Process Mining: Control-Flow Mining Algorithms

Ana Karla Alves de Medeiros

Eindhoven University of Technology
Department of Information Systems
a.k.medeiros@tue.nl

Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Process Mining

Event Logs

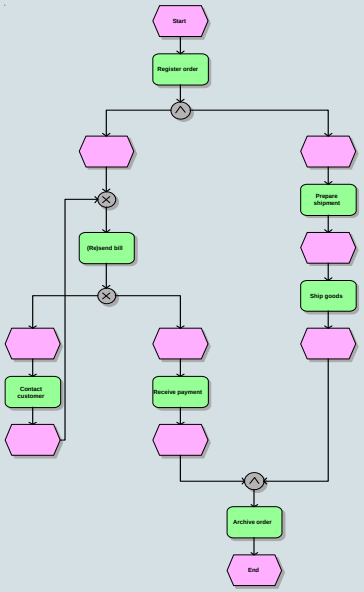
1. Register at hospital
2. Talk to doctor
3. Undergo surgery
4. Take medicines
5. Take blood sample
6. Talk to doctor
7. Go home

Process Mining

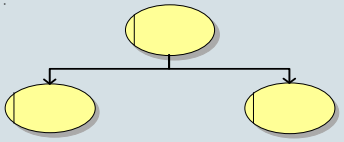
Event Log

Mining Techniques

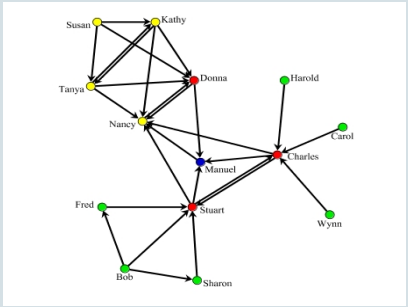
Process Model



Organizational Model



Social Network

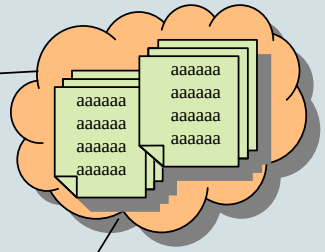


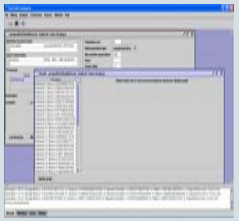
Performance Analysis

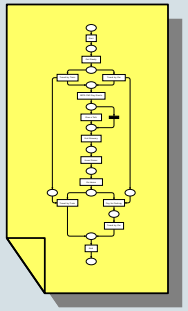
Auditing / Security

Mined Models

Event Logs are Everywhere!



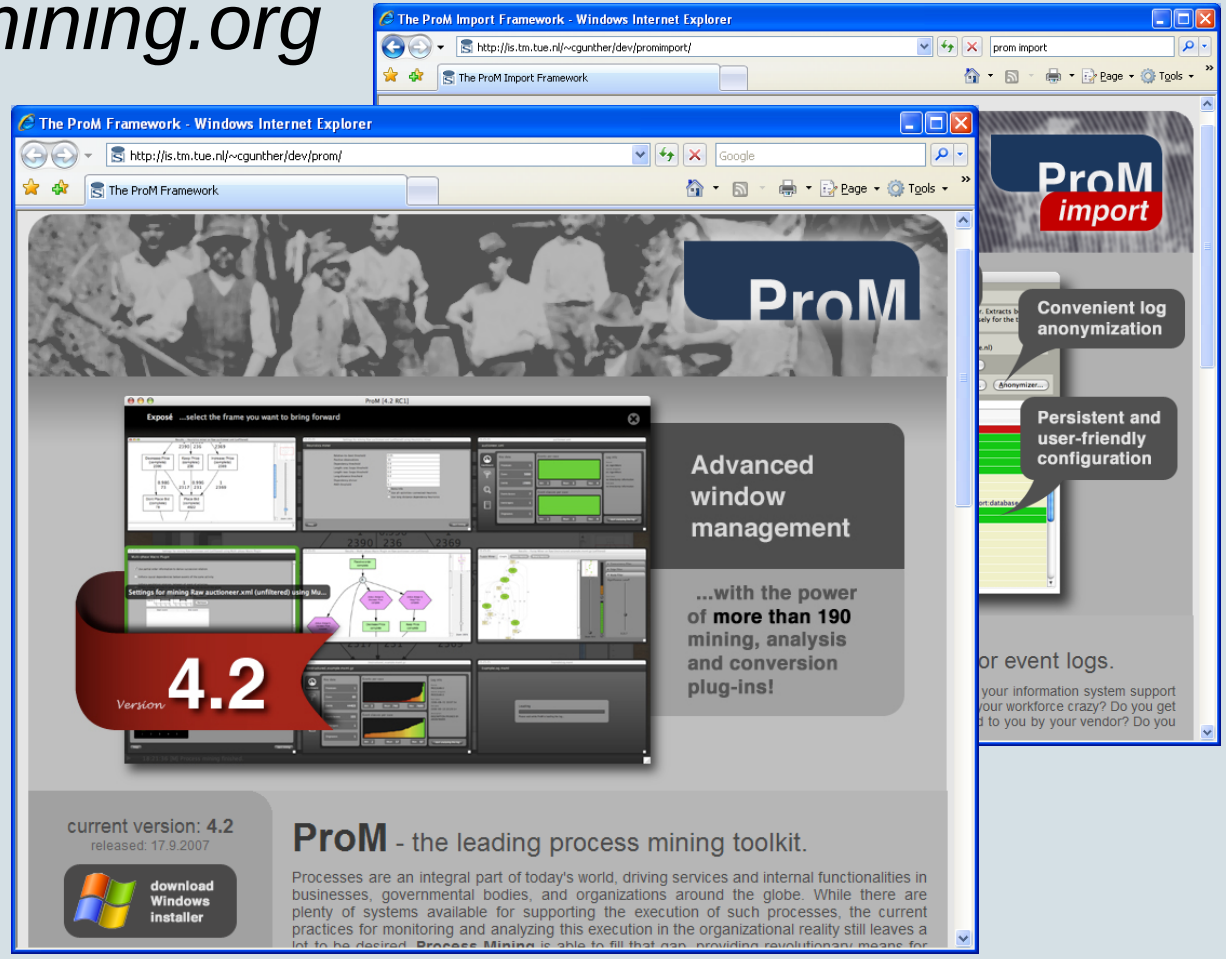




Machines, Municipalities, Airports, Internet, Hospitals, etc.

Tools

- *www.processmining.org*
- ProM 4.2
- ProM*import*
- Free tools!



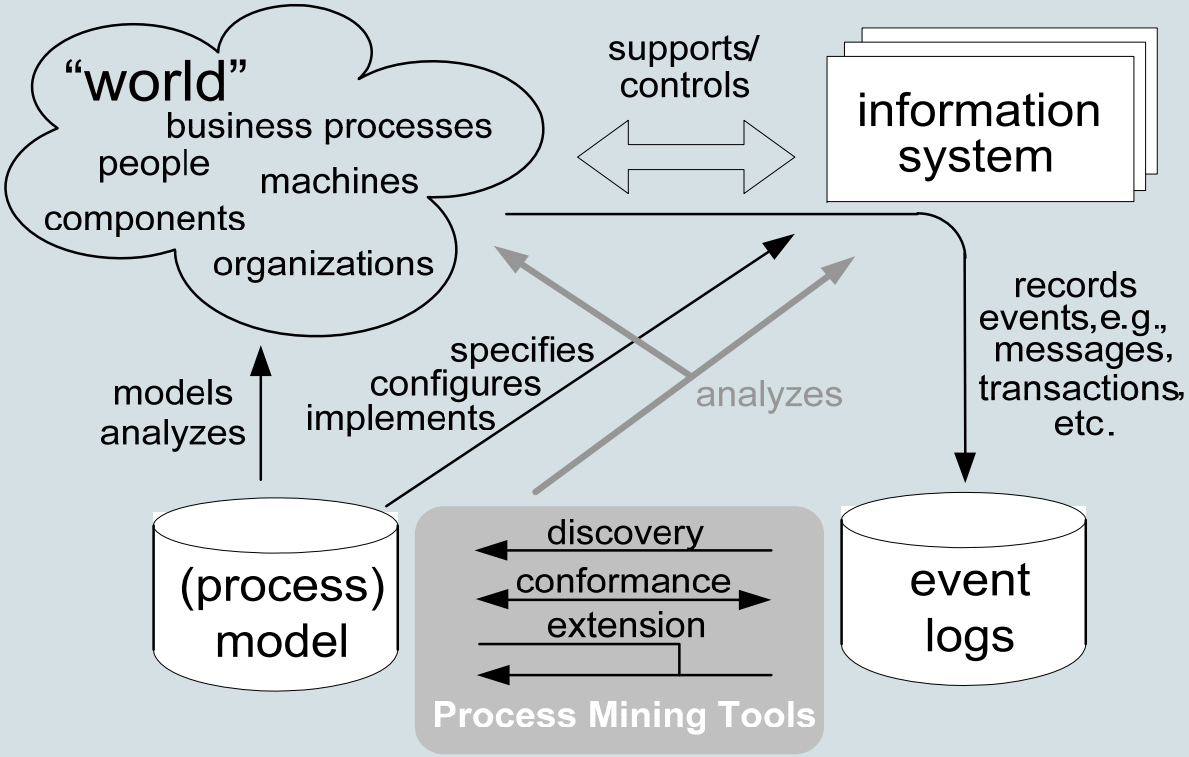
Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

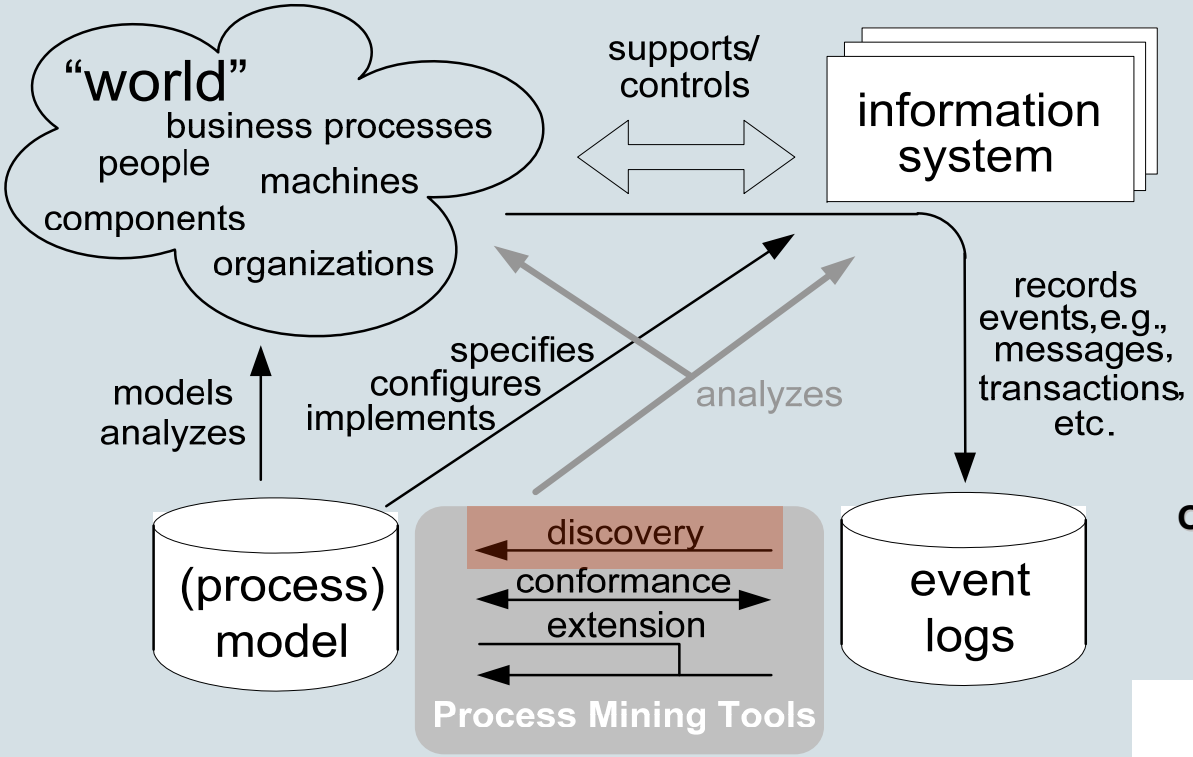
Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

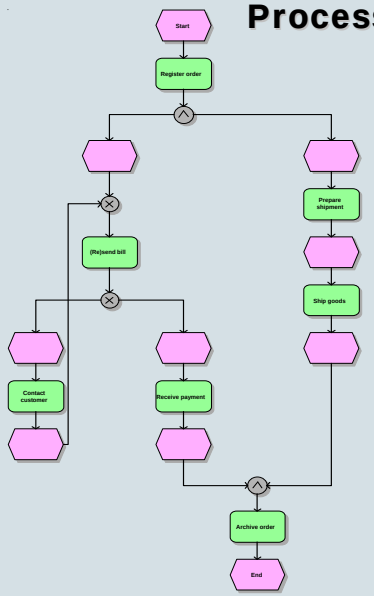
Types of Algorithms



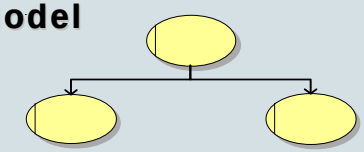
Types of Algorithms

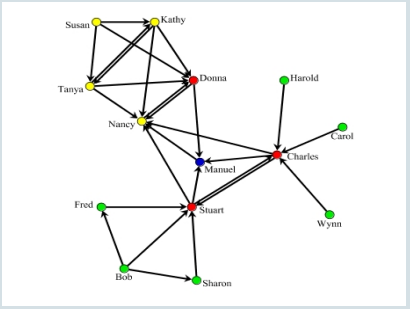


Process Model



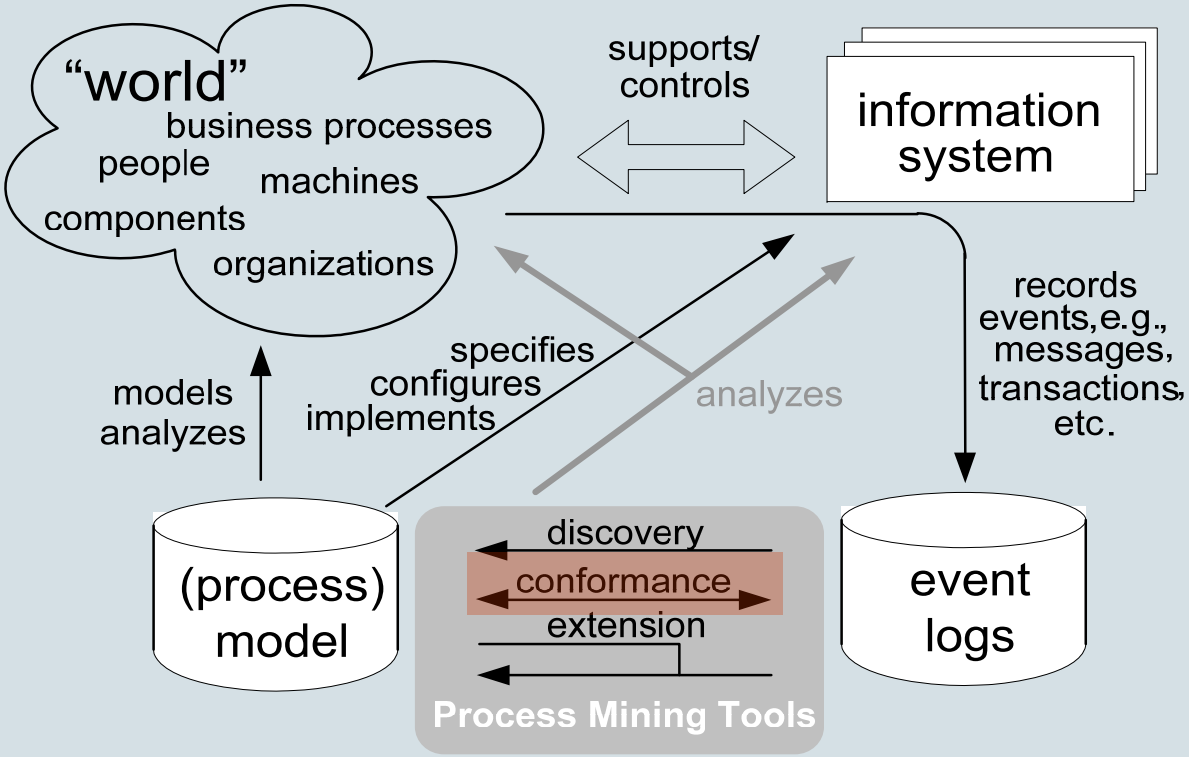
Organizational Model

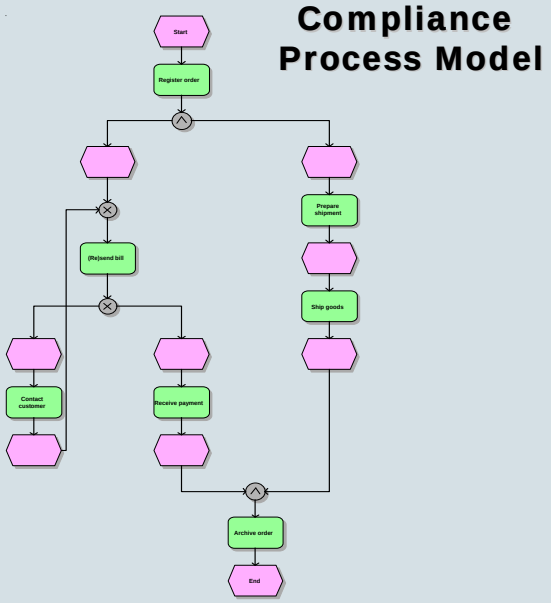




Social Network

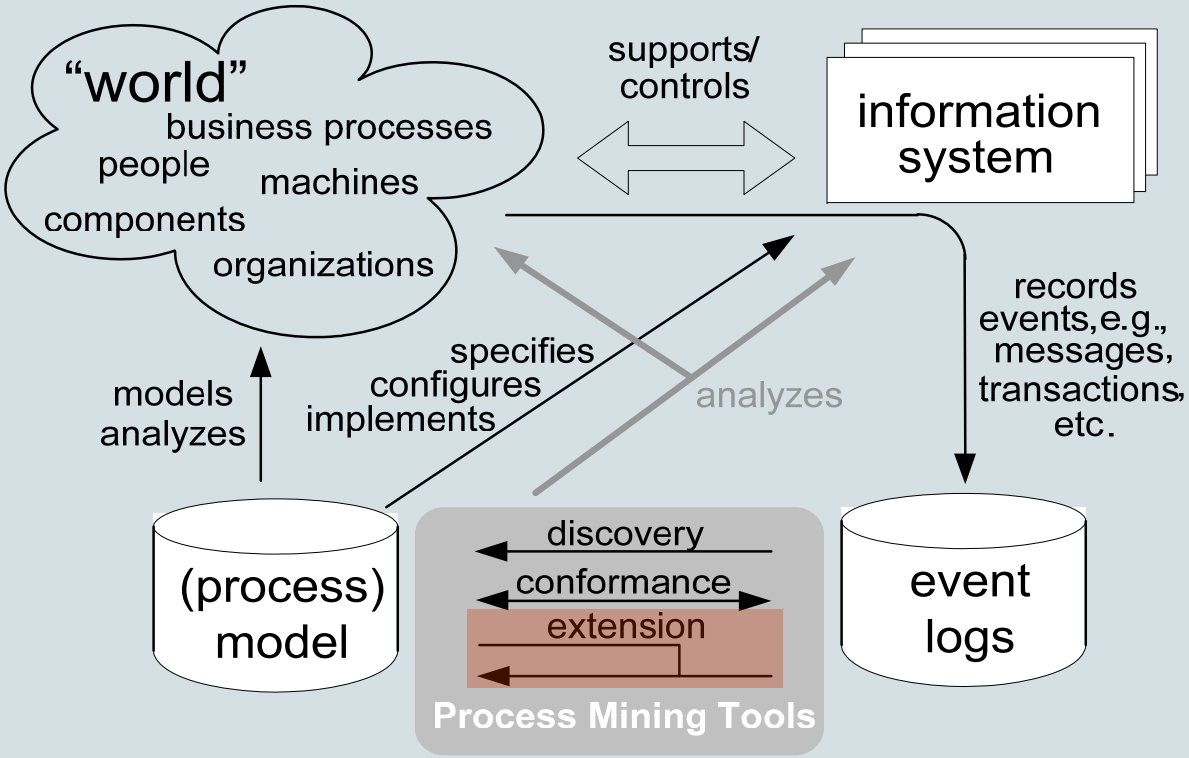
Types of Algorithms

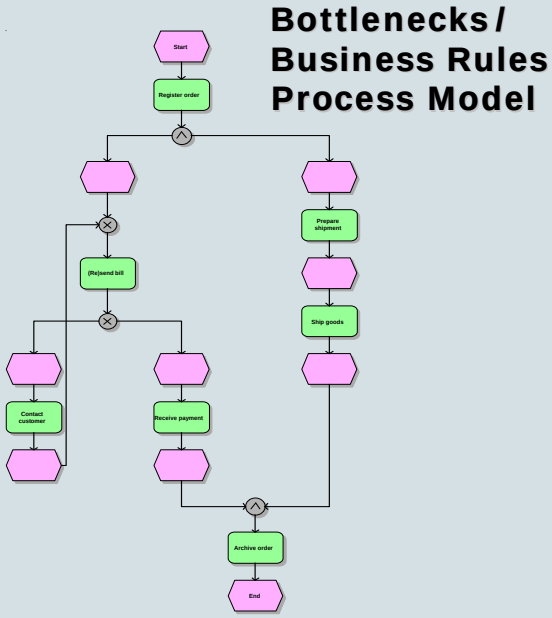




Auditing / Security

Types of Algorithms





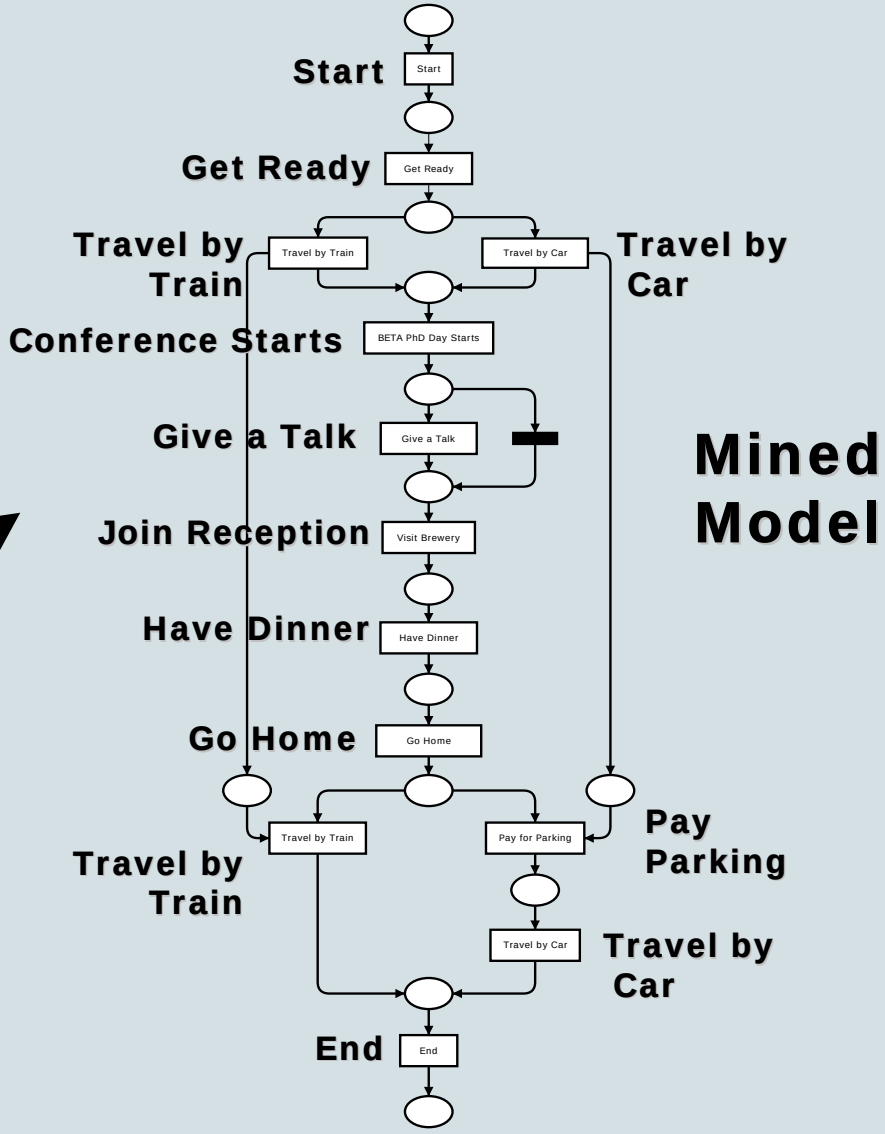
Performance Analysis

Process Mining

1. Start
2. Start
1. Start

1. Start
2. Get Ready
3. Travel by Car
4. Conference Starts
5. Join Reception
6. Have Dinner
7. Go Home
8. Pay Parking
9. Travel by Car
10. End

Event Log



Mined Model

Discovery Techniques:
Control-Flow Mining

Process Mining

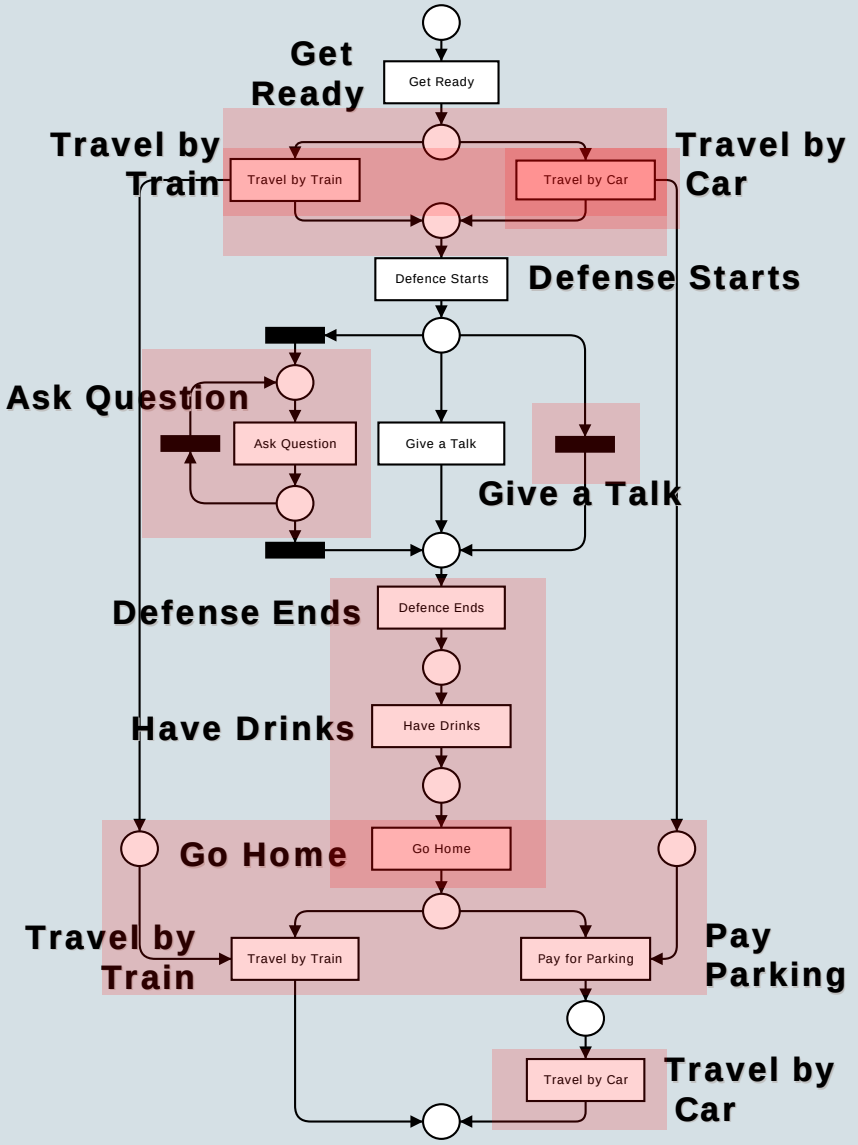
- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Common Constructs

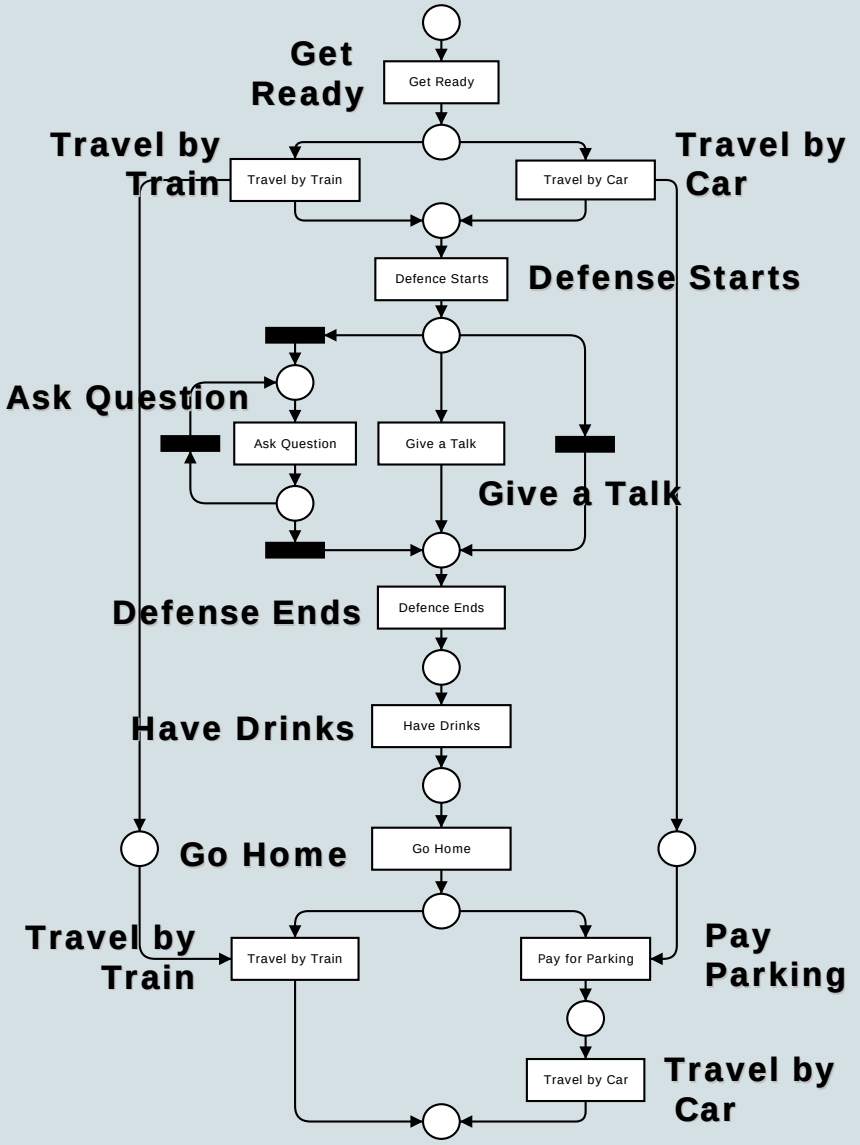
- Sequence
- Splits
- Joins
- Loops
- Non-Free Choice
- Invisible Tasks
- Duplicate Tasks



Common Constructs

- Sequence
- Splits
- Joins
- Loops
- Non-Free Choice
- Invisible Tasks
- Duplicate Tasks

+ noise in logs!



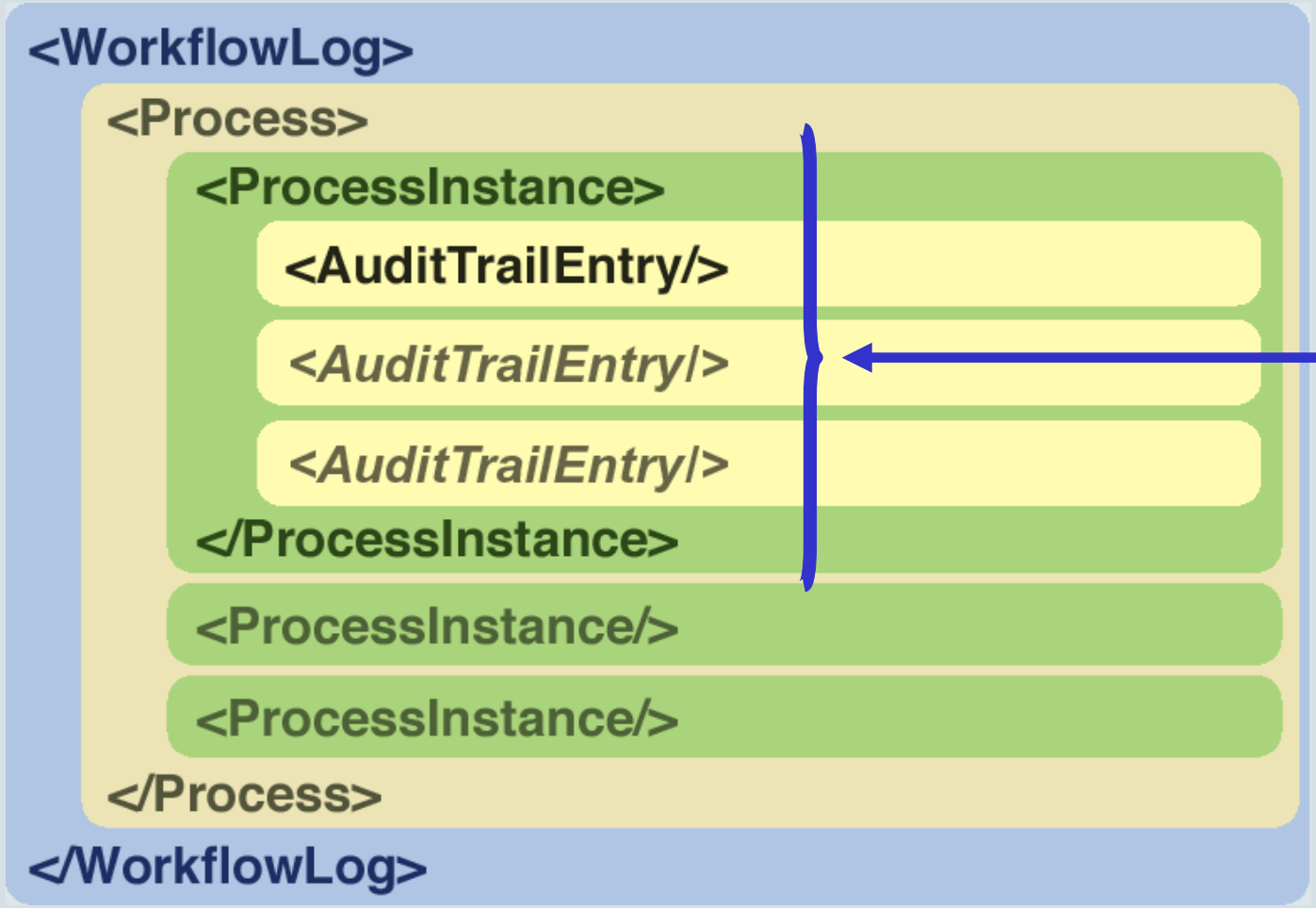
Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Process Mining

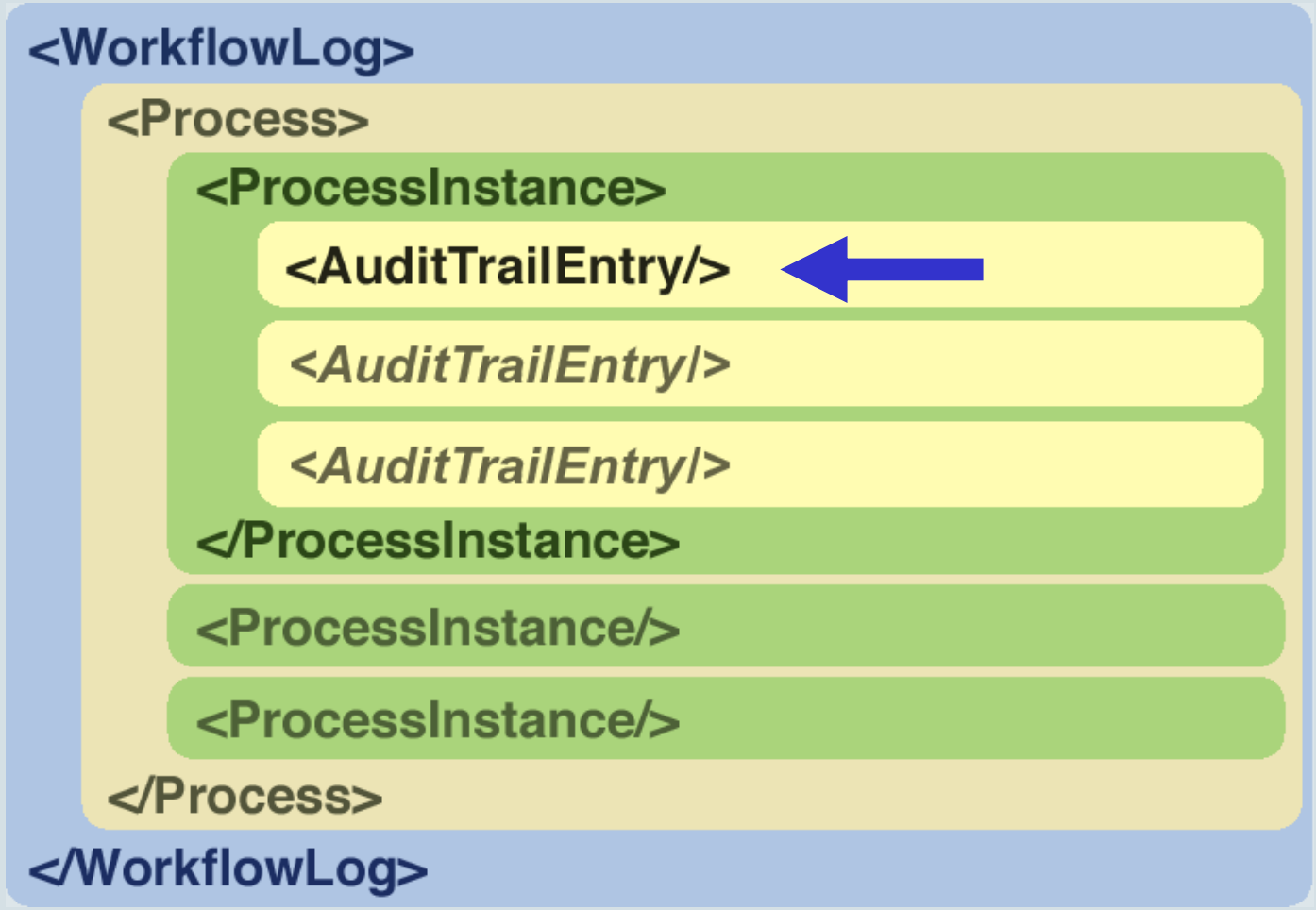
- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Event Log: Mining XML (MXML)

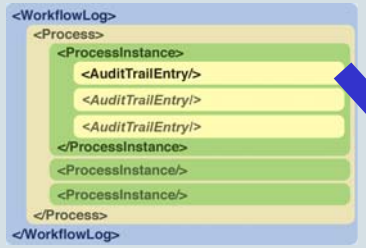


The notion of which tasks belong to a same instance is crucial for applying process mining techniques!

Event Log: Mining XML (MXML)



Event Log: Mining XML (MXML)



Compulsory fields!

<AuditTrailEntry>

<WorkflowModelElement/> *Task A* **</Wf.M.E.>**

<EventType> *complete* **</EventType>**

<TimeStamp> *2005-10-26T12:37:33...* **</TimeStamp>**

<Originator> *John Doe* **</Originator>**

<Data>

<Attribute name=”x”> *1* **</Attribute>**

<Attribute name=”y”> *whatever* **</Attribute>**

</Data>

</AuditTrailEntry>

Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Process Mining

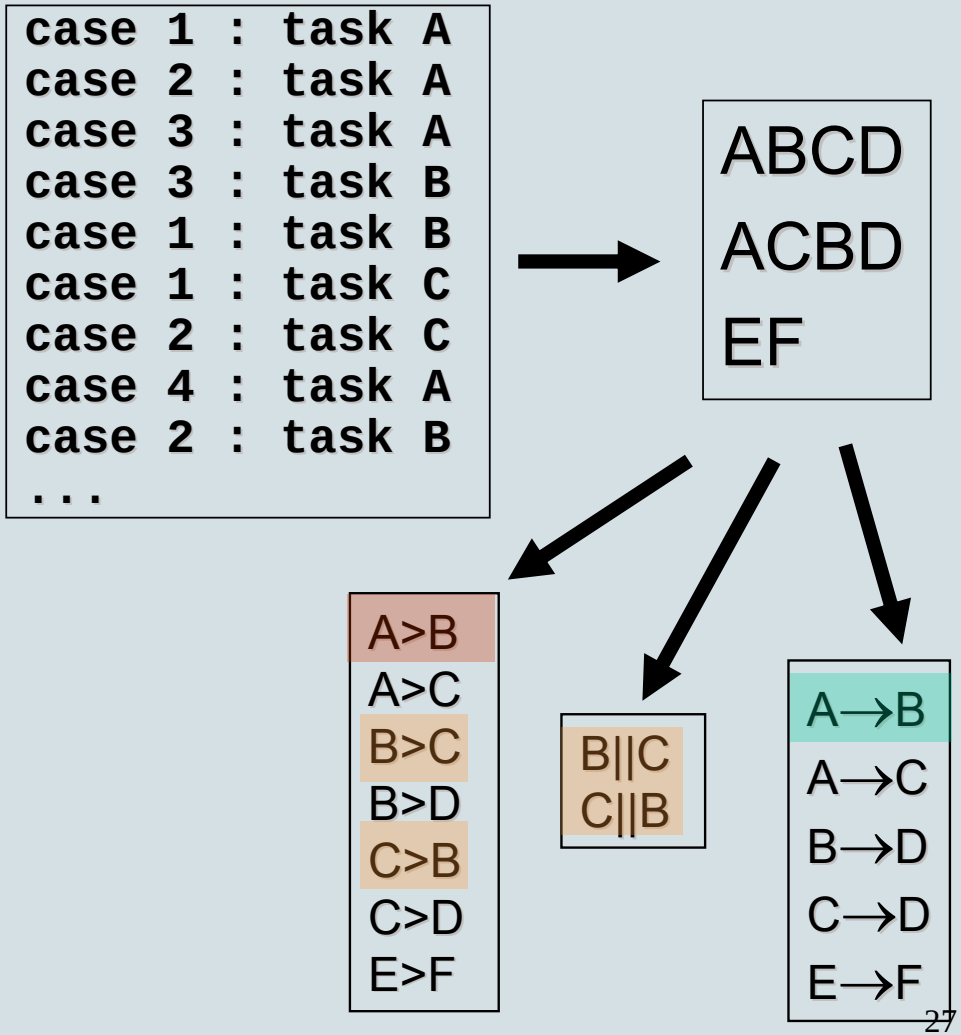
- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

α-algorithm

1. Read a log
2. Get the set of tasks
3. Infer the ordering relations ⟵ **Core Step!**
4. Build the net based on inferred relations
5. Output the net

α-algorithm - Ordering Relations >,→,||,#

- **Direct succession**: x>y iff for some case x is directly followed by y.
- **Causality**: x→y iff x>y and not y>x.
- **Parallel**: x||y iff x>y and y>x
- **Unrelated**: x#y iff not x>y and not y>x.

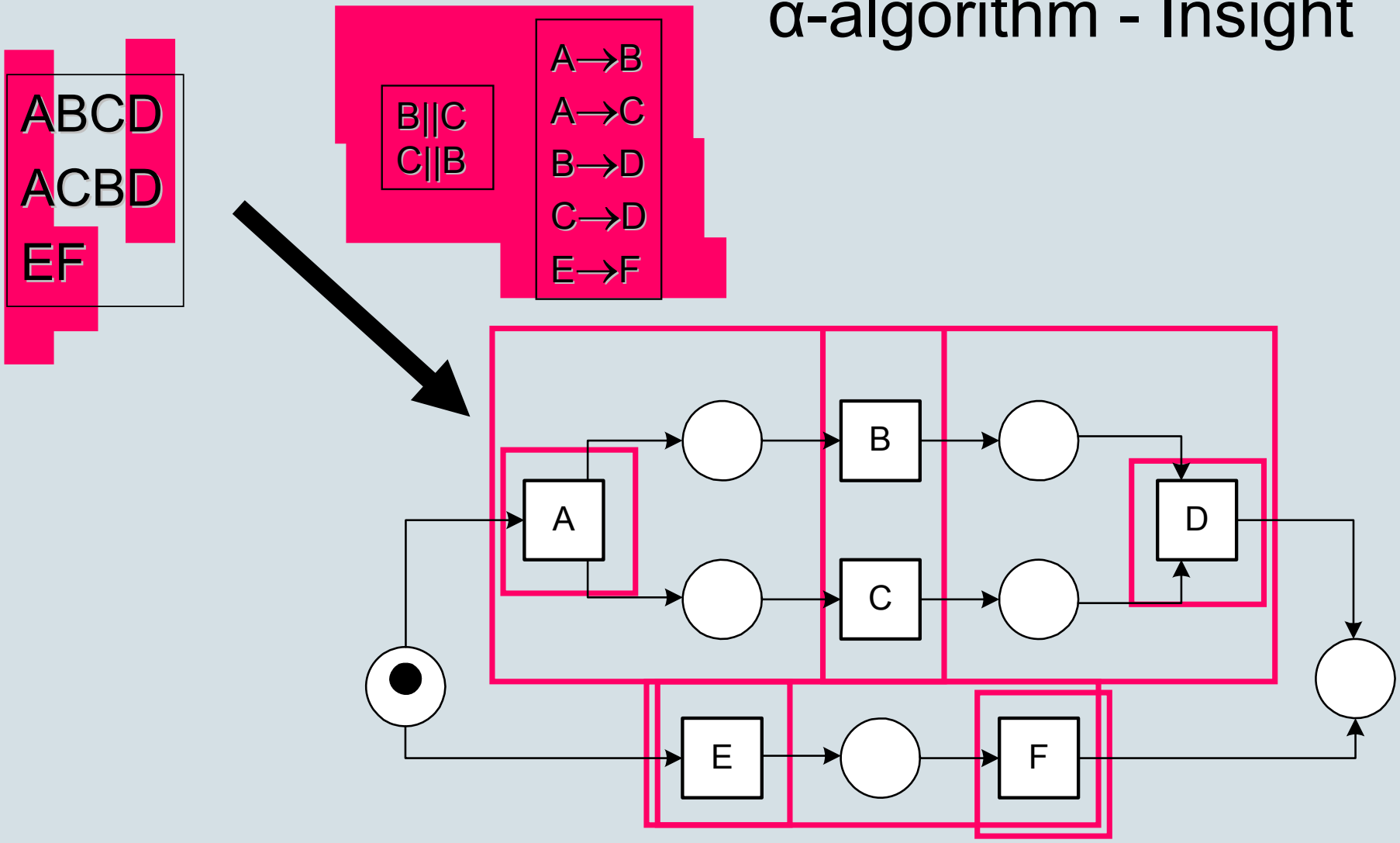


α-algorithm - Formalization

Let W be a workflow log over T. $\alpha(W)$ is defined as follows.

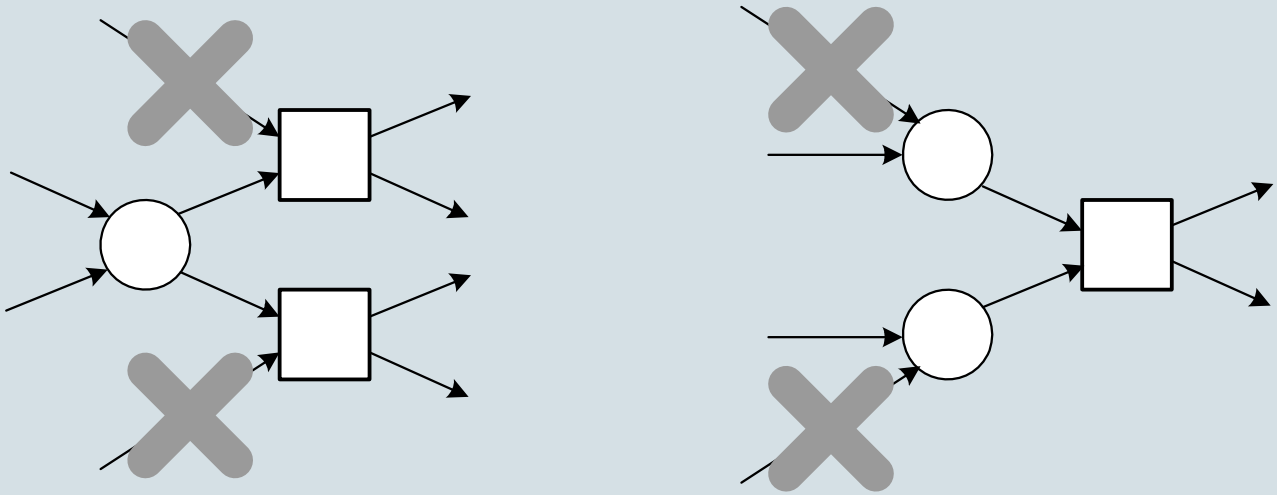
1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{last}(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\, a_1 \#_W a_2 \wedge \forall_{b1,b2 \in B}\, b_1 \#_W b_2 \}$,
5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$,
7. $F_W = \{ (a,p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)},b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_W,t) \mid t \in T_I \} \cup \{ (t,o_W) \mid t \in T_O \}$, and
8. $\alpha(W) = (P_W, T_W, F_W)$.

α-algorithm - Insight



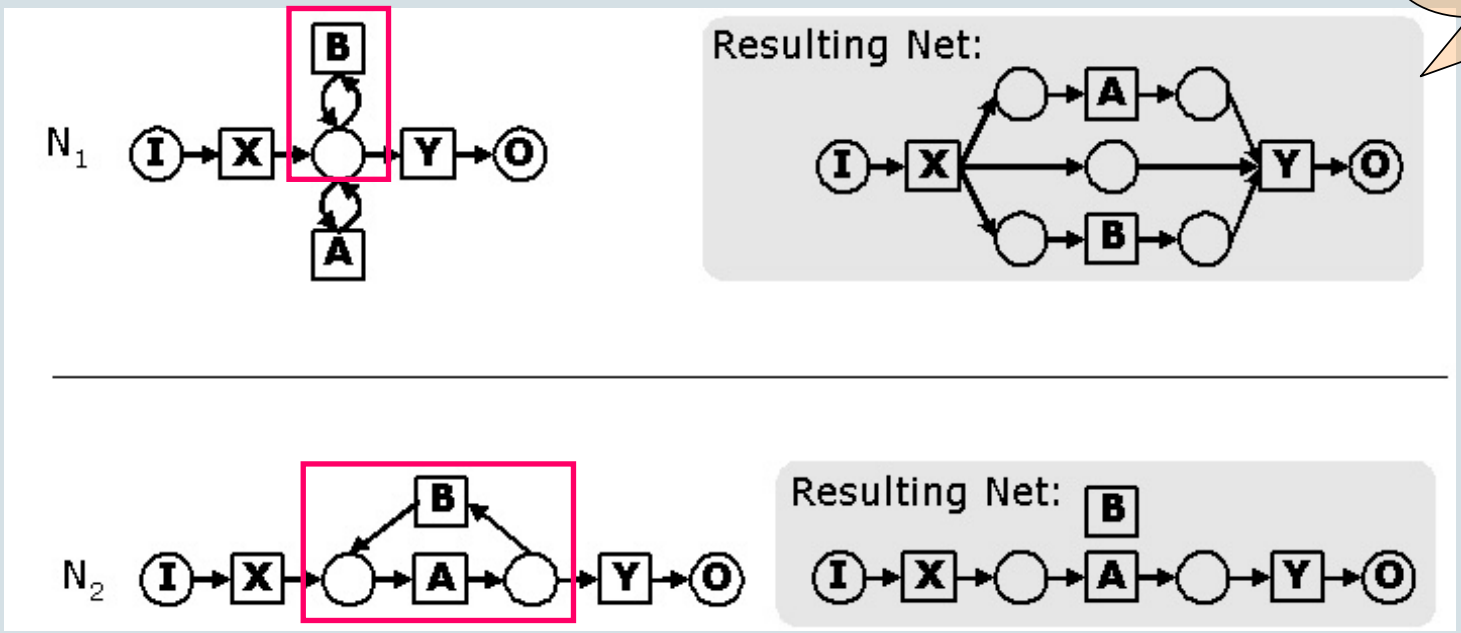
α-algorithm – Log properties + target nets

- If log is complete with respect to relation >, it can be used to mine SWF-net without short loops
- *Structured Workflow Nets* (SWF-nets) have no implicit places and the following two constructs cannot be used:



α-algorithm – No short loops

B>B and not B>B implies B→B (impossible!)



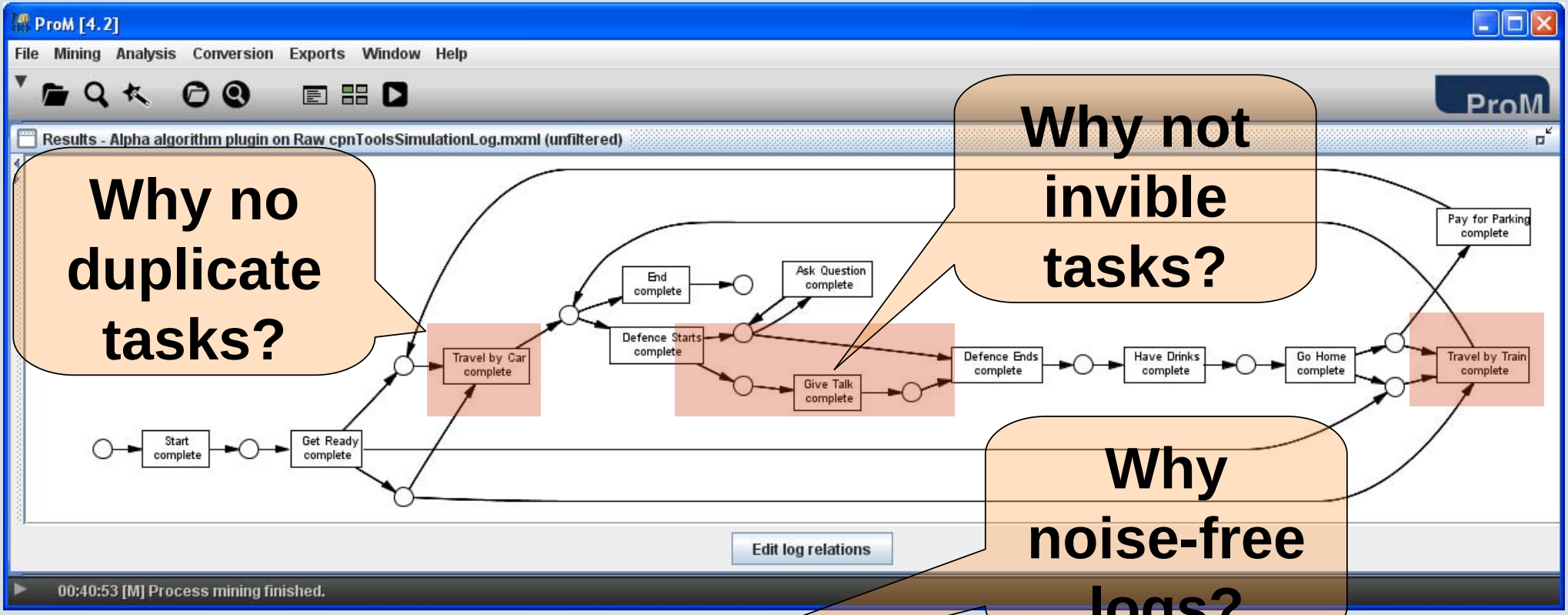
Why no short loops?

One-length

Two-length

A>B and B>A implies A||B and B||A instead of A→B and B→A

α-algorithm – Common Constructs



- No invisible tasks, non-free-choice or duplicate tasks
- No noisy logs

Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Algorithm
- Fuzzy Miner

Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Algorithm
- Fuzzy Miner

Heuristics Miner

1. Read a log
2. Get the set of tasks
3. Infer the ordering relations **based on their frequencies**
4. Build the net based on inferred relations
5. Output the net

Heuristics Miner

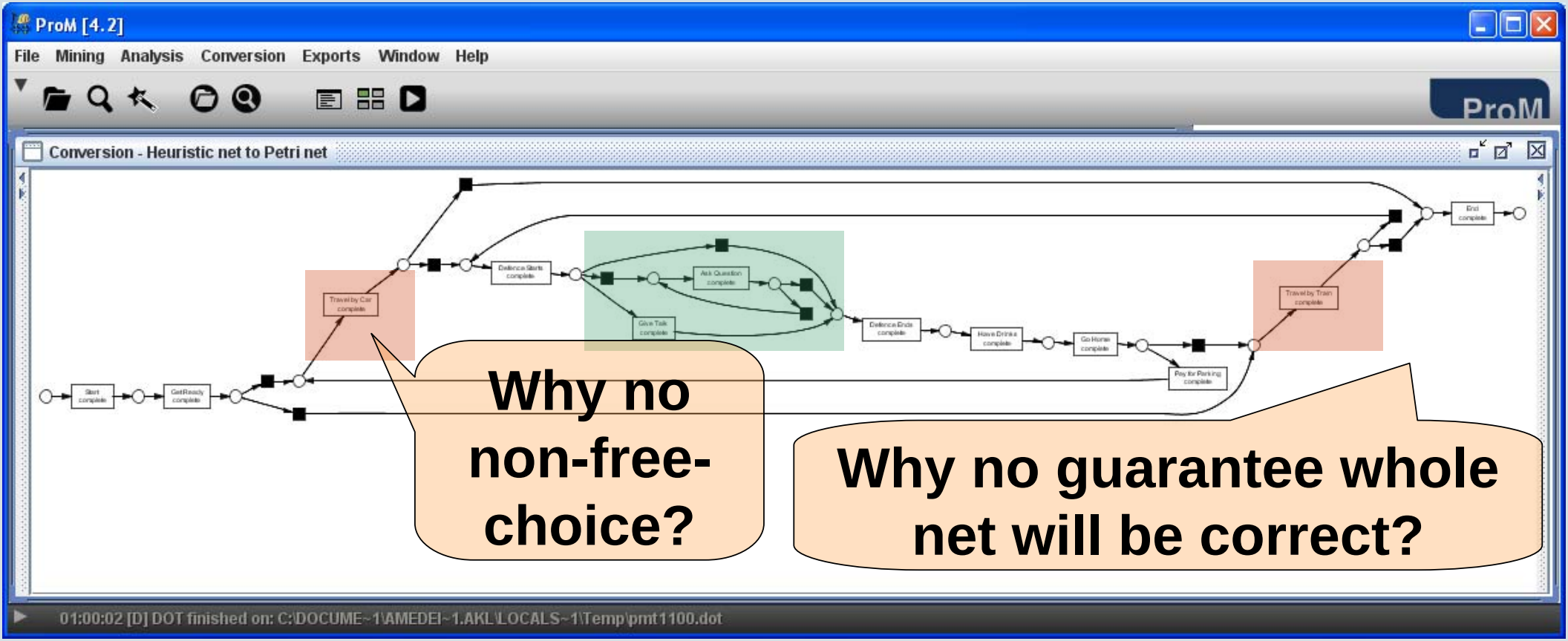
Let W be an event log over T, and $a, b \in T$:

- $|a >_W b|$ is the number of times $a >_W b$ occurs in W,
- $a \Rightarrow_W b = \left(\frac{|a >_W b| - |b >_W a|}{|a >_W b| + |b >_W a| + 1} \right)$

Insight:

The more frequently a task A directly follows another task B, and the less frequently the opposite occurs, the higher the probability that A causally follows B!

α-algorithm – Common Constructs



- No non-free-choice or duplicate tasks
- Robust to invisible tasks and noisy logs

Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Process Mining

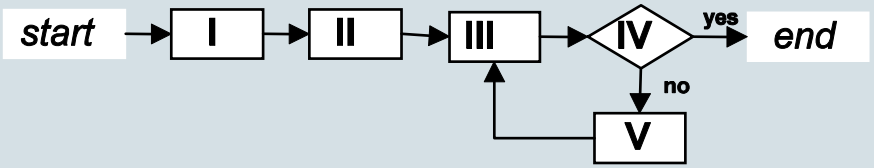
- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Genetic Process Mining (GPM)

- Genetic Algorithms + Process Mining
- Genetic Algorithms
  - Search technique that mimics the process of evolution in biological systems
- Advantages
  - Tackle all common structural constructs
  - Robust to noise
- Disadvantages
  - Computational Time

Genetic Process Mining (GPM)

Algorithm:



Internal Representation

Fitness Measure

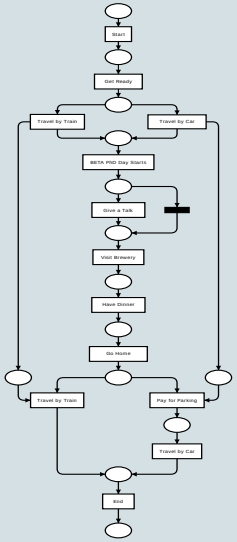
Genetic Operators

Step	Description
I	*Read event log*
II	*Build the initial population* ●
III	*Calculate fitness of the individuals in the population* ● ●
IV	*Stop and return the fittest individuals?* ●
V	*Create next population – use elitism and genetic operators* ● ●

GPM – Fitness Measure

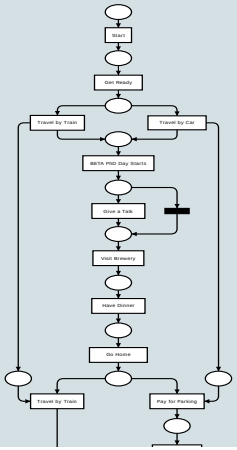
- Guides the search!

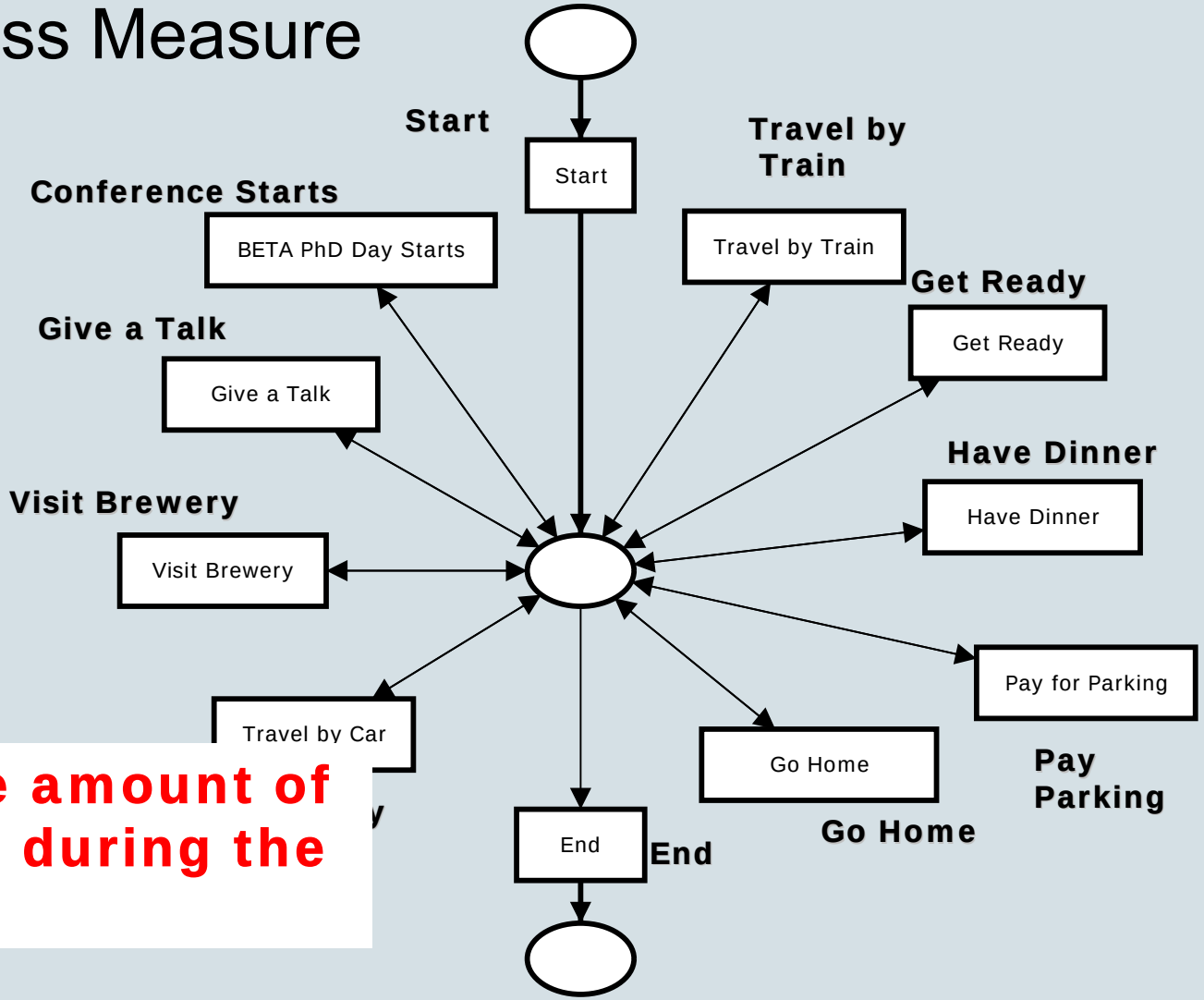
GPM – Fitness Measure



GPM – Fitness Measure

Overgeneral solution

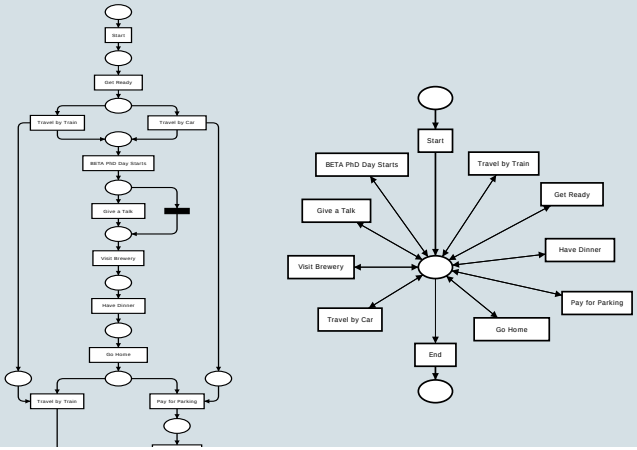




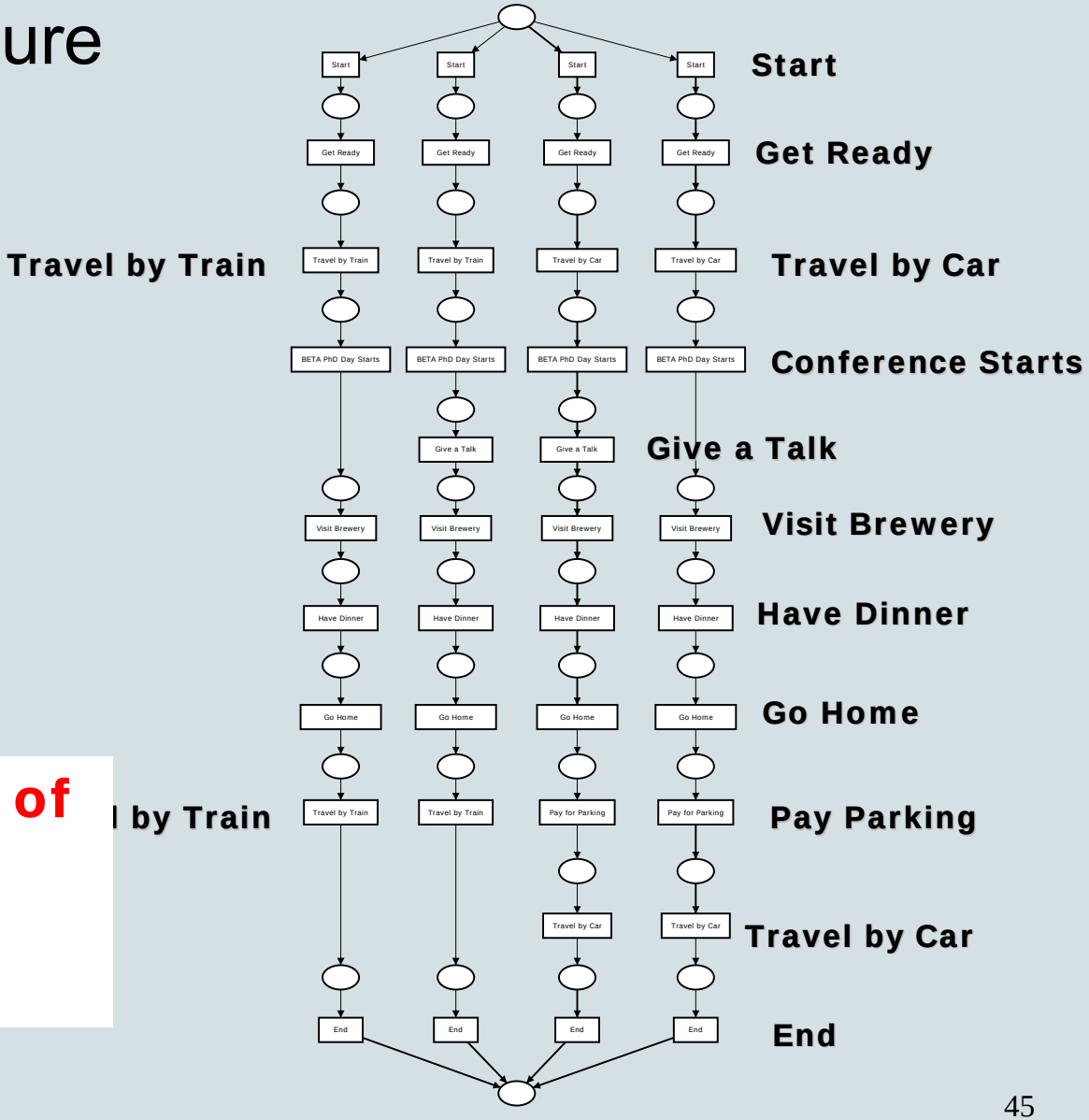
Punish for the amount of enabled tasks during the parsing!

GPM – Fitness Measure

Overspecific solution



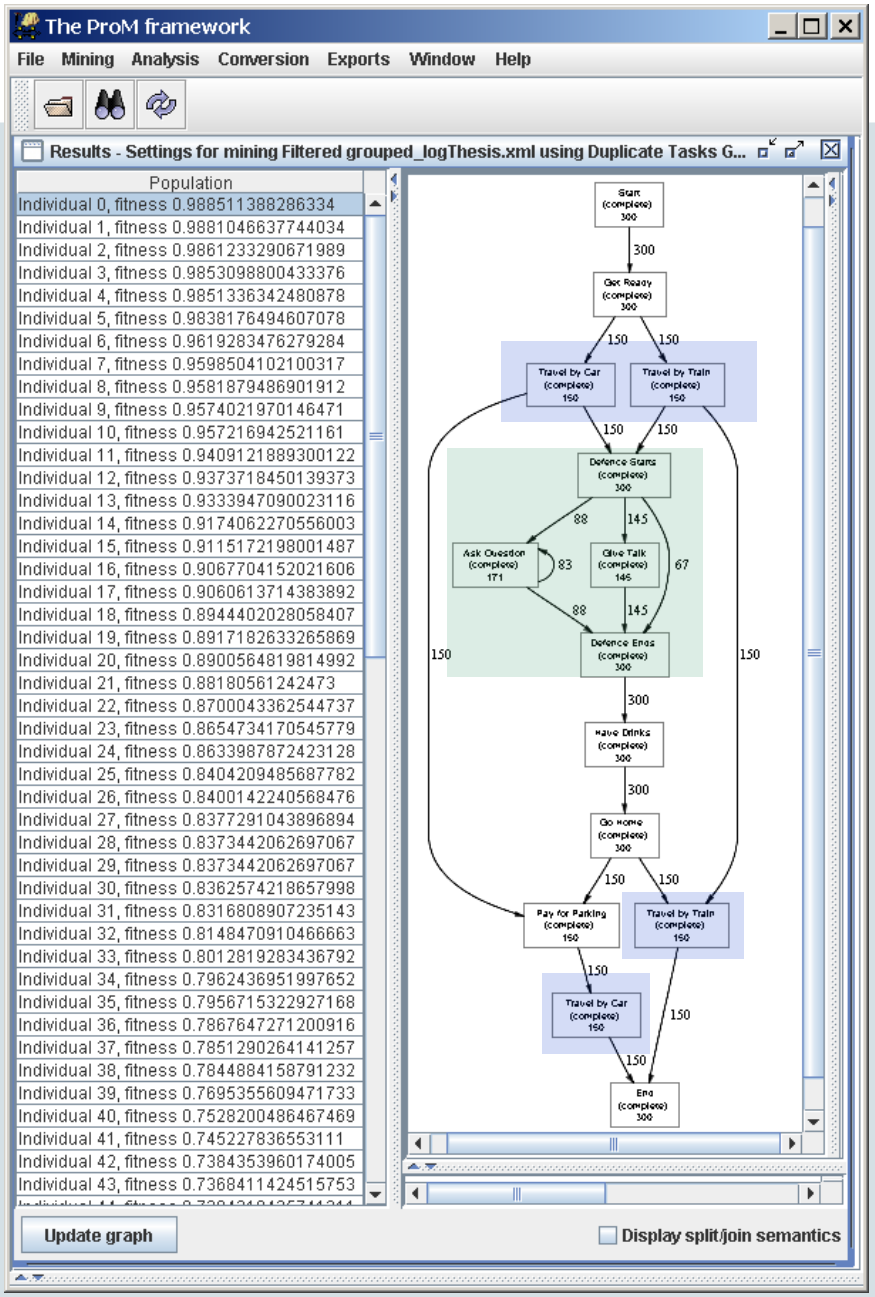
Punish for the amount of duplicate tasks with common input/output tasks!



GPM – DGA ProM Plug-in

Why does the GA Miner takes so much time?

How could we improve its running time without changing the code?



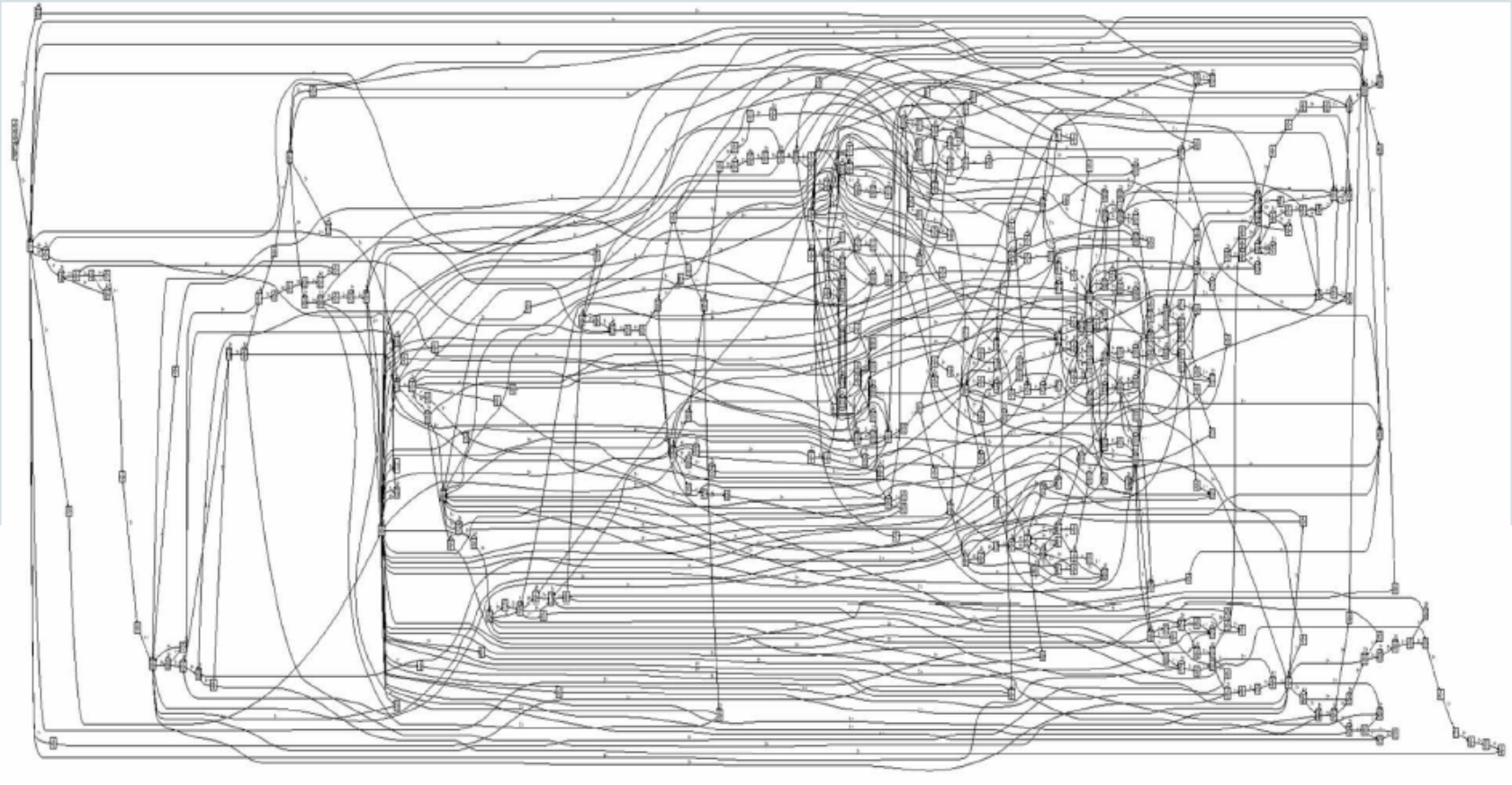
Process Mining

- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Process Mining

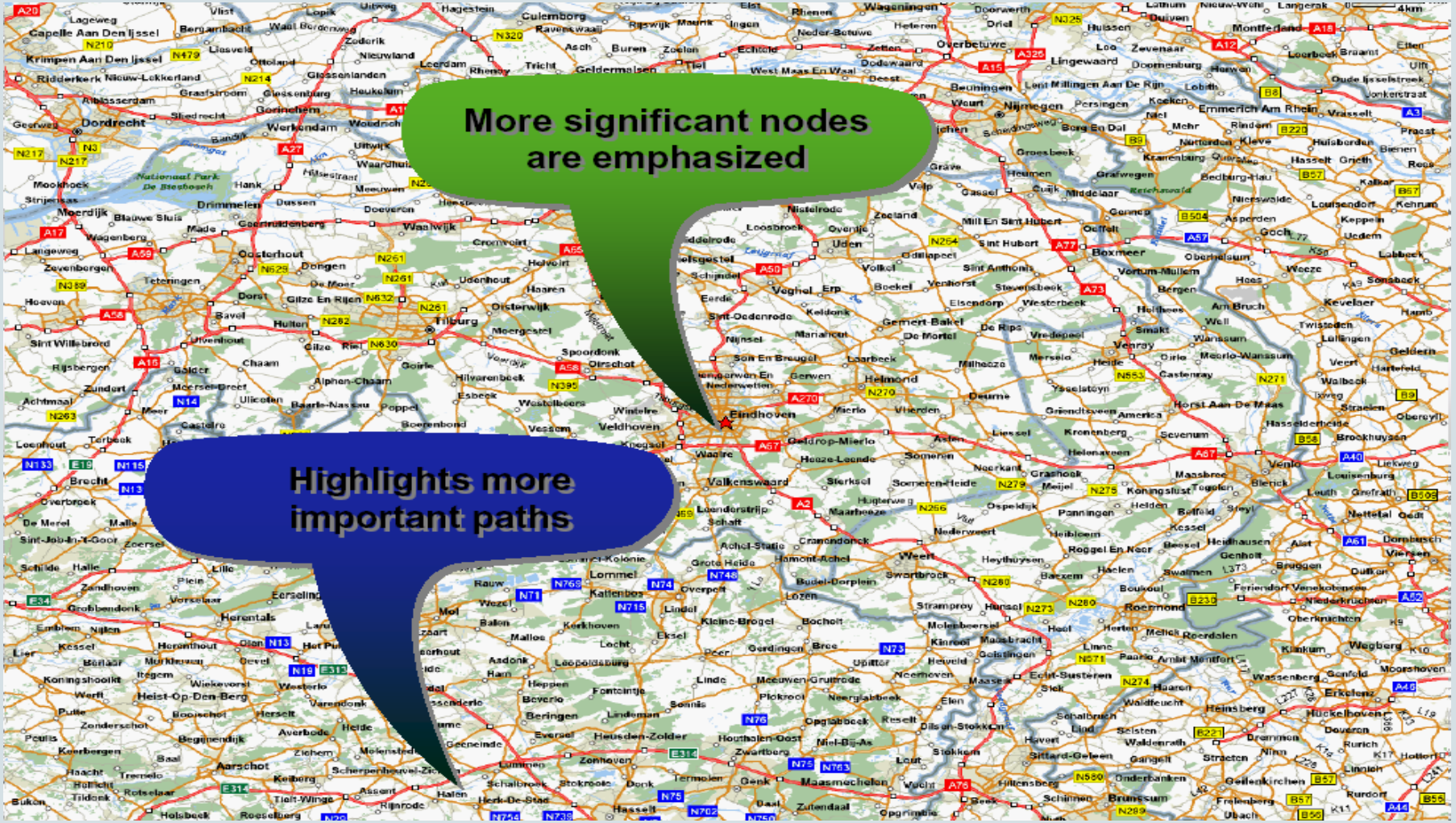
- Short Recap
- Types of Process Mining Algorithms
- Common Constructs
- Input Format
- α-algorithm
- Heuristics Miner
- Genetic Miner
- Fuzzy Miner

Fuzzy Miner - Motivation



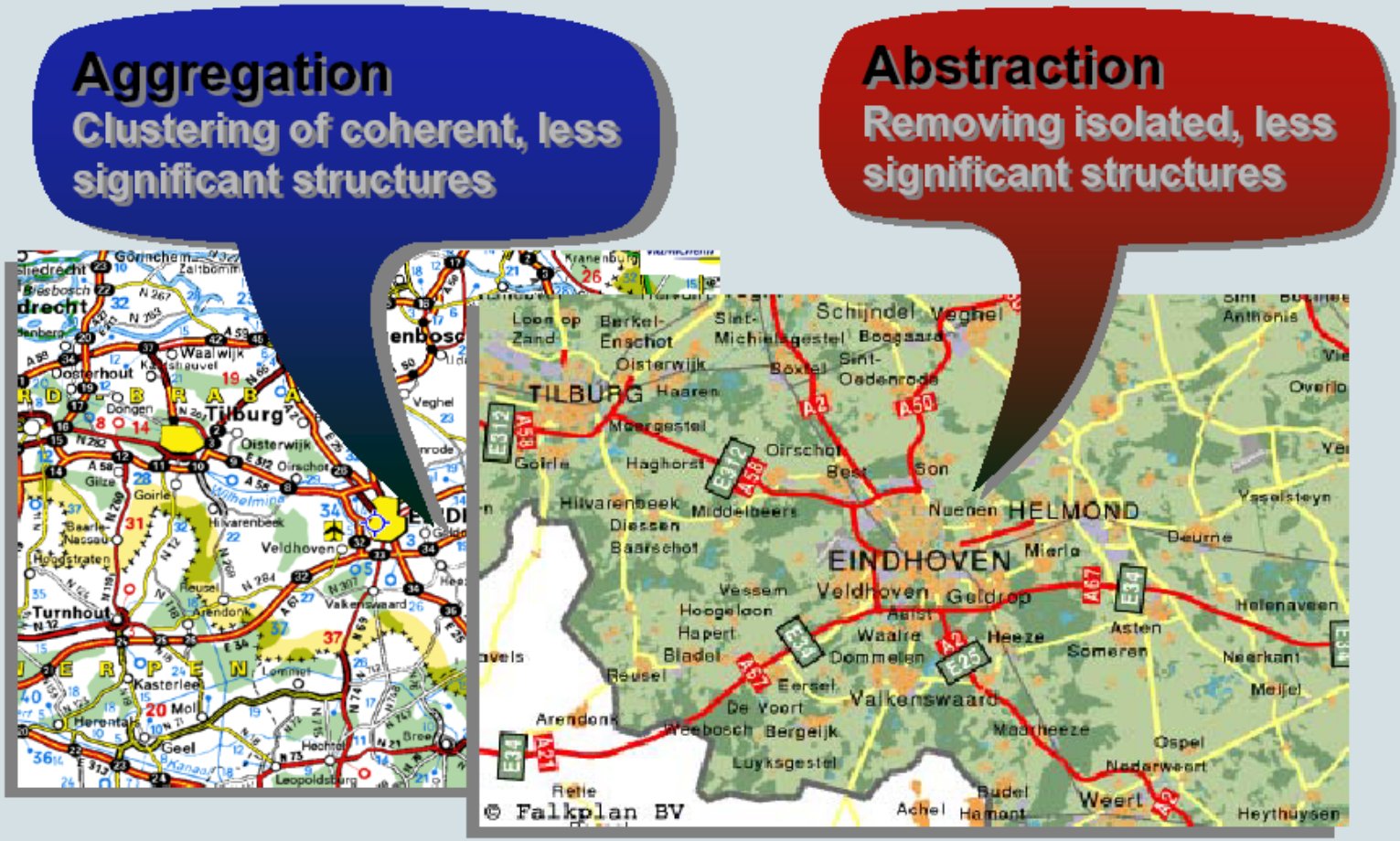
Mine less structured processes!

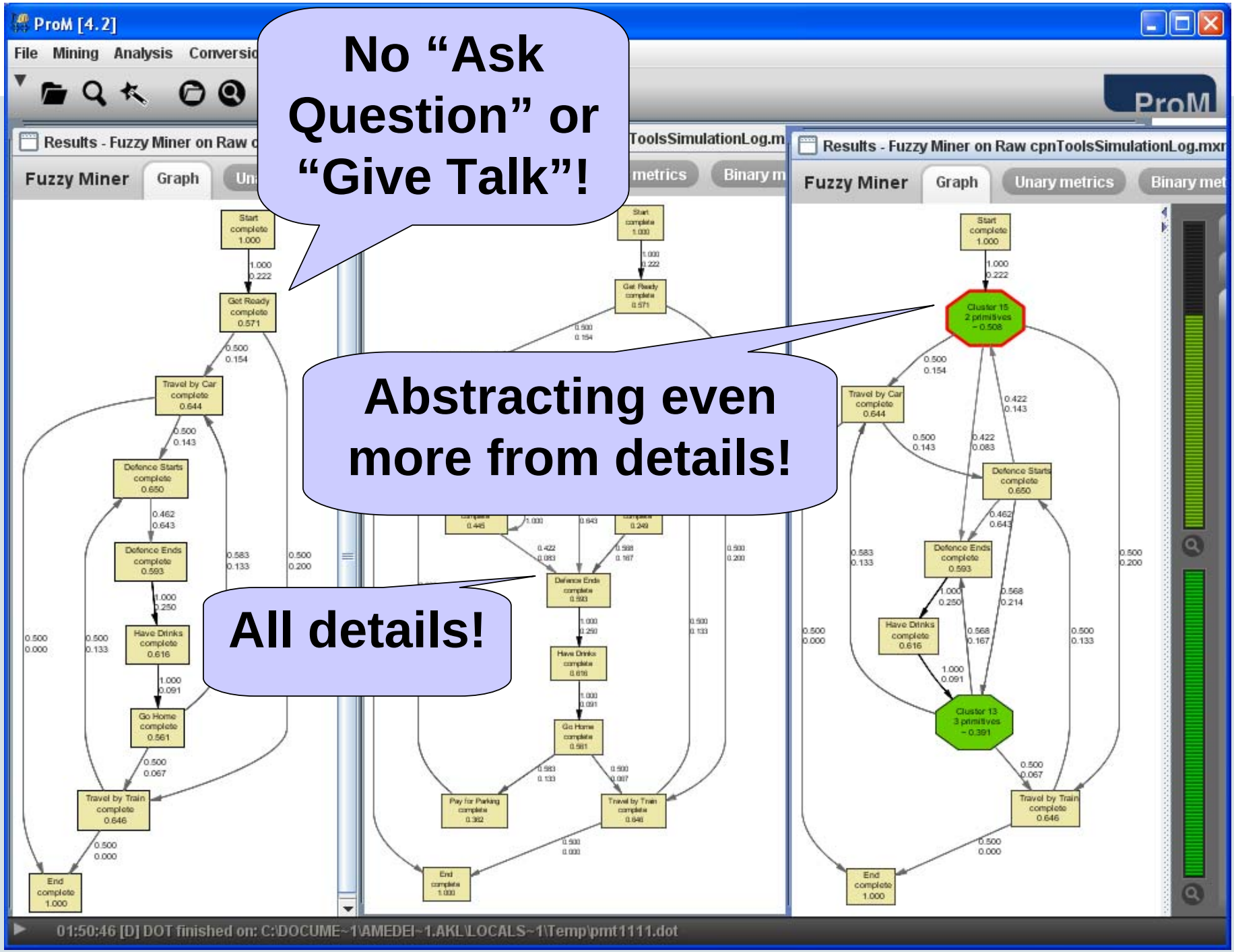
Fuzzy Miner - Motivation



Fuzzy Miner

More to learn from maps...




No “Ask Question” or “Give Talk”!
Abstracting even more from details!
All details!
ProM [4.2]
File Mining Analysis
ProM
Results - Fuzzy Miner on Raw
Fuzzy Miner
Graph
Results - Fuzzy Miner on Raw cpnToolsSimulationLog.mxr
Fuzzy Miner
Graph
Unary metrics
Binary met
01:50:46 [D] DOT finished on: C:\DOCUME~1\AMEDEI~1.AKL\LOCALS~1\Temp\pmt1111.dot

Conclusions

- The notion of a process instance is crucial!
- Ordering of tasks is the basic information
- Frequencies are important to handle noise
- Local approaches
  - α-algorithm, Heuristics Miner
- Global approaches
  - Genetic Miner and Fuzzy Miner