Model Quality, Alpha Miner & Process Exploration

Benedikt Knopp

BPI-Instruction 4

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Today‘s Agenda

Model Quality Assessment

Alpha Miner: Step-by-Step Example

Alpha Miner: Many Examples

Process Exploration (ProM, Celonis)

Chair of Process and Data Science

Recap: Model Quality Assessment

Fitness (Recall):
To what extend can the behavior recorded in the log be replayed by the model?

Precision:
To what extend is the model behavior present in the log?

Generalization:
How likely is it that an unseen trace can be replayed by the model?

Hard / Ambiguous answers.
Requires knowledge or assumptions about the process apart from what is seen in the log.

Simplicity:
How simple and readable is the model?

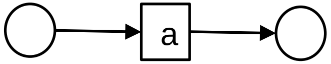
E.g. number of nodes, no label duplications…
Let's keep the discussion informal here

Recap: Model Quality Assessment

Generalization:
How likely is it that an unseen trace can be replayed by the model?

Log-based assessment: Gain confidence from observing log frequencies.

Non-Log based assessment: Make assumptions about the process apart from what is seen in the log.



$L_1 = [\langle a \rangle^1]$
$L_2 = [\langle a \rangle^{100}]$

For this model, generalization w.r.t L_2 is higher than w.r.t L_1 because we have more evidence that future traces look the same.

See next slides…

Chair of Process and Data Science

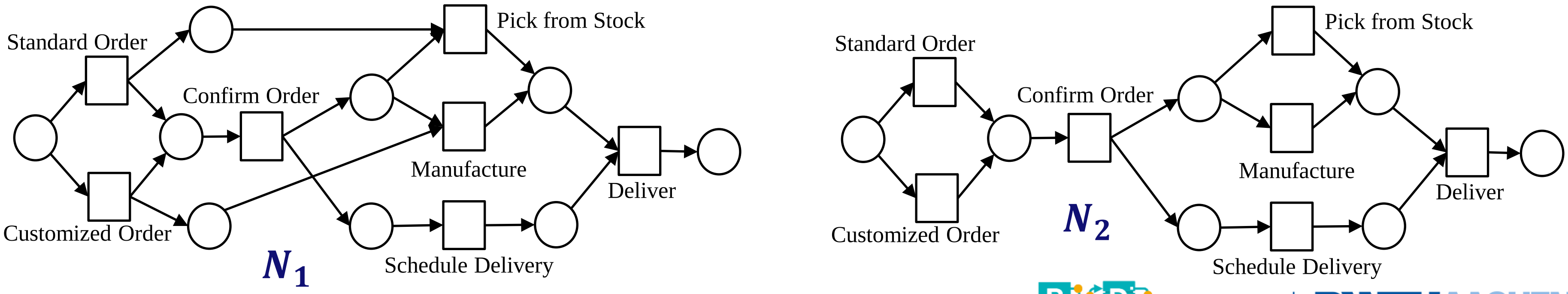
RWTH AACHEN UNIVERSITY

Exercise 1a: Model Quality Assessment

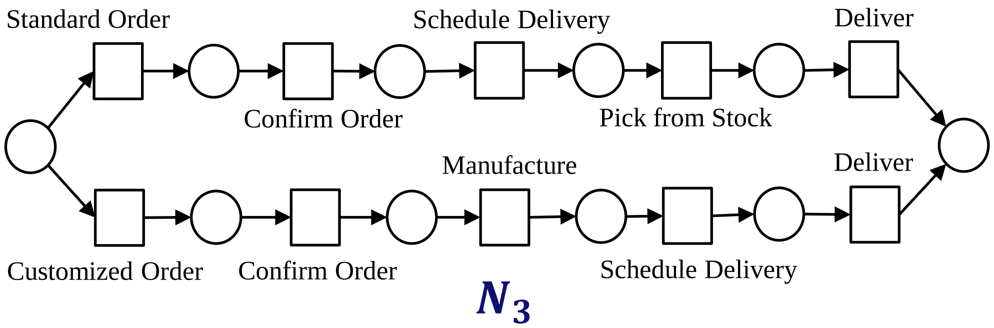
Recall the „carpenter" process from last instruction (Exercise 6). Given the following event log L describing observed behavior:

$$L = [\langle Standard, Confirm, Schedule, Pick, Deliver\rangle, \langle Customized, Confirm, Manufacture, Schedule, Deliver\rangle]$$

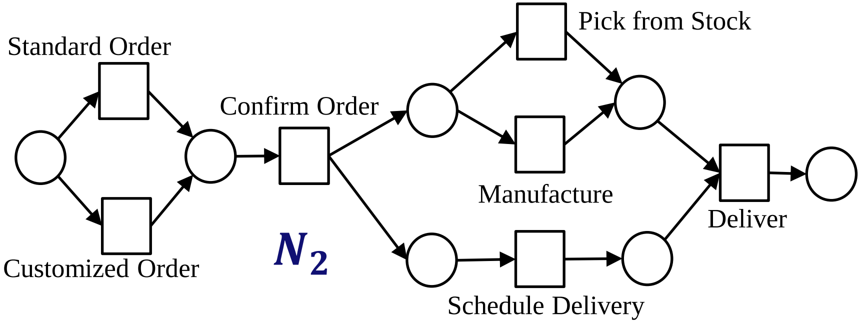
argue which of the following two Petri nets N_1, N_2 is better with respect to precision, fitness, generalization and simplicity.



Exercise 1b: Model Quality Assessment



VS.



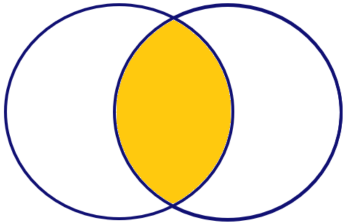
...argue which of the following two Petri nets N_3, N_2 is better with respect to precision, fitness, generalization and simplicity.

Chair of Process and Data Science

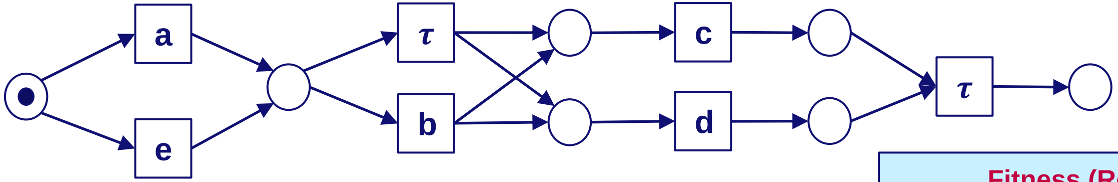
RWTH AACHEN UNIVERSITY

Exercise 2a: Model Quality Metrics

Consider the following model and the following event log. Argue about fitness and precision of the model using the complete model and log traces as model/log behavior.



L_{model} L



$$L = [\langle a, c\rangle^2, \langle e, d, c\rangle^5, \langle e, b, c, d\rangle^5]$$

Fitness (Recall):
To what extend can the behavior recorded in the log be replayed by the model?

Precision:
To what extend is the model behavior present in the log?

Fitness: $\dfrac{|L_{model} \cap L|}{|L|}$

With respect to trace frequencies

Precision: $\dfrac{|L_{model} \cap L|}{|L_{model}|}$

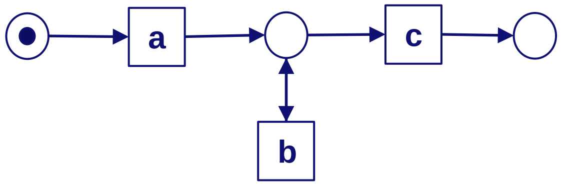
Without respect to trace frequencies

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Exercise 2b: Model Quality Metrics

Consider the following model and the following event log. When reasoning about model quality as before, what is the problem here?



$$L = [\langle a, c\rangle^5, \langle a, b, c\rangle^3, \langle a, b, b, c\rangle^3]$$

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Recap – Alpha Miner

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

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

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Recap – Alpha Miner

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

1. $T_L = \{ t \in T \mid \exists_{\sigma \in L}\, t \in \sigma \}$,

Identifying activities

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

Chair of Process and Data Science

Recap – Alpha Miner

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

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

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Recap – Alpha Miner

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

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

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Recap – Alpha Miner

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

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

(A, B) pairs

Recap – Alpha Miner

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

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

Deleting non maximal pairs

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Recap – Alpha Miner

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

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

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Recap – Alpha Miner

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

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

Alpha Miner - Example

Consider the following event log L:

$$L = [\langle a, b, c \rangle, \langle a, c \rangle]$$

1. Give the directly-follows matrix and the footprint matrix of L.
2. Derive the set of pairs of transition sets X and identify the maximal pairs Y.
3. Write down the resulting Petri net $\alpha(L)$.
4. Is the discovered model a sound workflow net?
5. Does the discovered process model represent both traces in the event log? If not, explain why.

Chair of Process and Data Science

Alpha Miner - Example

Give the directly-follows matrix and the footprint matrix of L.

$$L = [\langle a, b, c \rangle, \langle a, c \rangle]$$

Directly-follows relations

	a	b	c
a		>	>
b			>
c			

Footprint Matrix

	a	b	c
a	#	$\rightarrow$	$\rightarrow$
b	$\leftarrow$	#	$\rightarrow$
c	$\leftarrow$	$\leftarrow$	#

Causality**: x $\rightarrow$ y iff x > y and not y > x.**
Parallelism**: x || y iff x > y and y > x**
Independence**: x # y iff not x > y and not y > x.**

Alpha Miner - Example

Derive the set of pairs of maximal transition sets Y.

We look at all pairs of non-empty activity sets

$$X_L = \{ (A,B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_L b \;\wedge\; \forall_{a1,a2 \in A}\, a_1 \#_L a_2 \;\wedge\; \forall_{b1,b2 \in B}\, b_1 \#_L b_2 \}$$

Each activity in A should be causal $\rightarrow_L$ to each in B

All activities in each A and B should be independent

Then Y is the maximal pairs in X.

Suggestion:...

Chair of Process and Data Science

Alpha Miner - Example

Derive the set of pairs of maximal transition sets Y.

Suggestion: Write down pairs of sets A, B. Start with what you find in the footprint matrix. Mind that here already, a#a, b#b, c#c needs to hold.

A	B
$\{a\}$	$\{b\}$
$\{a\}$	$\{c\}$
$\{b\}$	$\{c\}$

Chair of Process and Data Science

Alpha Miner - Example

Derive the set of pairs of maximal transition sets Y.

Iteratively, combine rows by joining A with A and B with B, if applicable.

Footprint Matrix

	a	b	c
a	#	→	→
b	←	#	→
c	←	←	#

	A	B	
1	$\{a\}$	$\{b\}$	
2	$\{a\}$	$\{c\}$	
3	$\{b\}$	$\{c\}$	
1+2	~~$\{a\}$~~	~~$\{b,c\}$~~	**Cannot do that: b#c does not hold.**
2+3	~~$\{a,b\}$~~	~~$\{c\}$~~	**Cannot do that: a#b does not hold.**
1+3	~~$\{a,b\}$~~	~~$\{b,c\}$~~	

Cannot do that: a#b and b#c and b→b do not hold.

Chair of Process and Data Science

Alpha Miner - Example

Derive the set of pairs of maximal transition sets Y.

We are done: This is X.

A	B
$\{a\}$	$\{b\}$
$\{a\}$	$\{c\}$
$\{b\}$	$\{c\}$

All pairs A, B are maximal. Therefore, $Y = X$ in this case.

Chair of Process and Data Science

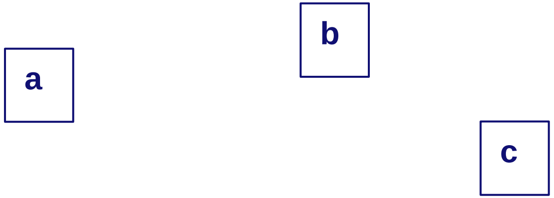
Alpha Miner - Example

L= [<a, b, c>, <a, c>]

1. $\boxed{T_L = \{ a, b, c\},}$
2. $T_I = \{ a \},$
3. $T_O = \{ c \},$

Footprint

	a	b	c
a	#	→	→
b	←	#	→
c	←	←	#



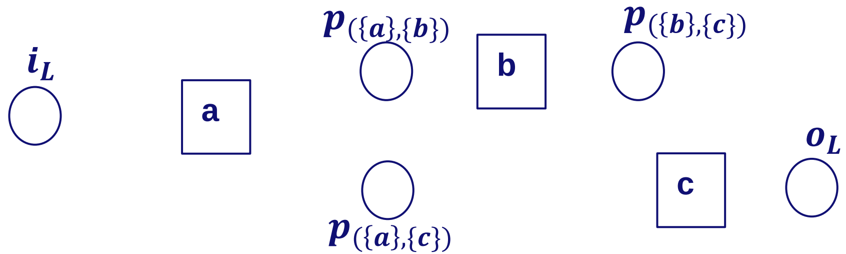
Alpha Miner - Example

L= [<a, b, c>, <a, c>]

1. $T_L = \{ a, b, c \}$,
2. $T_I = \{ a \}$,
3. $T_O = \{ c \}$,
4. $X_L = \{ (\{a\}, \{b\}), (\{b\}, \{c\}), (\{a\}, \{c\}) \}$
5. $Y_L = \{ (\{a\}, \{b\}), (\{b\}, \{c\}), (\{a\}, \{c\}) \}$
6. $\boxed{P_L = \{ p_{(A,B)} \mid (A,B) \in Y_L \} \cup \{i_L, o_L\},}$

Footprint

	a	b	c
a	#	→	→
b	←	#	→
c	←	←	#



Chair of Process and Data Science

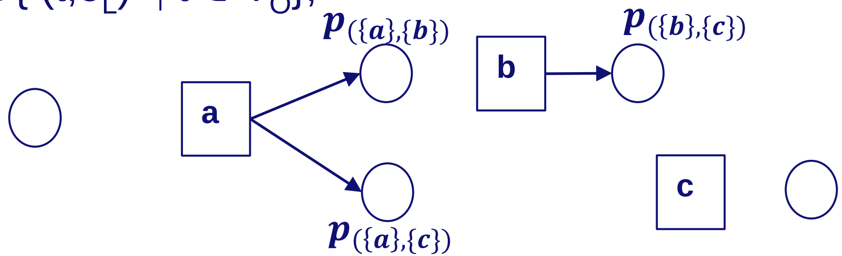
Alpha Miner - Example

L= [<a, b, c>, <a, c>]

1. $T_L = \{ a, b, c\}$,
2. $T_I = \{ a \}$,
3. $T_O = \{ c \}$,
4. $X_L = \{ (\{a\}, \{b\}), (\{b\}, \{c\}), (\{a\}, \{c\}) \}$
5. $Y_L = \{ (\{a\}, \{b\}), (\{b\}, \{c\}), (\{a\}, \{c\}) \}$
6. $P_L = \{ p_{(A,B)} \mid (A,B) \in Y_L \} \cup \{i_L,o_L\}$,
7. $F_L = \boxed{\{ (a,p_{(A,B)}) \mid (A,B) \in Y_L \wedge a \in A \}} \cup \{ (p_{(A,B)},b) \mid (A,B) \in Y_L \wedge b \in B \} \cup \{ (i_L,t) \mid t \in T_I\} \cup \{ (t,o_L) \mid t \in T_O\}$,

Footprint

	a	b	c
a	#	→	→
b	←	#	→
c	←	←	#



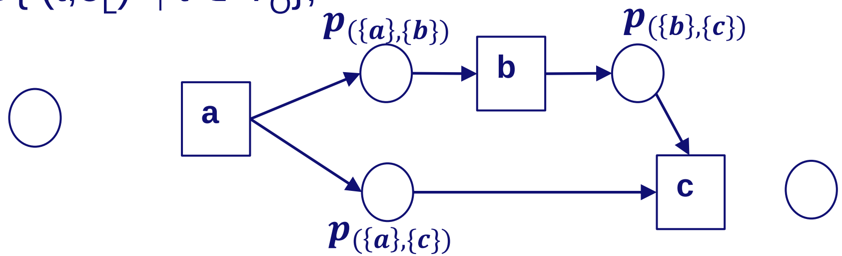
Alpha Miner - Example

L= [<a, b, c>, <a, c>]

1. $T_L = \{ a, b, c\}$,
2. $T_I = \{ a \}$,
3. $T_O = \{ c \}$,
4. $X_L = \{ (\{a\}, \{b\}), (\{b\}, \{c\}), (\{a\}, \{c\}) \}$
5. $Y_L = \{ (\{a\}, \{b\}), (\{b\}, \{c\}), (\{a\}, \{c\}) \}$
6. $P_L = \{ p_{(A,B)} \mid (A,B) \in Y_L \} \cup \{i_L, o_L\}$,
7. $F_L = \{ (a,p_{(A,B)}) \mid (A,B) \in Y_L \wedge a \in A \} \cup \boxed{\{ (p_{(A,B)},b) \mid (A,B) \in Y_L \wedge b \in B \}} \cup \{ (i_L,t) \mid t \in T_I\} \cup \{ (t,o_L) \mid t \in T_O\}$,

Footprint

	a	b	c
a	#	→	→
b	←	#	→
c	←	←	#



Chair of Process and Data Science

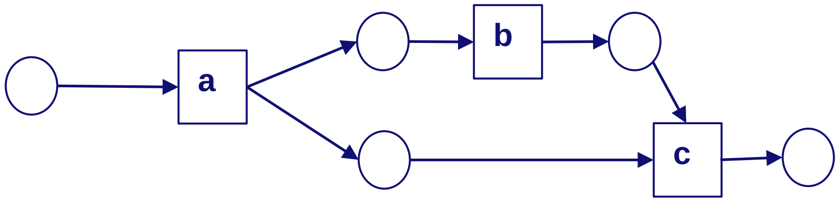
Alpha Miner - Example

L= [<a, b, c>, <a, c>]

1. $T_L = \{ a, b, c\}$,
2. $T_I = \{ a \}$,
3. $T_O = \{ c \}$,
4. $X_L = \{ (\{a\}, \{b\}), (\{b\}, \{c\}), (\{a\}, \{c\}) \}$
5. $Y_L = \{ (\{a\}, \{b\}), (\{b\}, \{c\}), (\{a\}, \{c\}) \}$
6. $P_L = \{ p_{(A,B)} \mid (A,B) \in Y_L \} \cup \{i_L, o_L\}$,
7. $F_L = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_L \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_L \wedge b \in B \} \cup \boxed{\{ (i_L, t) \mid t \in T_I\} \cup \{ (t, o_L) \mid t \in T_O\}}$,

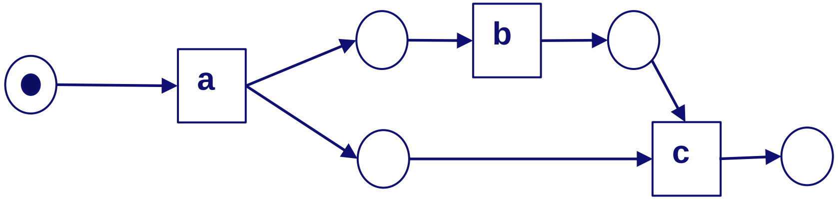
Footprint

	a	b	c
a	#	→	→
b	←	#	→
c	←	←	#



Chair of Process and Data Science

Exercise 3: Alpha Miner Example



$$L = [\langle a, b, c \rangle, \langle a, c \rangle]$$

1. Is the discovered model a sound workflow net?
2. Does the discovered process model represent both traces in the event log? If not, explain why.

Exercise 4: Alpha Miner – Practice, Practice

Consider the event logs given below. For each event log, think about what a Petri net might look like that describes the behavior suitably. Apply the Alpha algorithm, validate your result with ProM, and compare the result with your conceived net.
If the net is a sound workflow net, argue about precision and fitness (and generalization).

a) $L_1 = [\langle \textit{order, pay} \rangle, \langle \textit{order, reminder, pay} \rangle, \langle \textit{order, reminder, reminder, pay} \rangle]$
b) $L_2 = [\langle a,b,c,d,e,f,b,d,c,e,g \rangle, \langle a,b,d,c,e,g \rangle, \langle a,b,c,d,e,f,b,c,d,e,f,b,d,c,e,g \rangle]$
c) $L_3 = [\langle \textit{awake, eat, study, sleep} \rangle, \langle \textit{awake, study, eat, sleep} \rangle]$
d) $L_4 = [\langle \textit{awake, eat, study, eat, study, sleep} \rangle]$

Let $L_5 = [\langle \textit{awake, eat, study, eat, study, sleep} \rangle, \langle \textit{awake, study, eat, study, eat, sleep} \rangle]$. What Petri net would describe this process well? Compare the directly-follows matrices of L_3, L_4 and L_5. What is $\alpha(L_5)$?

Exercise 4: Alpha Miner – Practice, Practice

Consider the event logs given below. For each event log, think about what a Petri net might look like that describes the behavior suitably. Apply the Alpha algorithm, validate your result with ProM, and compare the result with your conceived net.
If the net is a sound workflow net, argue about precision and fitness (and generalization).

a) $L_1 = [\langle$*order, pay*$\rangle$, $\langle$*order, reminder, pay*$\rangle$, $\langle$*order, reminder, reminder, pay*$\rangle]$

b) $L_2 = [\langle$*a,b,c,d,e,f,b,d,c,e,g*$\rangle$, $\langle$*a,b,d,c,e,g*$\rangle$, $\langle$*a,b,c,d,e,f,b,c,d,e,f,b,d,c,e,g*$\rangle]$

c) $L_3 = [\langle$*awake, eat, study*

d) $L_4 = [\langle$*awake, eat, study*

Let $L_5 = [\langle$*awake, eat, study*
What Petri net would descr
matrices of L_3, L_4 and L_5. What is $\alpha(L_5)$?

***Use the alpha_L[xxx]* files. If you want to practice with more logs, you can just modify any of these files (with a standard Texteditor) so that it features the log you want.**
Take care to have a correct sorting (timestamps) and that each trace has a unique case id!

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Exercise 4: Alpha Miner – Practice, Practice

Consider the event logs given below. For each event log, think about what a Petri net might look like that describes the behavior suitably. Apply the Alpha algorithm, validate your result with ProM, and compare the result with your conceived net.
If the net is a sound workflow net, argue about precision and fitness (and generalization).

Be careful here!

a) $L_1 = [\langle \textit{order, pay} \rangle, \langle \textit{order, reminder, pay} \rangle, \langle \textit{order, reminder, reminder, pay} \rangle]$

b) $L_2 = [\langle a,b,c,d,e,f,b,d,c,e,g \rangle, \langle a,b,d,c,e,g \rangle, \langle a,b,c,d,e,f,b,c,d,e,f,b,d,c,e,g \rangle]$

c) $L_3 = [\langle \textit{awake, eat, study, sleep} \rangle, \langle \textit{awake, study, eat, sleep} \rangle]$

d) $L_4 = [\langle \textit{awake, eat, study, eat, study, sleep} \rangle]$

Hint: c and d are concurrent

Let $L_5 = [\langle \textit{awake, eat, study, eat, study, sleep} \rangle, \langle \textit{awake, study, eat, study, eat, sleep} \rangle]$**.**
What Petri net would describe this process well? Compare the directly-follows matrices of L_3, L_4 **and** L_5**. What is** $\alpha(L_5)$?

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Exercise 4: Alpha Miner – Practice, Practice

Consider the event logs given below. For each event log, think about what a Petri net might look like that describes the behavior suitably. Apply the Alpha algorithm, validate your result with ProM, and compare the result with your conceived net.
If the net is a sound workflow net, argue about precision and fitness (and generalization).

a) $L_1 = [\langle order, pay\rangle, \langle order, reminder, pay\rangle, \langle order, reminder, reminder, pay\rangle]$

b) $L_2 = [\langle a,b,c,d,e,f,b,d,c,e,g\rangle, \langle a,b,d,c,e,g\rangle, \langle a,b,c,d,e,f,b,c,d,e,f,b,d,c,e,g\rangle]$

c) $L_3 = [\langle awake, eat, study, sleep\rangle, \langle awake, study, eat, sleep\rangle]$

d) $L_4 = [\langle awake, eat, study, eat, study, sleep\rangle]$

Note:
Loop of length 1
Loop of length 2
Loop of length 3

Let $L_5 = [\langle awake, eat, study, eat, study, sleep\rangle, \langle awake, study, eat, study, eat, sleep\rangle]$. What Petri net would describe this process well? Compare the directly-follows matrices of L_3, L_4 and L_5. What is $\alpha(L_5)$?

Exercise 5 – Process Exploration

Hands On ProM and Celonis

Load the provided *delivery_management* event log into both ProM and Celonis to:

Suggestions / Hints

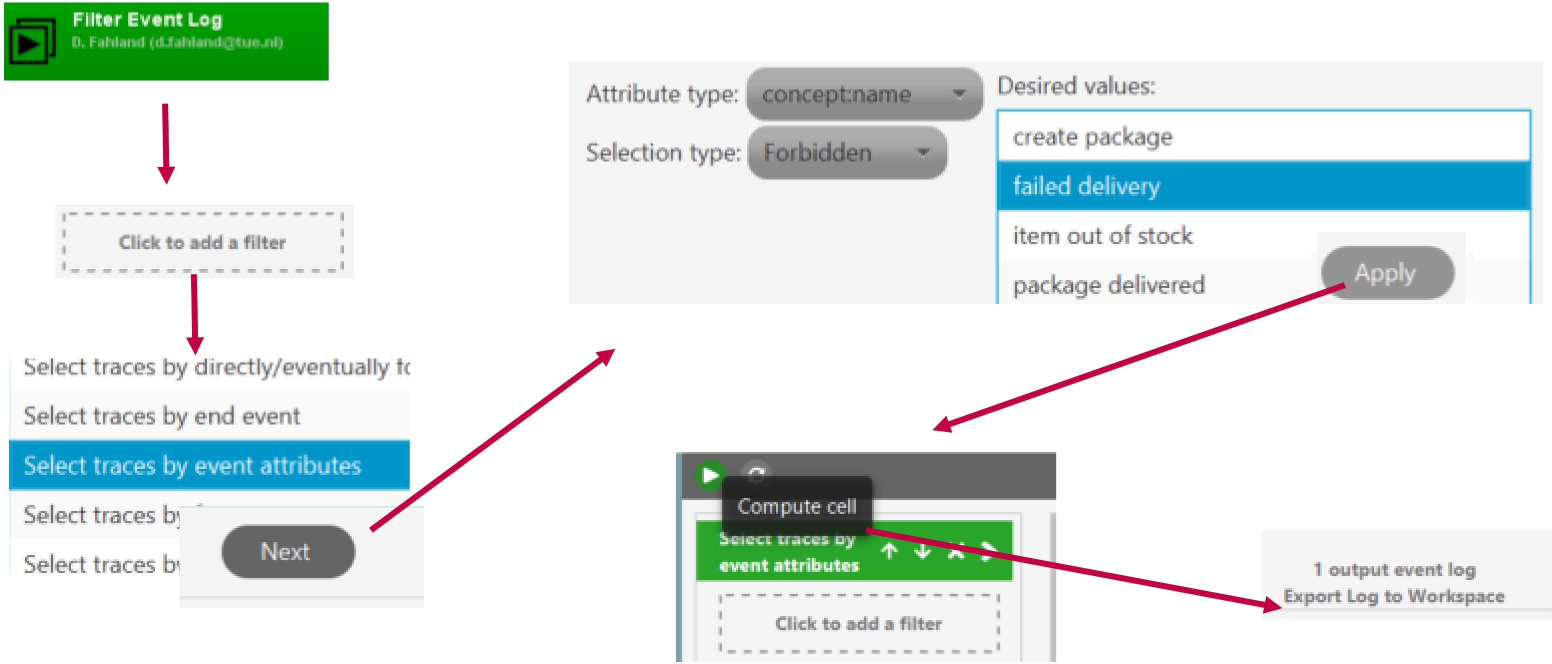
Task	ProM Plugins	Celonis Sheets
1. Give an overview of what days are busy (ProM) and the average throughput times of cases (Celonis).	Dotted Chart	Process Overview, (New Sheet with Line Chart)
2. Show the two most frequent variants and their DFGs (both tools).	(Explore Event Log), Inductive Visual Miner	New Sheet with Variant Explorer
3. Discover a model. *Advanced*: Discover a model only for cases that **do not** contain “failed delivery“ (both tools).	(any) Miner, Filter Event Log (next slide)	Conformance

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

ProM: Advanced Filtering

Example



Chair of Process and Data Science

RWTH AACHEN UNIVERSITY