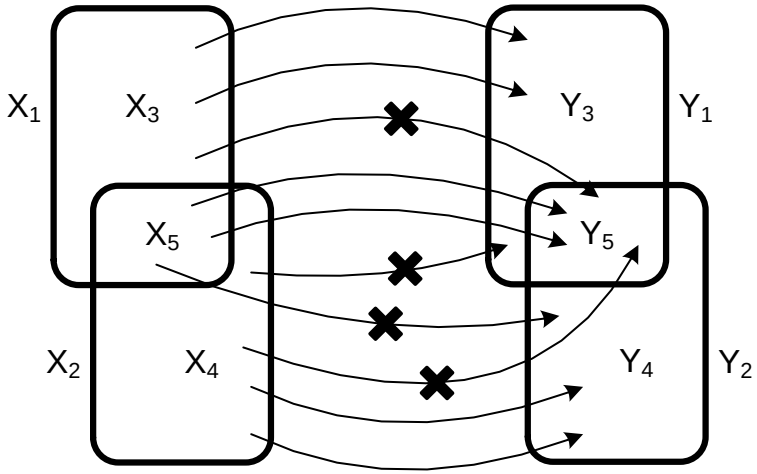


Properties of passages

If

$$P_1 = (X_1, Y_1) \in pas(G)$$

$$P_2 = (X_2, Y_2) \in pas(G)$$



then:

$P_3 = P_1 \setminus P_2$ is a passage if $P_3 \neq (\emptyset, \emptyset)$,

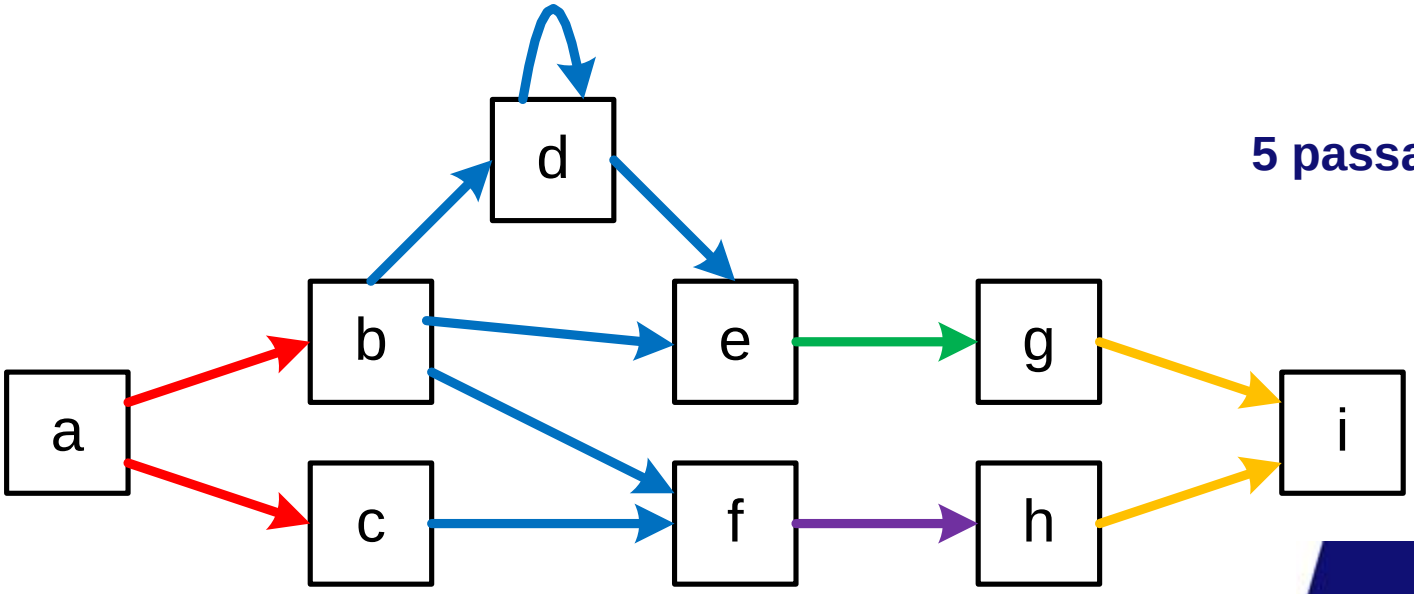
$P_4 = P_2 \setminus P_1$ is a passage if $P_4 \neq (\emptyset, \emptyset)$,

$P_5 = P_1 \cap P_2$ is a passage if $P_5 \neq (\emptyset, \emptyset)$, and

$P_6 = P_1 \cup P_2$ is a passage.

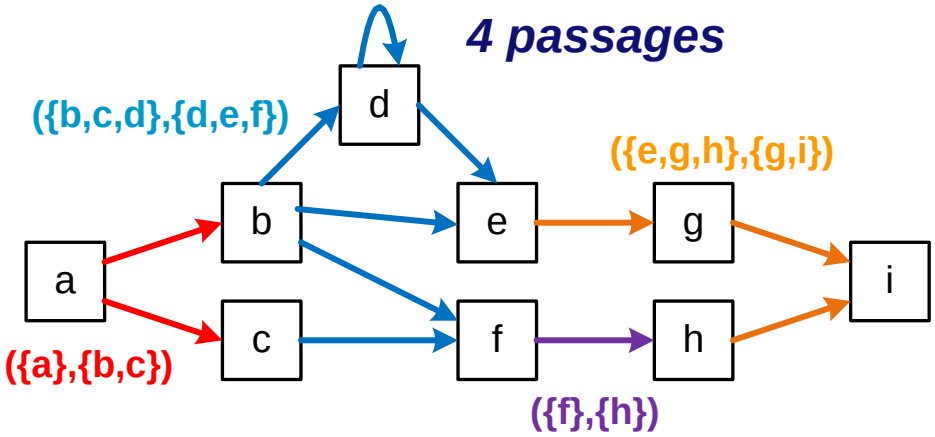
Passage partitioning

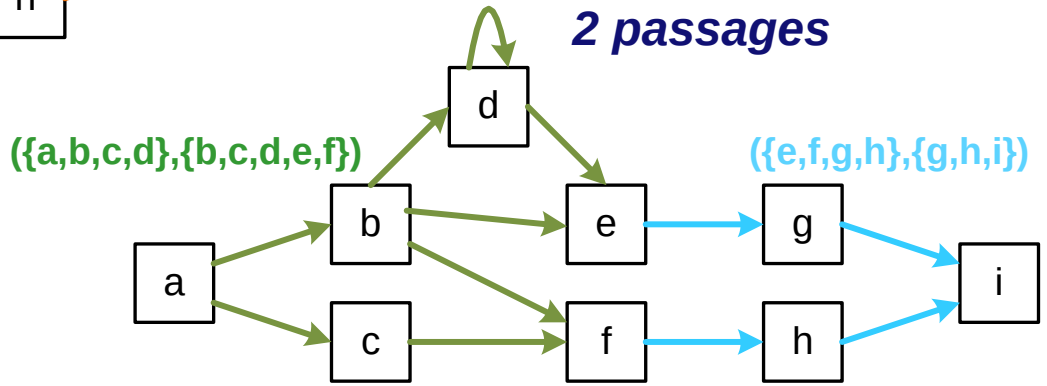
- **$P_1, P_2, \ldots, P_n$ is a passage partitioning if and only if**
  - **$P_1, P_2, \ldots, P_n$ are passages,**
  - **the passages are disjoint, and**
  - **the passages cover all edges.**

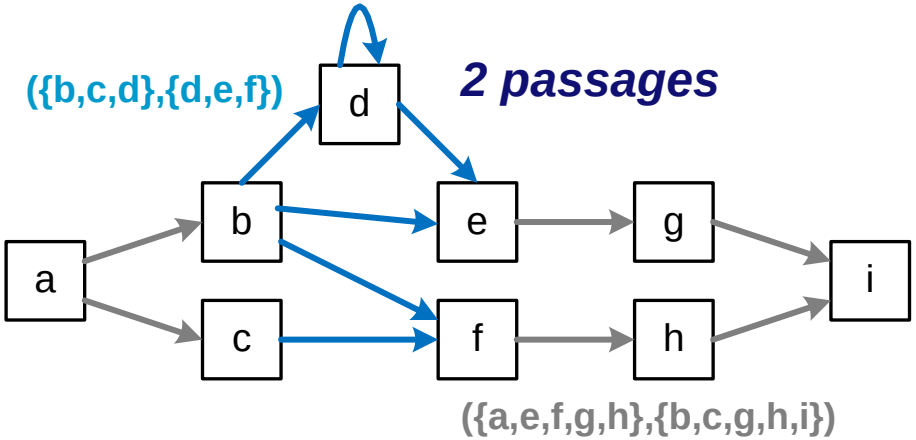


5 passages

More examples of passage partitionings

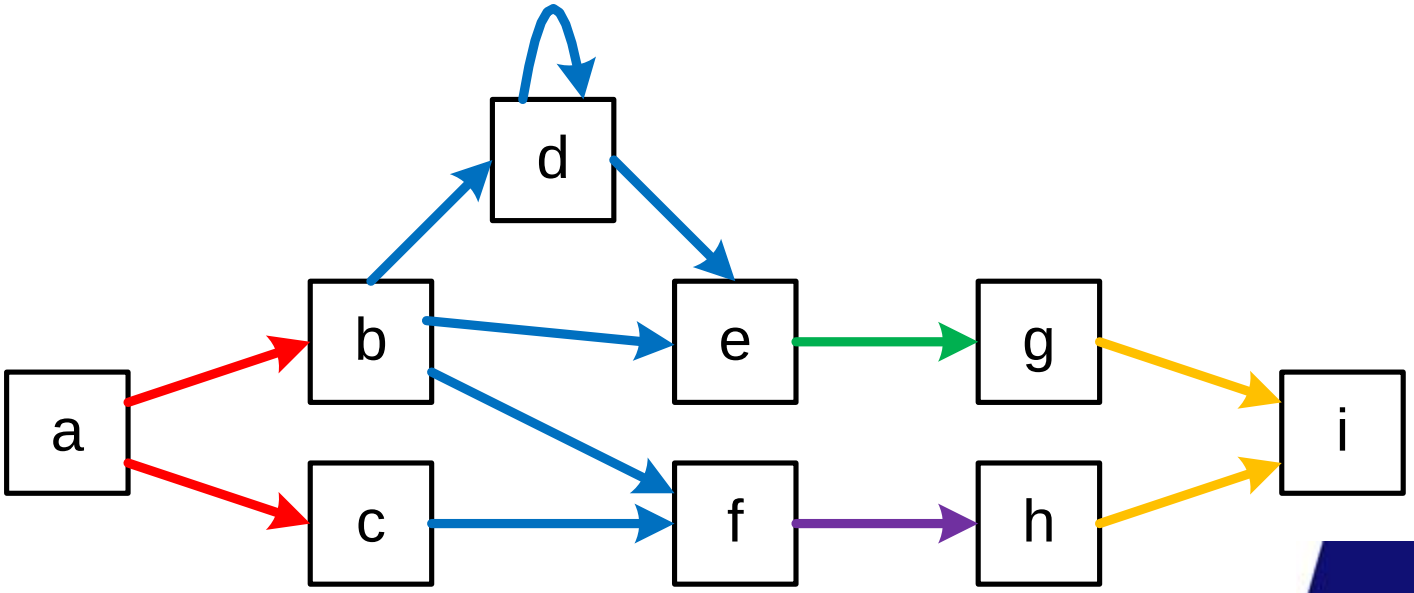






Minimal passages

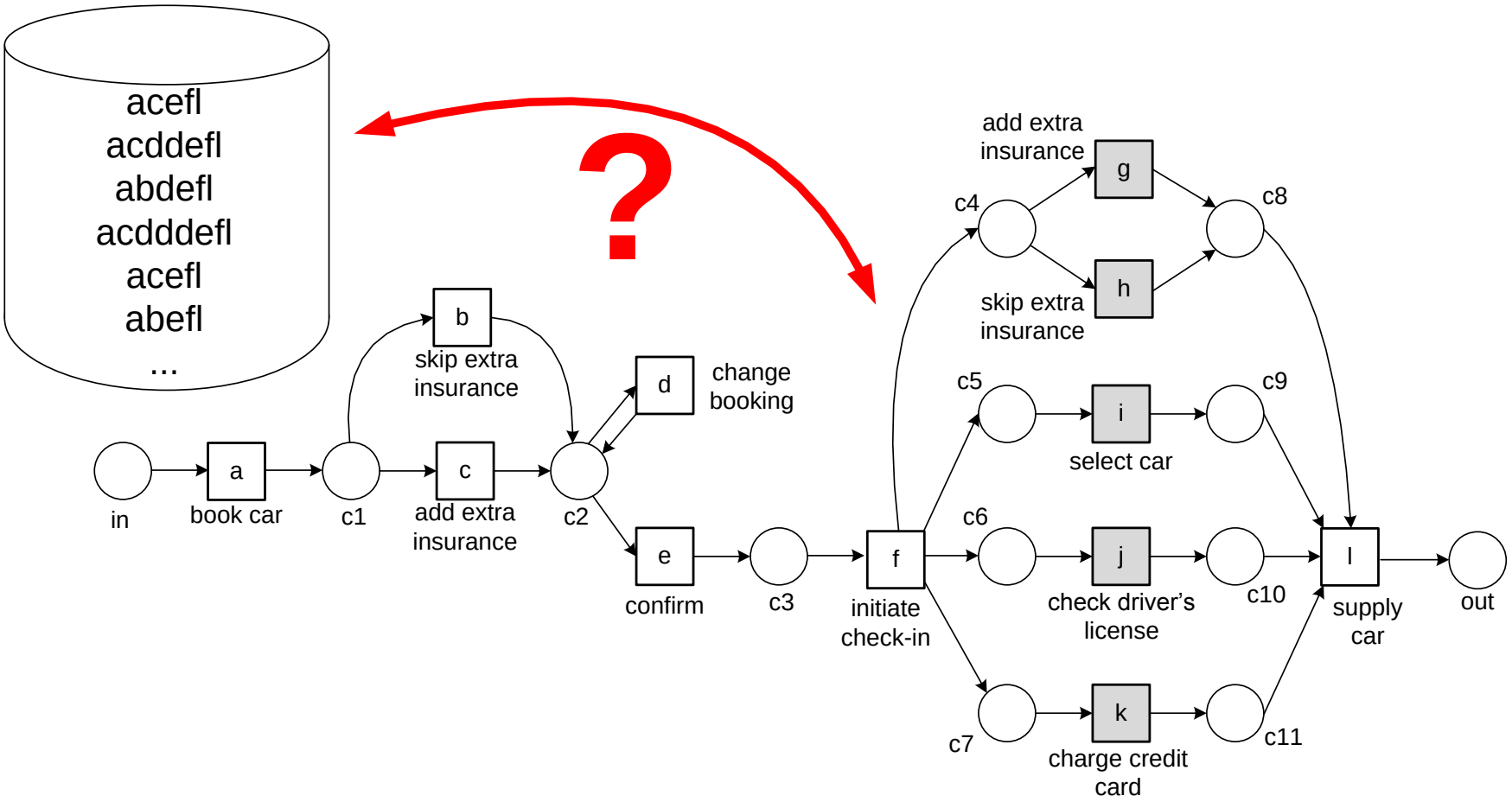
- **A passage is minimal if it does not contain smaller passages.**
- **Each edge is involved in precisely one minimal passage.**
- **Passages are composed of minimal passages.**
- **The minimal passages form a passage partitioning.**



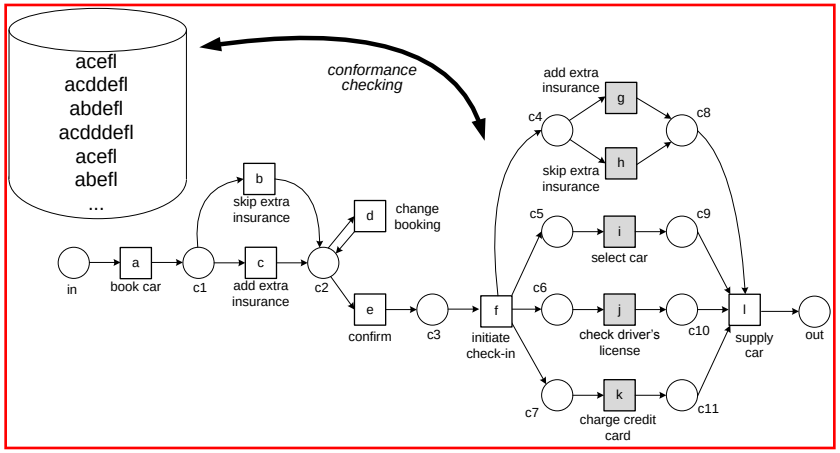
Passages are special case of decomposition

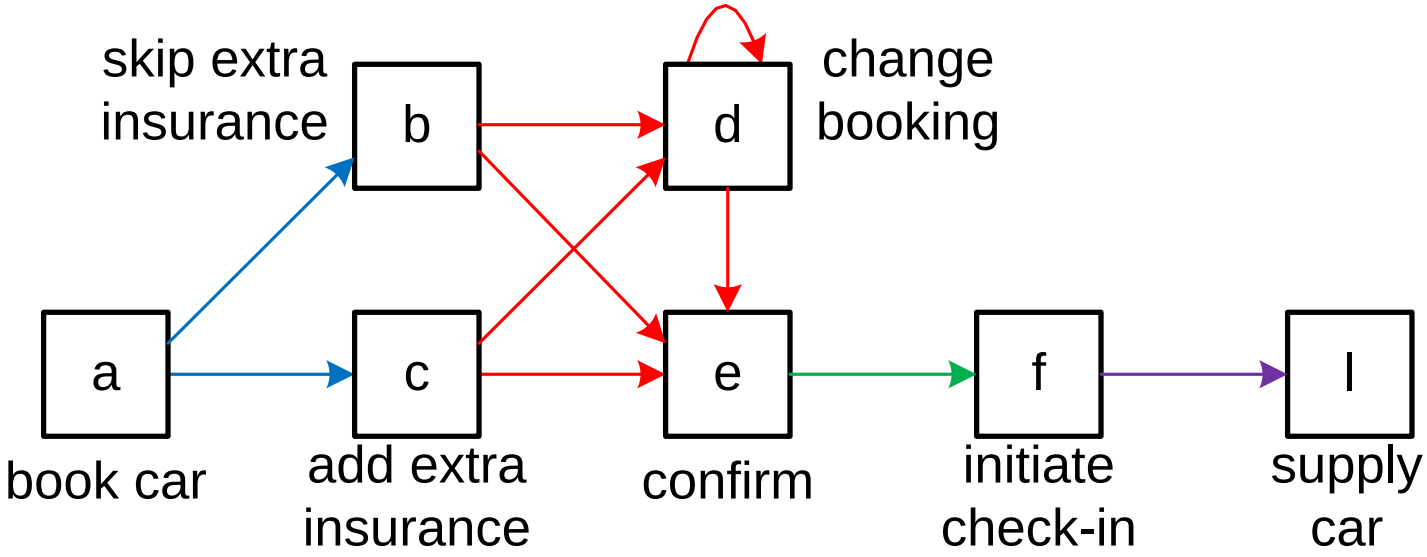
- **Conformance checking:**
  1. Compute causal structure from Petri net
  2. Compute (minimal) passages and construct corresponding subnets
  3. Subnets provide a valid decomposition
  4. Hence, conformance checking* can be decomposed/distributed!
- **Process discovery:**
  1. Compute causal structure from event log*
  2. Compute (minimal) passages and construct corresponding sublogs
  3. Discover subnets for sublogs*
  4. Merge subnets into overall model

Conformance checking

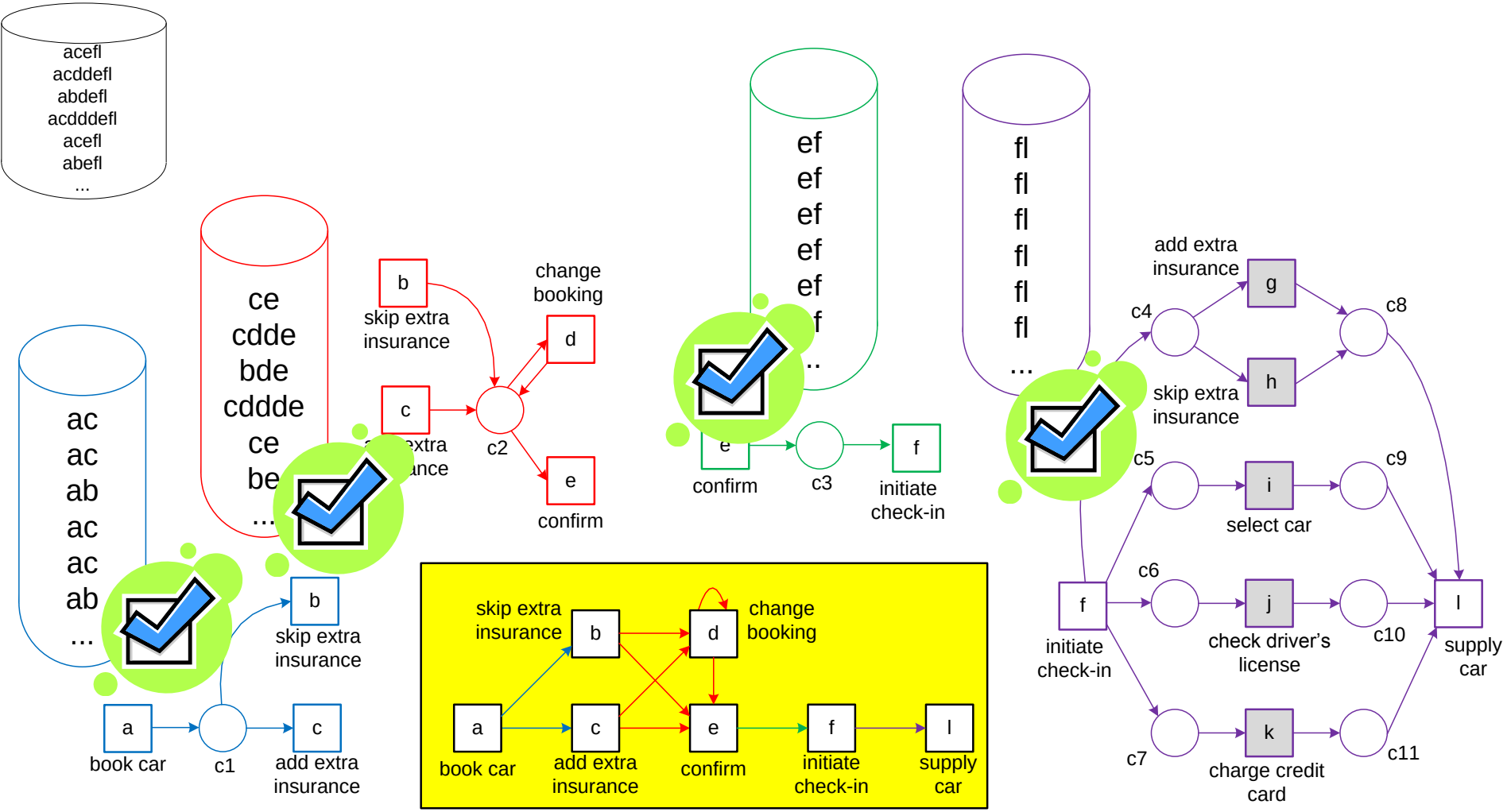


Create Skeleton

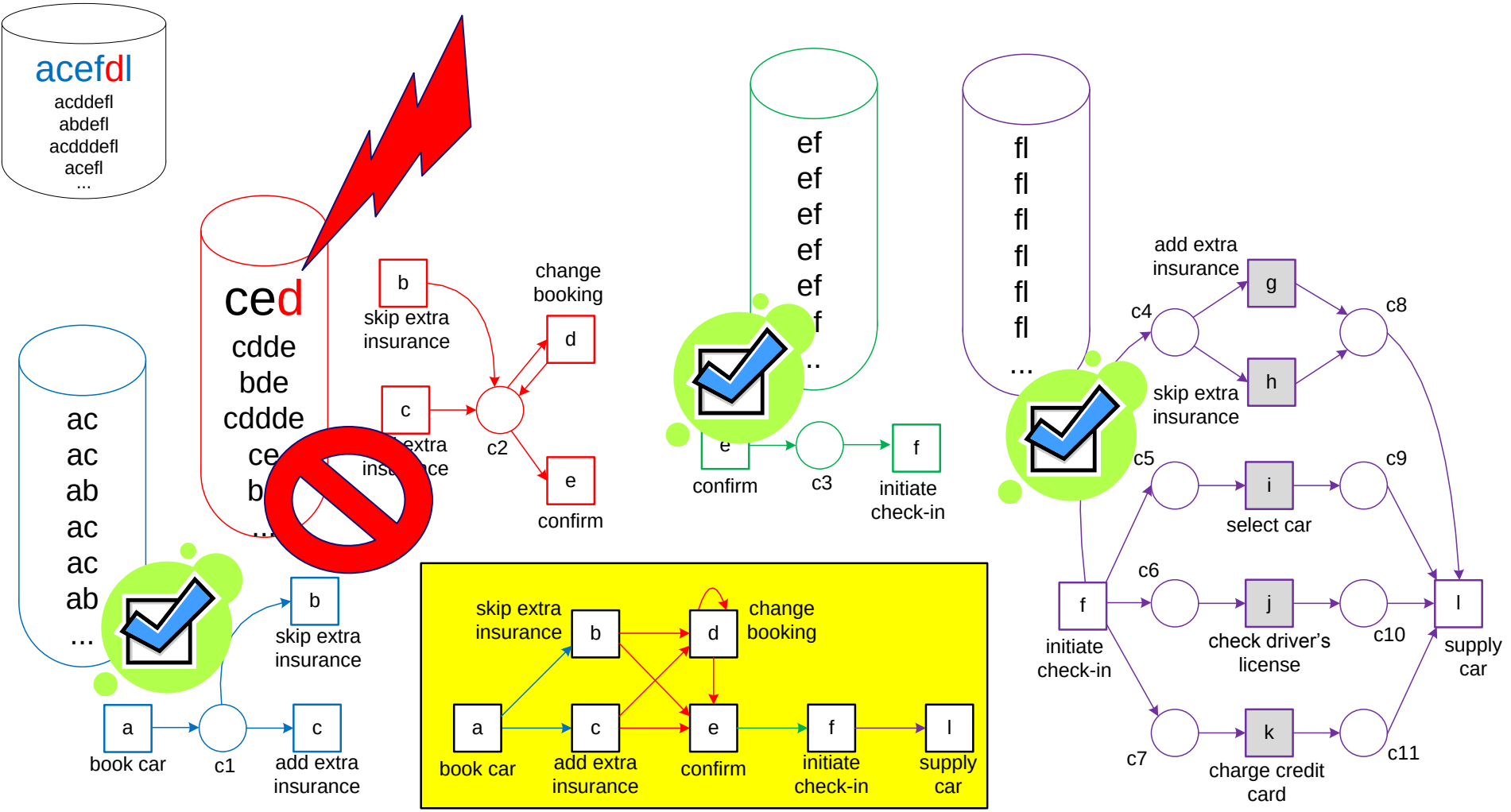




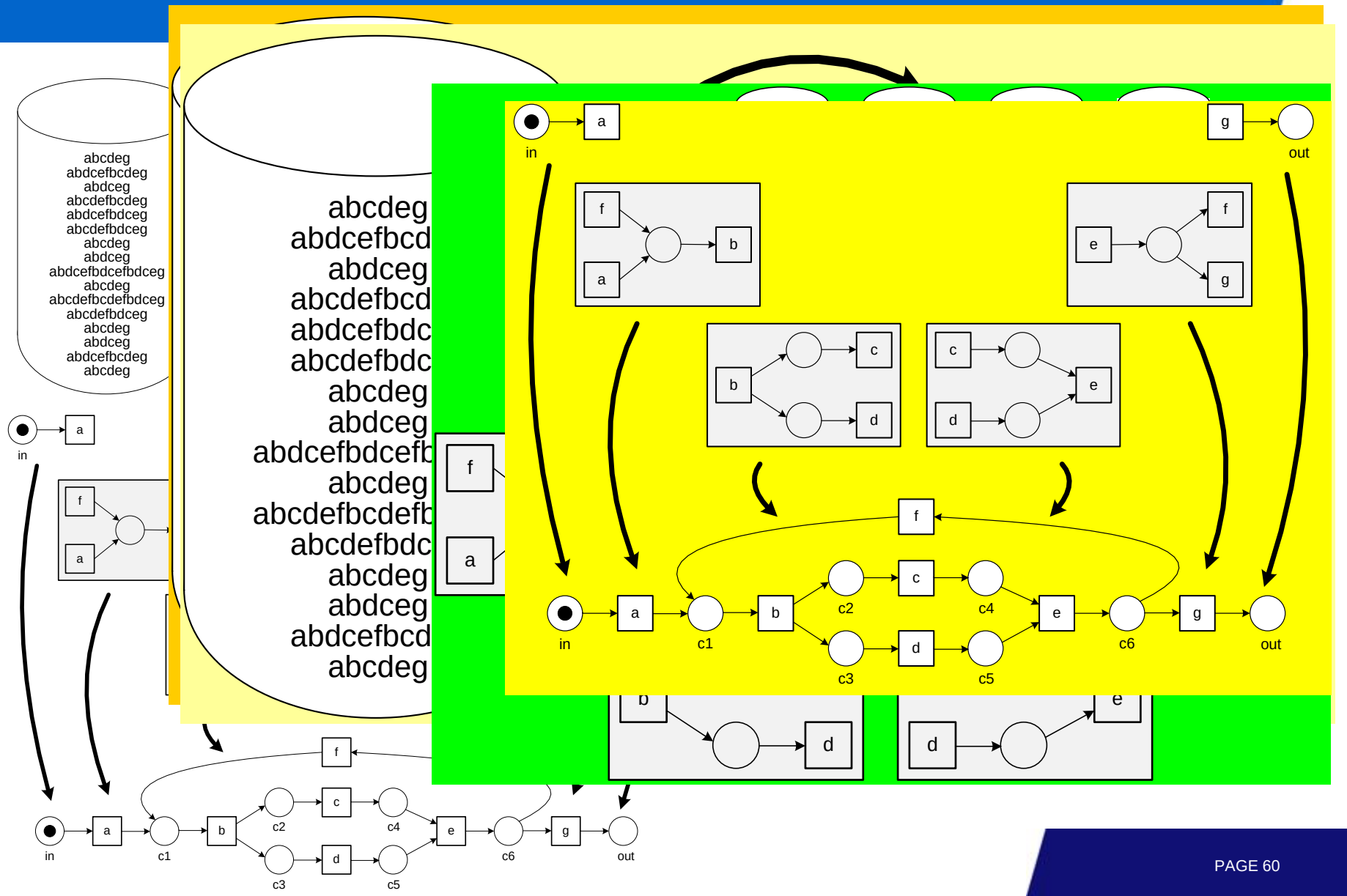
Net fragments per passage



Suppose d is executed too late



Discovery example



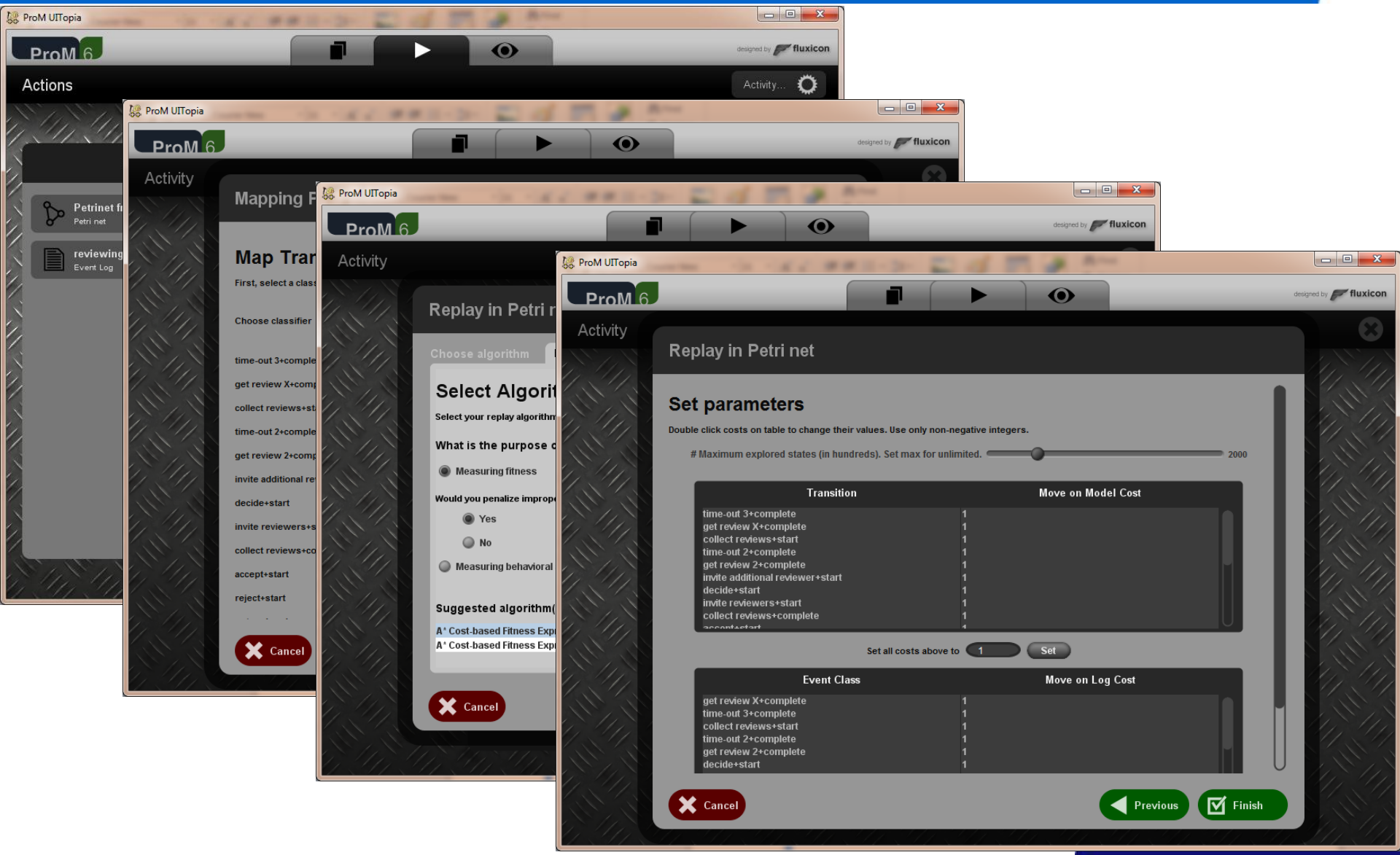
Tool Support in ProM

Tool Support in ProM
(implemented by Eric Verbeek)

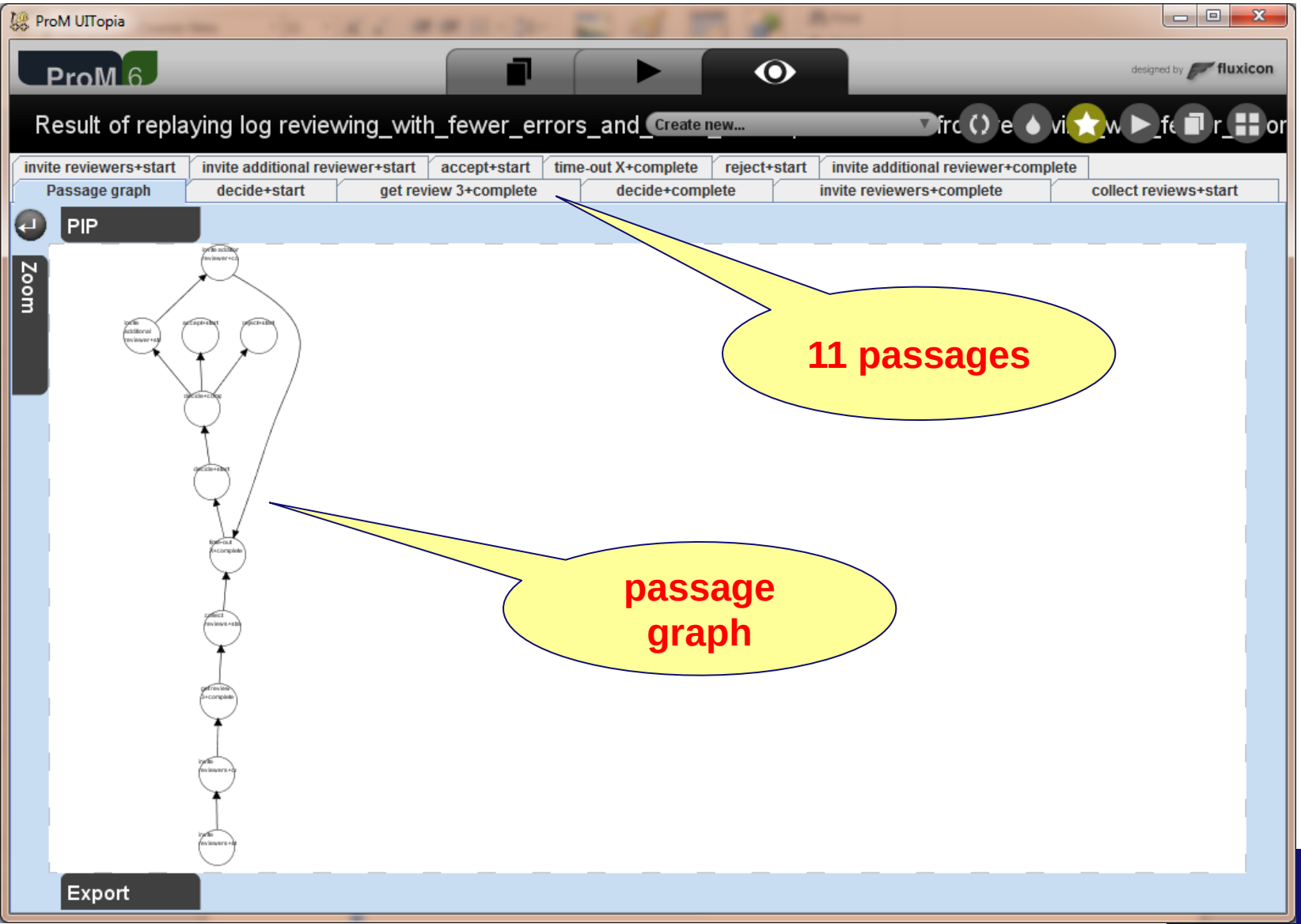
discovery

conformance

Passage-Based Conformance Checking

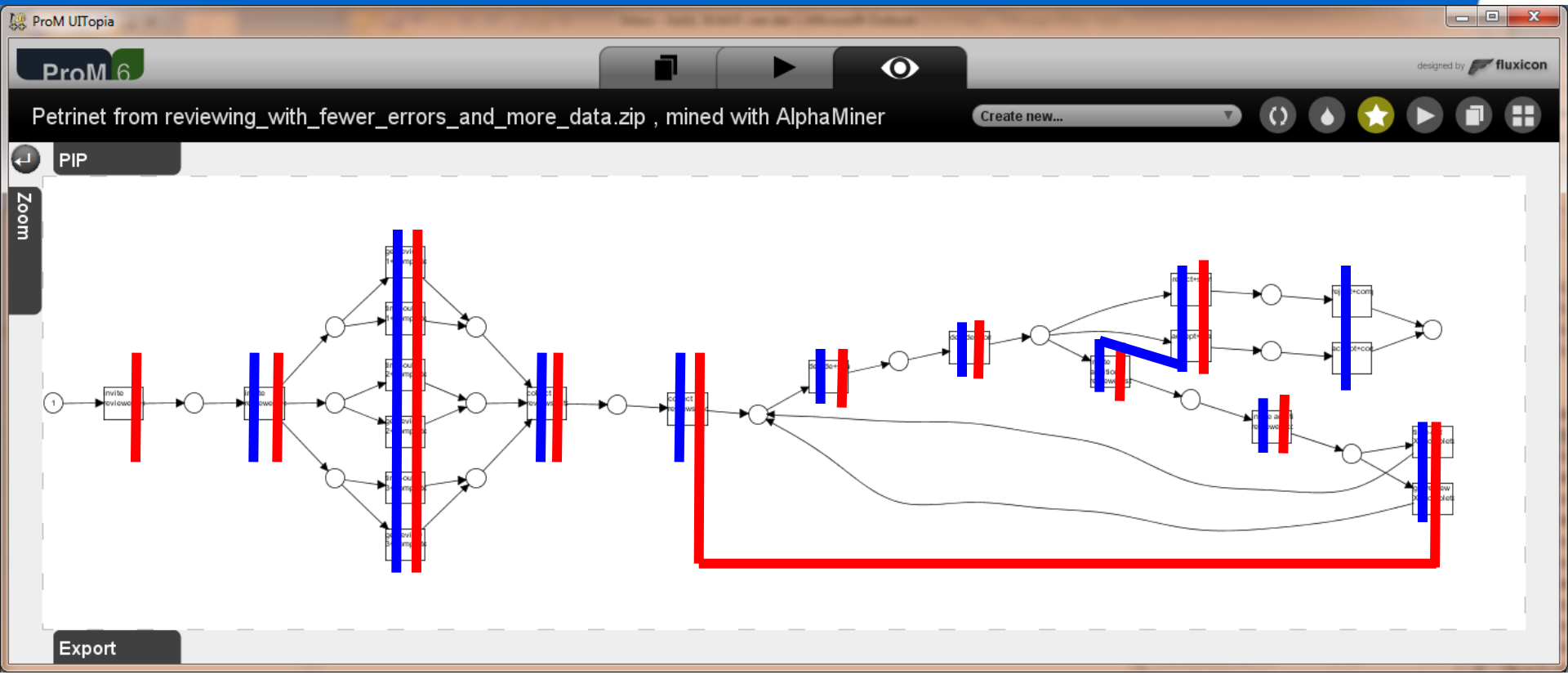


Result


ProM UITopia
ProM 6
designed by fluxicon
Result of replaying log reviewing_with_fewer_errors_and_
Create new...
invite reviewers+start
invite additional reviewer+start
accept+start
time-out X+complete
reject+start
invite additional reviewer+complete
Passage graph
decide+start
get review 3+complete
decide+complete
invite reviewers+complete
collect reviews+start
PIP
Zoom
11 passages
passage graph
Export

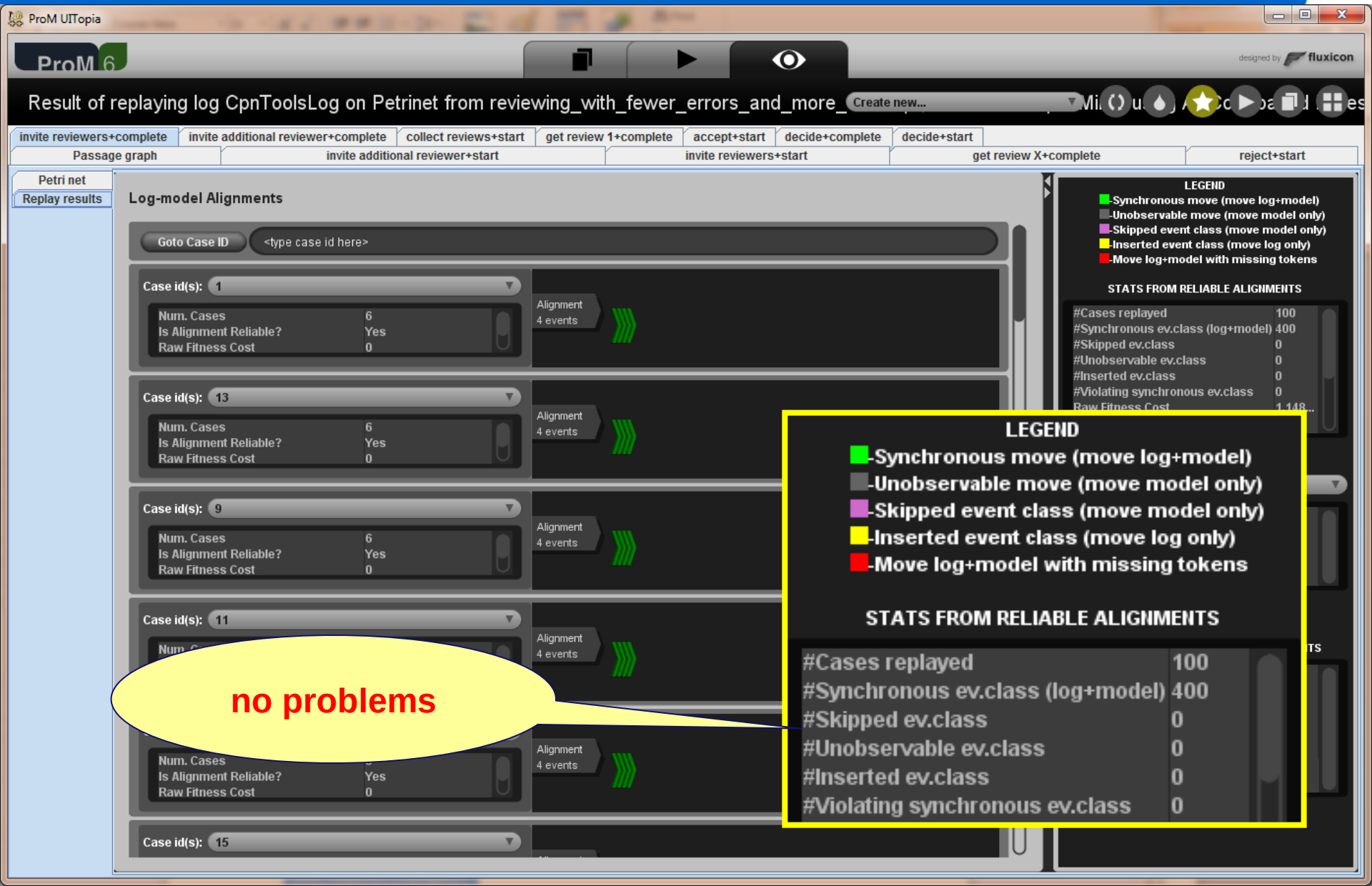
Example Passage

Process Model has 11 Passages

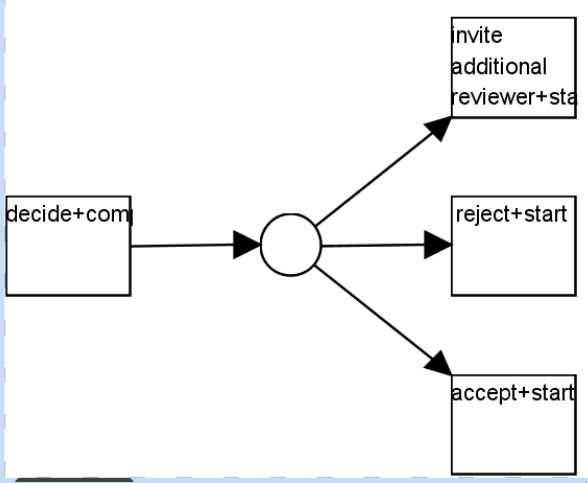


(X,Y)

Diagnostics per Passage



Problematic Passage



ProM UITopia

Result of replaying log CpnToolsLog on Petrinet from reviewing_with_fewer_errors_and_more_

Create new

invite reviewers+complete | invite additional reviewer+complete | collect reviews+start | get review 1+complete | accept+start | decide+complete

Passage graph | invite additional reviewer+start | invite reviewers+start

Petri net | Replay results

Log-model Alignments

Goto Case ID <type case id here>

Case id(s): 7

Num. Cases	12
Is Alignment Reliable?	Yes
Raw Fitness Cost	1

Alignment 1 events

Case id(s): 1

Num. Cases	8
Is Alignment Reliable?	Yes
Raw Fitness Cost	1

Alignment 1 events

Case id(s): 9

LEGEND

- Synchronous move (move log+model)
- Unobservable move (move model only)
- Skipped event class (move model only)
- Inserted event class (move log only)
- Move log+model with missing tokens

STATS FROM RELIABLE ALIGNMENTS

#Cases replayed	51
#Synchronous ev.class (log+model)	30
#Skipped ev.class	0
#Unobservable ev.class	0
#Inserted ev.class	49
#Violating synchronous ev.class	0
Raw Fitness Cost	0.98
Queued States	5.204...

ALIGNMENT STATISTICS

Raw Fitness Cost

0.96
3
0
0.63
34

51
30
0
0
49
0

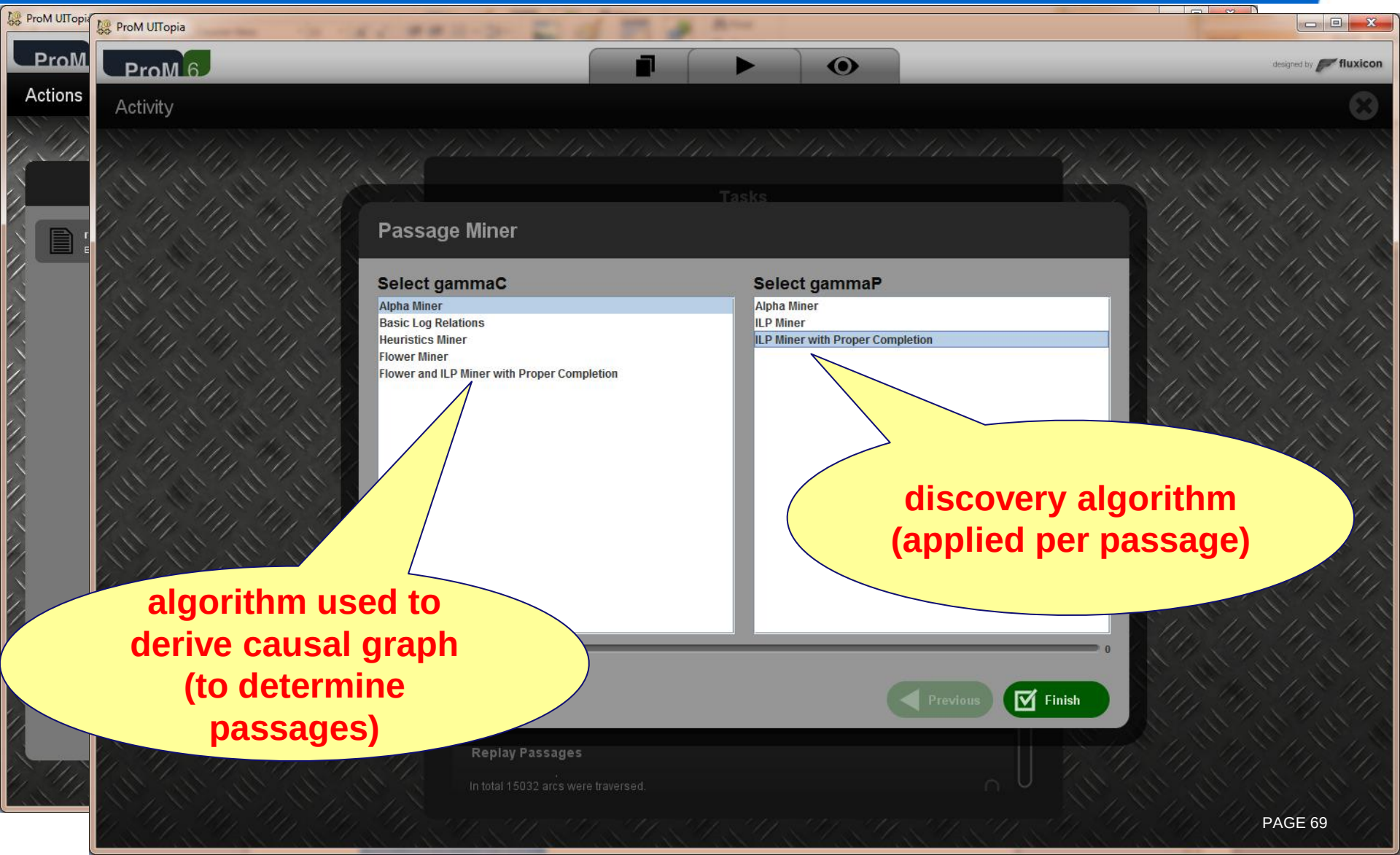
Case id(s): 7

Num. Cases	12
Is Alignment Reliable?	Yes
Raw Fitness Cost	1

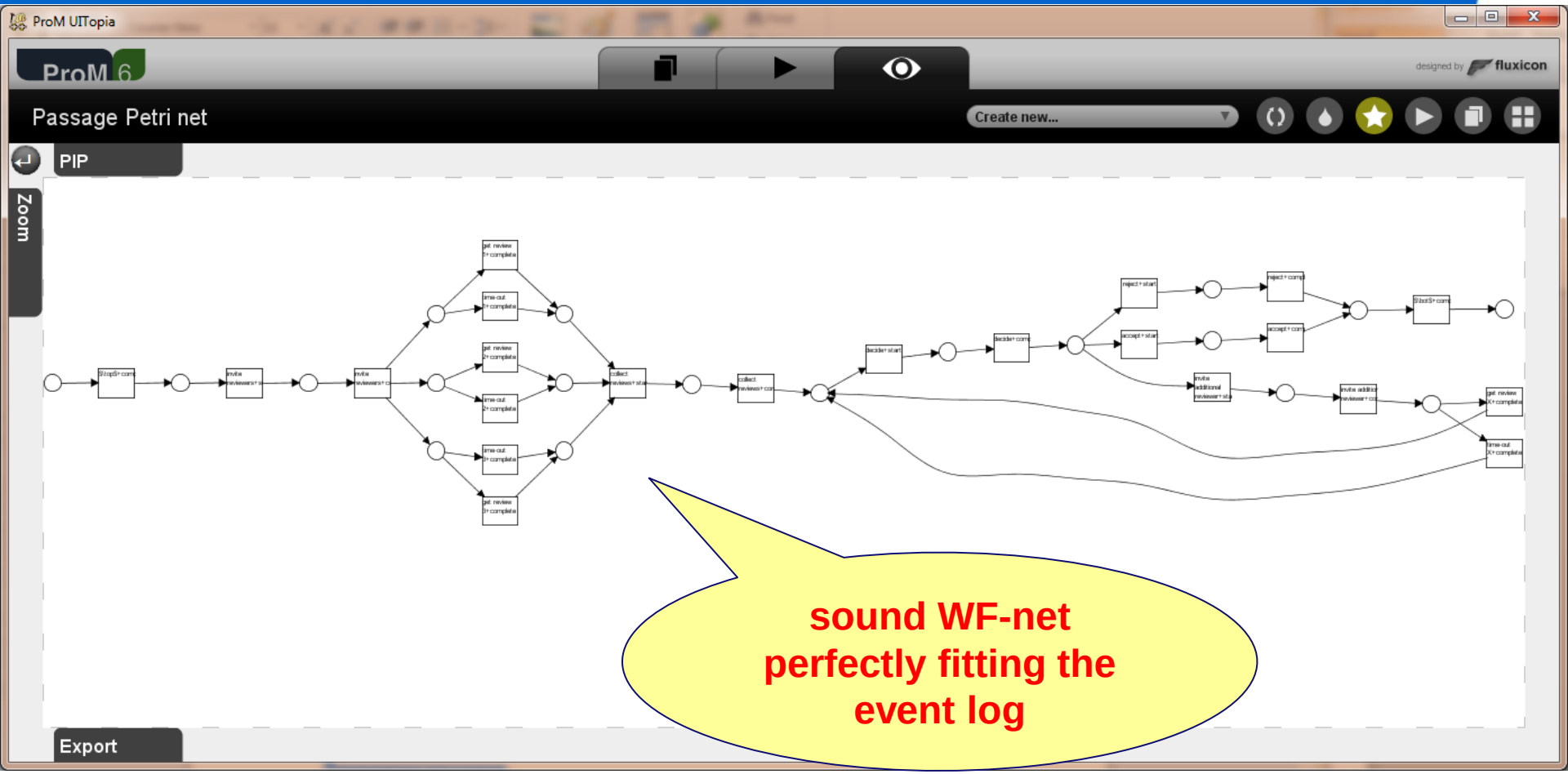
Alignment 1 events

decide+complete

Discovery (no initial model, just events)

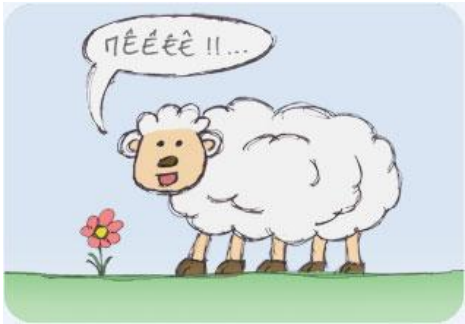


Result (alpha + ILP with proper completion)



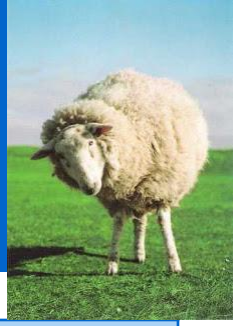
Conclusion

- **Process discovery and conformance checking can be decomposed!**
- **Generic approach with various formal guarantees independent of the techniques used.**
- **Open issues:**
  - **Experimental evaluation (in progress, see Eric's presentation)**
  - **Better algorithms for computing a causal structure are needed (independent of decomposition)**
  - **We are still missing sheep with five legs**




energy efficient
politically correct
1
formal (not just a picture)
2
fast (should not take years)
ability to balance all conformance dimensions (fitness, precision, generalization, and simplicity) incl. noise
3
4
sound (result should at least be free of deadlocks, etc.)
5
provide guarantees (not just a best effort)

Today's sheep have only 1.5-3.5 legs



	formal	fast	balance	sound	guaran.
alpha	+	+	-	-	-
heuristic	+	+	+/-	-	-
genetic AK	+	-	+/-	-	+/-
genetic JB	+	-	+	+	+/-
fuzzy miner	-	+	+/-	-	-
Disco	-	+	+/-	-	-
regions SB	+	-	-	+	+
regions LB	+	-	-	+	+
alpha ++	+	+	-	-	-
alpha #	+	+	-	-	-
SMT	+	-	-	+	+
MFM	+	+	-	-	+