On the α-Reconstructibility of Workflow Nets

Eric Badouel

INRIA - Rennes

June 2012

Discovering workflow nets from event logs

The (maximal) firing sequences of the worflow net =

all activities pertaining to a case from the time it enters the system (input place i is marked) until the case terminates and exits from the system (output place o is marked).

Discovering workflow nets from event logs

The (maximal) firing sequences of the worflow net =

all activities pertaining to a case from the time it enters the system (input place i is marked) until the case terminates and exits from the system (output place o is marked).

A workflow net is a contact-free, initially life, place simple and connected elementary net system $N = (P, T, F, M_0)$ where P contains an input place i and an output place o

Discovering workflow nets from event logs

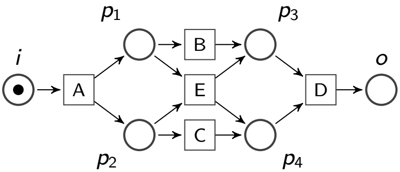
The (maximal) firing sequences of the worflow net =

all activities pertaining to a case from the time it enters the system (input place i is marked) until the case terminates and exits from the system (output place o is marked).

A workflow net is a contact-free, initially life, place simple and connected elementary net system $N = (P, T, F, M_0)$ where P contains an input place i and an output place o

- $^\bullet i = o^\bullet = \emptyset$

$$M[t\rangle M' \quad \Leftrightarrow \quad {}^\bullet t \subseteq M \;\wedge\; M \cap t^\bullet = \emptyset \\ M' = M \setminus {}^\bullet t \cup t^\bullet$$



$$\mathcal{L}(N) = \{u \in T^* \mid M_0[u\rangle M_t\} = \{ABCD, ACBD, AED\}$$

A net is contact free if $^\bullet t \subseteq M$ entails $M \cap t^\bullet = \emptyset$ for every reachable marking M; hence $M[t\rangle M' \Leftrightarrow \left({}^\bullet t \subseteq M \;\wedge\; M' = M \setminus {}^\bullet t \cup t^\bullet\right)$.

Discovering workflow nets from event logs

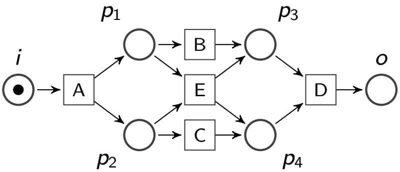
The (maximal) firing sequences of the worflow net =

all activities pertaining to a case from the time it enters the system (input place i is marked) until the case terminates and exits from the system (output place o is marked).

A workflow net is a contact-free, initially life, place simple and connected elementary net system $N = (P, T, F, M_0)$ where P contains an input place i and an output place o

- $^\bullet i = o^\bullet = \emptyset$
- p inner place: $^\bullet p \neq \emptyset \quad \wedge \quad p^\bullet \neq \emptyset$

$$M[t\rangle M' \quad \Leftrightarrow \quad \begin{array}{l} ^\bullet t \subseteq M \ \wedge \ M \cap t^\bullet = \emptyset \\ M' = M \setminus {^\bullet t} \cup t^\bullet \end{array}$$



$$\begin{array}{rcl} \mathcal{L}(N) & = & \{ u \in T^* \mid M_0[u\rangle M_t \} \\ & = & \{ABCD, ACBD, AED\} \end{array}$$

A net is contact free if $^\bullet t \subseteq M$ entails $M \cap t^\bullet = \emptyset$ for every reachable marking M; hence $M[t\rangle M' \Leftrightarrow \left(^\bullet t \subseteq M \ \wedge \ M' = M \setminus {^\bullet t} \cup t^\bullet \right)$.

Discovering workflow nets from event logs

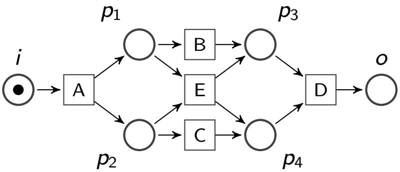
The (maximal) firing sequences of the worflow net =

all activities pertaining to a case from the time it enters the system (input place i is marked) until the case terminates and exits from the system (output place o is marked).

A workflow net is a contact-free, initially life, place simple and connected elementary net system $N = (P, T, F, M_0)$ where P contains an input place i and an output place o

- $^\bullet i = o^\bullet = \emptyset$
- p inner place: $^\bullet p \neq \emptyset \;\wedge\; p^\bullet \neq \emptyset$
- termination: $M_t = \{o\}$ is reachable from any marking reachable from $M_0 = \{i\}$.

$$M[t\rangle M' \quad \Leftrightarrow \quad \begin{array}{l} ^\bullet t \subseteq M \;\wedge\; M \cap t^\bullet = \emptyset \\ M' = M \setminus {}^\bullet t \cup t^\bullet \end{array}$$



$$\begin{array}{rcl} \mathcal{L}(N) & = & \{u \in T^* \mid M_0[u\rangle M_t\} \\ & = & \{ABCD, ACBD, AED\} \end{array}$$

A net is contact free if $^\bullet t \subseteq M$ entails $M \cap t^\bullet = \emptyset$ for every reachable marking M; hence $M[t\rangle M' \Leftrightarrow \left(^\bullet t \subseteq M \;\wedge\; M' = M \setminus {}^\bullet t \cup t^\bullet\right)$.

Discovering workflow nets from event logs

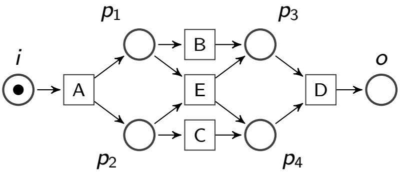
The (maximal) firing sequences of the worflow net =

all activities pertaining to a case from the time it enters the system (input place i is marked) until the case terminates and exits from the system (output place o is marked).

A workflow net is a contact-free, initially life, place simple and connected elementary net system $N = (P, T, F, M_0)$ where P contains an input place i and an output place o

- ${}^\bullet i = o^\bullet = \emptyset$
- p inner place: ${}^\bullet p \neq \emptyset \;\wedge\; p^\bullet \neq \emptyset$
- termination: $M_t = \{o\}$ is reachable from any marking reachable from $M_0 = \{i\}$.
- no scories: M_t is the only reachable marking that contains o.

$$M[t\rangle M' \quad \Leftrightarrow \quad {}^\bullet t \subseteq M \;\wedge\; M \cap t^\bullet = \emptyset$$
$$M' = M \setminus {}^\bullet t \cup t^\bullet$$



$$\mathcal{L}(N) = \{u \in T^* \mid M_0[u\rangle M_t\} = \{ABCD, ACBD, AED\}$$

A net is contact free if ${}^\bullet t \subseteq M$ entails $M \cap t^\bullet = \emptyset$ for every reachable marking M; hence $M[t\rangle M' \Leftrightarrow \left({}^\bullet t \subseteq M \;\wedge\; M' = M \setminus {}^\bullet t \cup t^\bullet\right)$.

α-abstraction of a log

α-Abstraction of a language $W \subseteq T^*$

$$
\begin{array}{rcll}
Abs(W) & = & \left\{ \boxed{\quad \bullet\, t} \mid t.T^* \cap W \neq \emptyset \right\} & t \in I_W \\
 & \cup & \left\{ \boxed{t \bullet t'} \mid T^*.t.t'.T^* \cap W \neq \emptyset \right\} & (t, t') \in C_W \\
 & \cup & \left\{ \boxed{t \bullet \quad} \mid T^*.t \cap W \neq \emptyset \right\} & t \in O_W
\end{array}
$$

α-abstraction of a log

α-Abstraction of a language $W \subseteq T^*$

$$
\begin{array}{rcll}
Abs(W) & = & \left\{ \boxed{\quad \bullet\, t} \mid t.T^* \cap W \neq \emptyset \right\} & t \in I_W \\
 & \cup & \left\{ \boxed{t \bullet t'} \mid T^*.t.t'.T^* \cap W \neq \emptyset \right\} & (t, t') \in C_W \\
 & \cup & \left\{ \boxed{t\, \bullet \quad} \mid T^*.t \cap W \neq \emptyset \right\} & t \in O_W
\end{array}
$$

Derived relations

$$
\begin{array}{llcl}
\text{causality:} & t \rightarrow_W t' & \Leftrightarrow & \boxed{t \bullet t'} \wedge \neg \boxed{t' \bullet t} \\
\text{conflict:} & t \sharp_W t' & \Leftrightarrow & \neg \boxed{t \bullet t'} \wedge \neg \boxed{t' \bullet t} \\
\text{concurrency:} & t \parallel_W t' & \Leftrightarrow & \boxed{t \bullet t'} \wedge \boxed{t' \bullet t}
\end{array}
$$

α-abstraction of a log

α-Abstraction of a language $W \subseteq T^*$

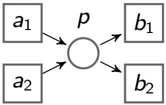
$$
\begin{array}{lcll}
Abs(W) & = & \left\{ \boxed{\quad\bullet\, t} \mid t.T^* \cap W \neq \emptyset \right\} & t \in I_W \\
 & \cup & \left\{ \boxed{t \bullet t'} \mid T^*.t.t'.T^* \cap W \neq \emptyset \right\} & (t,t') \in C_W \\
 & \cup & \left\{ \boxed{t\, \bullet \quad} \mid T^*.t \cap W \neq \emptyset \right\} & t \in O_W
\end{array}
$$

Derived relations

$$
\begin{array}{llcl}
\text{causality:} & t \rightarrow_W t' & \Leftrightarrow & \boxed{t \bullet t'} \wedge \neg \boxed{t' \bullet t} \\
\text{conflict:} & t \,\sharp_W\, t' & \Leftrightarrow & \neg \boxed{t \bullet t'} \wedge \neg \boxed{t' \bullet t} \\
\text{concurrency:} & t \parallel_W t' & \Leftrightarrow & \boxed{t \bullet t'} \wedge \boxed{t' \bullet t}
\end{array}
$$

For sets of transitions $A, B \subseteq T$, let $A \prec_W B$ when

1. $(\forall a \in A)(\forall b \in B) \quad a \rightarrow_W b$,
2. $(\forall a_1, a_2 \in A) \quad a_1 \sharp_W a_2$, and
3. $(\forall b_1, b_2 \in B) \quad b_1 \sharp_W b_2$



α-abstraction of a log

α-Abstraction of a language $W \subseteq T^*$

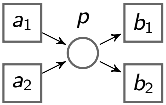
$$
\begin{array}{rcll}
Abs(W) & = & \{\, \boxed{\quad \bullet\, t} \mid t.T^* \cap W \neq \emptyset \,\} & t \in I_W \\
 & \cup & \{\, \boxed{t \bullet t'} \mid T^*.t.t'.T^* \cap W \neq \emptyset \,\} & (t, t') \in C_W \\
 & \cup & \{\, \boxed{t\, \bullet \quad} \mid T^*.t \cap W \neq \emptyset \,\} & t \in O_W
\end{array}
$$

Derived relations

$$
\begin{array}{lcccc}
\text{causality:} & t \rightarrow_W t' & \Leftrightarrow & & \boxed{t \bullet t'} \wedge \neg \boxed{t' \bullet t} \\
\text{conflict:} & t \sharp_W t' & \Leftrightarrow & \neg & \boxed{t \bullet t'} \wedge \neg \boxed{t' \bullet t} \\
\text{concurrency:} & t \parallel_W t' & \Leftrightarrow & & \boxed{t \bullet t'} \wedge \boxed{t' \bullet t}
\end{array}
$$

For sets of transitions $A, B \subseteq T$, let $A \prec_W B$ when

1. $(\forall a \in A)(\forall b \in B) \quad a \rightarrow_W b$,
2. $(\forall a_1, a_2 \in A) \quad a_1 \sharp_W a_2$, and
3. $(\forall b_1, b_2 \in B) \quad b_1 \sharp_W b_2$



Let $A \prec_W^m B$ when A and B are maximal sets with the property $A \prec_W B$, i.e.,
$A \prec_W^m B \Leftrightarrow (A \prec_W B) \wedge (A' \prec_W B' \wedge A \subseteq A' \wedge B \subseteq B' \Rightarrow A = A' \wedge B = B')$.

α-algorithm

$\alpha(W) = (P, T, F, M_0)$ defined as follows:

1. $P = \{i, o\} \cup \{p_{A,B} \mid \emptyset \neq A, B \subseteq T \wedge A \prec_W^m B\}$,
2. ${}^\bullet i = \emptyset$, and $i^\bullet = I_W$,
3. ${}^\bullet o = O_W$, and $o^\bullet = \emptyset$,
4. ${}^\bullet p_{A,B} = A$, and $p_{A,B}{}^\bullet = B$,
5. $M_0 = \{i\}$.

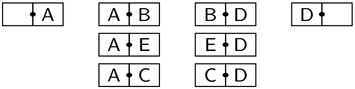
$W = \{ABCD, ACBD, AED\}$

α-algorithm

$\alpha(W) = (P, T, F, M_0)$ defined as follows:

1. $P = \{i, o\} \cup \{p_{A,B} \mid \emptyset \neq A, B \subseteq T \wedge A \prec_W^m B\}$,
2. ${}^\bullet i = \emptyset$, and $i^\bullet = I_W$,
3. ${}^\bullet o = O_W$, and $o^\bullet = \emptyset$,
4. ${}^\bullet p_{A,B} = A$, and $p_{A,B}{}^\bullet = B$,
5. $M_0 = \{i\}$.

$W = \{ABCD, ACBD, AED\}$

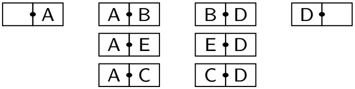


α-algorithm

$\alpha(W) = (P, T, F, M_0)$ defined as follows:

1. $P = \{i, o\} \cup \{p_{A,B} \mid \emptyset \neq A, B \subseteq T \wedge A \prec_W^m B\}$,
2. ${}^\bullet i = \emptyset$, and $i^\bullet = I_W$,
3. ${}^\bullet o = O_W$, and $o^\bullet = \emptyset$,
4. ${}^\bullet p_{A,B} = A$, and $p_{A,B}{}^\bullet = B$,
5. $M_0 = \{i\}$.

$W = \{ABCD, ACBD, AED\}$



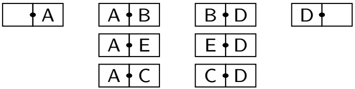
	B	C	D	E
A	$A \rightarrow_W B$	$A \rightarrow_W C$	$A \sharp_W D$	$A \rightarrow_W E$
B		$B \parallel_W C$	$B \rightarrow_W D$	$B \sharp_W E$
C			$C \rightarrow_W D$	$C \sharp_W E$
D				$E \rightarrow_W D$

α-algorithm

$\alpha(W) = (P, T, F, M_0)$ defined as follows:

1. $P = \{i, o\} \cup \{p_{A,B} \mid \emptyset \neq A, B \subseteq T \wedge A \prec^m_W B\}$,
2. ${}^\bullet i = \emptyset$, and $i^\bullet = I_W$,
3. ${}^\bullet o = O_W$, and $o^\bullet = \emptyset$,
4. ${}^\bullet p_{A,B} = A$, and $p_{A,B}{}^\bullet = B$,
5. $M_0 = \{i\}$.

$W = \{ABCD, ACBD, AED\}$



	B	C	D	E
A	$A \rightarrow_W B$	$A \rightarrow_W C$	$A \sharp_W D$	$A \rightarrow_W E$
B		$B \parallel_W C$	$B \rightarrow_W D$	$B \sharp_W E$
C			$C \rightarrow_W D$	$C \sharp_W E$
D				$E \rightarrow_W D$

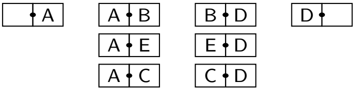
$\{A\} \prec^m_W \{B, E\}$ $\{A\} \prec^m_W \{C, E\}$
$\{B, E\} \prec^m_W \{D\}$ $\{C, E\} \prec^m_W \{D\}$

α-algorithm

$\alpha(W) = (P, T, F, M_0)$ defined as follows:

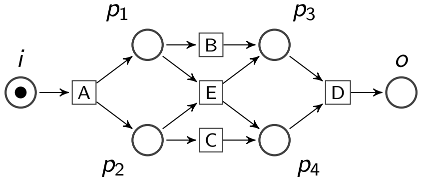
1. $P = \{i, o\} \cup \{p_{A,B} \mid \emptyset \neq A, B \subseteq T \wedge A \prec^m_W B\}$,
2. ${}^\bullet i = \emptyset$, and $i^\bullet = I_W$,
3. ${}^\bullet o = O_W$, and $o^\bullet = \emptyset$,
4. ${}^\bullet p_{A,B} = A$, and $p_{A,B}{}^\bullet = B$,
5. $M_0 = \{i\}$.

$W = \{ABCD, ACBD, AED\}$



	B	C	D	E
A	$A \rightarrow_W B$	$A \rightarrow_W C$	$A \sharp_W D$	$A \rightarrow_W E$
B		$B \parallel_W C$	$B \rightarrow_W D$	$B \sharp_W E$
C			$C \rightarrow_W D$	$C \sharp_W E$
D				$E \rightarrow_W D$

$\{A\} \prec^m_W \{B, E\}$ $\{A\} \prec^m_W \{C, E\}$
$\{B, E\} \prec^m_W \{D\}$ $\{C, E\} \prec^m_W \{D\}$



Some observations about α

Two places of a net constructed by algorithm α are incomparable for the order relation:

$$p \sqsubseteq q \Leftrightarrow ({}^\bullet p \subseteq {}^\bullet q \ \wedge \ p^\bullet \subseteq q^\bullet)$$

Some observations about α

Two places of a net constructed by algorithm α are incomparable for the order relation:

$$p \sqsubseteq q \Leftrightarrow ({}^\bullet p \subseteq {}^\bullet q \ \wedge \ p^\bullet \subseteq q^\bullet)$$

Do non maximal elements of $\prec_W$ always provide redundant places ?

A place p of a (contact-free) net system $N = (P, T, F, M_0)$ is a (structurally) implicit place if for every reachable marking M and transition $t \in p^\bullet$, ${}^\bullet t \setminus \{p\} \subseteq M \Rightarrow p \in M$.

Some observations about α

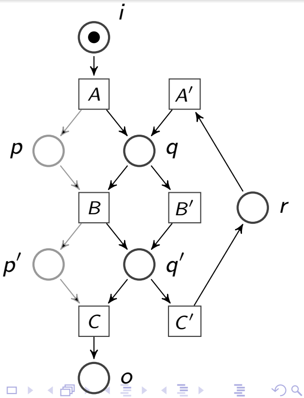
Two places of a net constructed by algorithm α are incomparable for the order relation:

$$p \sqsubseteq q \Leftrightarrow ({}^\bullet p \subseteq {}^\bullet q \ \wedge \ p^\bullet \subseteq q^\bullet)$$

Do non maximal elements of $\prec_W$ always provide redundant places ?

A place p of a (contact-free) net system $N = (P, T, F, M_0)$ is a (structurally) implicit place if for every reachable marking M and transition $t \in p^\bullet$, ${}^\bullet t \setminus \{p\} \subseteq M \Rightarrow p \in M$.

- $\mathcal{L}(N) = A\,(B'C'A')^*\,B\,(C'A'B')^*\,C$.
- A complete log is $W = \{ABC, AB'C'A'BC'A'B'C\}$.
- q, q' and r correspond to maximal elements of relation $\prec_W$:
  - $\{A, A'\} \prec_W^m \{B, B'\}$,
  - $\{B, B'\} \prec_W^m \{C, C'\}$,
  - $\{C'\} \prec_W^m \{A'\}$
- $p \sqsubset q$ and $p' \sqsubset q'$
- $\mathcal{L}(\alpha(W)) = A\,(B + B')\,(C'A'\,(B + B'))^*\,C$



Complete logs of a workflow net

Complete log

$W \subseteq \mathcal{L}(N)$ is a complete log of workflow net N if $Abs(W) = Abs(\mathcal{L}(N))$, i.e. it contains all the information used to synthesize $\alpha(\mathcal{L}(N))$; thus $\alpha(W) \cong \alpha(\mathcal{L}(N))$.

Complete logs of a workflow net

Complete log

$W \subseteq \mathcal{L}(N)$ is a complete log of workflow net N if $Abs(W) = Abs(\mathcal{L}(N))$, i.e. it contains all the information used to synthesize $\alpha(\mathcal{L}(N))$; thus $\alpha(W) \cong \alpha(\mathcal{L}(N))$.

Algorithm α is sober

- For any complete log $W \subseteq \mathcal{L}(N)$ of a workflow net N any larger log $W \subseteq W' \subseteq \mathcal{L}(N)$ is also complete
- The minimal size of complete logs of workflow nets is asymptotically negligible w.r.t. the size of their languages.

The size of $Abs(\mathcal{L}(N))$ is in $O(|T|^2)$. Moreover, a firing sequence of N contained in a log W may contribute several pairs of transitions in C_W.

Complete logs of a workflow net

Complete log

$W \subseteq \mathcal{L}(N)$ is a complete log of workflow net N if $Abs(W) = Abs(\mathcal{L}(N))$, i.e. it contains all the information used to synthesize $\alpha(\mathcal{L}(N))$; thus $\alpha(W) \cong \alpha(\mathcal{L}(N))$.

Algorithm α is sober

- For any complete log $W \subseteq \mathcal{L}(N)$ of a workflow net N any larger log $W \subseteq W' \subseteq \mathcal{L}(N)$ is also complete
- The minimal size of complete logs of workflow nets is asymptotically negligible w.r.t. the size of their languages.

The size of $Abs(\mathcal{L}(N))$ is in $O(|T|^2)$. Moreover, a firing sequence of N contained in a log W may contribute several pairs of transitions in C_W.

Sobriety means that one can assume $W \subseteq \mathcal{L}(N)$ to be a complete log of workflow net N as soon as it contains a reasonable number of its execution sequences.

Discovery of a workflow net from one of its complete logs

Workflow net discovery:

A workflow net N is α-reconstructible, i.e., $N \cong \alpha(\mathcal{L}(N))$ if and only if it can be discovered from any of its complete log $W \subseteq \mathcal{L}(N)$, i.e., $N \cong \alpha(W)$.

Discovery of a workflow net from one of its complete logs

Workflow net discovery:

A workflow net N is α-reconstructible, i.e., $N \cong \alpha(\mathcal{L}(N))$ if and only if it can be discovered from any of its complete log $W \subseteq \mathcal{L}(N)$, i.e., $N \cong \alpha(W)$.

Remark:

$W \subseteq \mathcal{L}(N)$ is a complete log of an unknown α-reconstructible workflow net N. if and only if the following two conditions hold:
(i) $\alpha(W)$ is a workflow net, such that $W \subseteq \mathcal{L}(\alpha(W))$, and
(ii) $Abs(W) = Abs(\alpha(W))$, i.e. W is also a complete log of $\alpha(W)$.

Discovery of a workflow net from one of its complete logs

Workflow net discovery:

A workflow net N is α-reconstructible, i.e., $N \cong \alpha(\mathcal{L}(N))$ if and only if it can be discovered from any of its complete log $W \subseteq \mathcal{L}(N)$, i.e., $N \cong \alpha(W)$.

Remark:

$W \subseteq \mathcal{L}(N)$ is a complete log of an unknown α-reconstructible workflow net N. if and only if the following two conditions hold:
(i) $\alpha(W)$ is a workflow net, such that $W \subseteq \mathcal{L}(\alpha(W))$, and
(ii) $Abs(W) = Abs(\alpha(W))$, i.e. W is also a complete log of $\alpha(W)$.

- $\Rightarrow$ Conditions (i) and (ii) holds because $W \subseteq \mathcal{L}(N)$ is a complete log of $N \cong \alpha(W)$.
- $\Leftarrow$ Let $N = \alpha(W)$, then W is a complete log of N (by i and ii) and N is α-reconstructible: $N = \alpha(W) \cong \alpha(\mathcal{L}(N))$

α construction is not a closure operation

One may have expected any net constructed by algorithm α to be an α reconstructible workflow net. Or more precisely, that α algorithm computes a closure operation providing the best approximation of a given log by a workflow net.

α construction is not a closure operation

One may have expected any net constructed by algorithm α to be an α reconstructible workflow net. Or more precisely, that α algorithm computes a closure operation providing the best approximation of a given log by a workflow net.

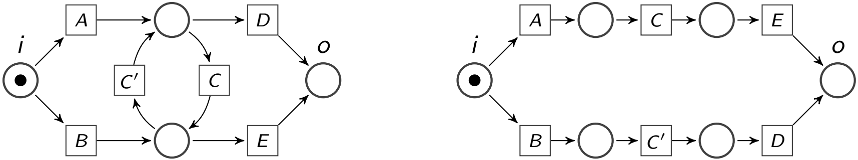
No Galois connection: $W \subseteq \mathcal{L}(N) \;\Leftrightarrow\; N \leq \alpha(W)$

α construction is not a closure operation

One may have expected any net constructed by algorithm α to be an α reconstructible workflow net. Or more precisely, that α algorithm computes a closure operation providing the best approximation of a given log by a workflow net.

No Galois connection: $W \subseteq \mathcal{L}(N) \Leftrightarrow N \leq \alpha(W)$

- $W = \{ACC'D, AD, BC'CE, BE\}$ is a complete log of net N (on the left)

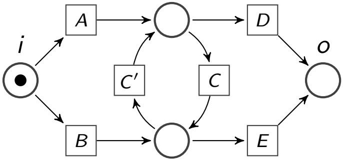


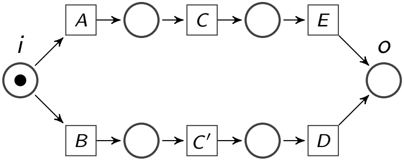
α construction is not a closure operation

One may have expected any net constructed by algorithm α to be an α reconstructible workflow net. Or more precisely, that α algorithm computes a closure operation providing the best approximation of a given log by a workflow net.

No Galois connection: $W \subseteq \mathcal{L}(N) \;\Leftrightarrow\; N \leq \alpha(W)$

- $W = \{ACC'D, AD, BC'CE, BE\}$ is a complete log of net N (on the left)
- Because of the short loop between C and C' the α-algorithm infers $C \|_W C'$



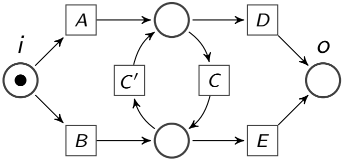


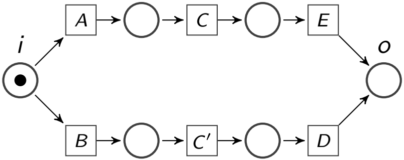
α construction is not a closure operation

One may have expected any net constructed by algorithm α to be an α reconstructible workflow net. Or more precisely, that α algorithm computes a closure operation providing the best approximation of a given log by a workflow net.

No Galois connection: $W \subseteq \mathcal{L}(N) \;\Leftrightarrow\; N \leq \alpha(W)$

- $W = \{ACC'D, AD, BC'CE, BE\}$ is a complete log of net N (on the left)
- Because of the short loop between C and C' the α-algorithm infers $C \|_W C'$
- The language of the resulting net $\alpha(W)$, shown on the right, namely $\{ACE, BC'D\}$ does not reproduce the execution sequences in W



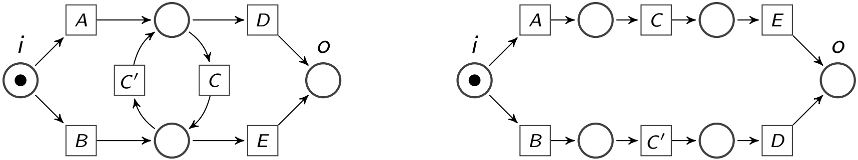


α construction is not a closure operation

One may have expected any net constructed by algorithm α to be an α reconstructible workflow net. Or more precisely, that α algorithm computes a closure operation providing the best approximation of a given log by a workflow net.

No Galois connection: $W \subseteq \mathcal{L}(N) \;\Leftrightarrow\; N \leq \alpha(W)$

- $W = \{ACC'D, AD, BC'CE, BE\}$ is a complete log of net N (on the left)
- Because of the short loop between C and C' the α-algorithm infers $C\|_W C'$
- The language of the resulting net $\alpha(W)$, shown on the right, namely $\{ACE, BC'D\}$ does not reproduce the execution sequences in W
- Thus, W is the complete log of some workflow net such that $W \not\subseteq \mathcal{L}(\alpha(W))$



Structural workflow nets: a sufficient condition for α-reconstructibility ...

Structured workflow nets

A workflow net $N = (P, T, F, M_0)$ is a structured workflow net if it has no structurally implicit places and the following condition holds:

$$\forall t \in T \quad |{}^\bullet t| > 1 \Rightarrow (\forall p \in {}^\bullet t \quad |{}^\bullet p| = 1 \ \wedge \ |p^\bullet| = 1) \qquad \text{(SWN)}$$

i.e., if a transition t requires the synchronization of several conditions (places), then each of these conditions has a unique cause ($|{}^\bullet p| = 1$) and a unique consequence ($|p^\bullet| = 1$), hence it cannot induce a conflict between t and another transition t'.

Structural workflow nets: a sufficient condition for α-reconstructibility ...

Structured workflow nets

A workflow net $N = (P, T, F, M_0)$ is a structured workflow net if it has no structurally implicit places and the following condition holds:

$$\forall t \in T \quad |{}^\bullet t| > 1 \Rightarrow (\forall p \in {}^\bullet t \quad |{}^\bullet p| = 1 \ \wedge \ |p^\bullet| = 1) \tag{SWN}$$

i.e., if a transition t requires the synchronization of several conditions (places), then each of these conditions has a unique cause ($|{}^\bullet p| = 1$) and a unique consequence ($|p^\bullet| = 1$), hence it cannot induce a conflict between t and another transition t'.

van der Aalst et al

Structured workflow nets without short loops are α-reconstructible

Structural workflow nets: a sufficient condition for α-reconstructibility ...

Structured workflow nets

A workflow net $N = (P, T, F, M_0)$ is a structured workflow net if it has no structurally implicit places and the following condition holds:

$$\forall t \in T \quad |^\bullet t| > 1 \Rightarrow (\forall p \in {}^\bullet t \quad |^\bullet p| = 1 \ \wedge \ |p^\bullet| = 1) \tag{SWN}$$

i.e., if a transition t requires the synchronization of several conditions (places), then each of these conditions has a unique cause ($|^\bullet p| = 1$) and a unique consequence ($|p^\bullet| = 1$), hence it cannot induce a conflict between t and another transition t'.

van der Aalst et al

Structured workflow nets without short loops are α-reconstructible

- Adding structurally implicit places to a net preserves its language
- removing places from a net satisfying condition (SWN) cannot invalidate this condition.

Structural workflow nets: a sufficient condition for α-reconstructibility ...

Structured workflow nets

A workflow net $N = (P, T, F, M_0)$ is a structured workflow net if it has no structurally implicit places and the following condition holds:

$$\forall t \in T \quad |{}^\bullet t| > 1 \Rightarrow (\forall p \in {}^\bullet t \quad |{}^\bullet p| = 1 \ \wedge \ |p^\bullet| = 1) \qquad \text{(SWN)}$$

i.e., if a transition t requires the synchronization of several conditions (places), then each of these conditions has a unique cause ($|{}^\bullet p| = 1$) and a unique consequence ($|p^\bullet| = 1$), hence it cannot induce a conflict between t and another transition t'.

van der Aalst et al

Structured workflow nets without short loops are α-reconstructible

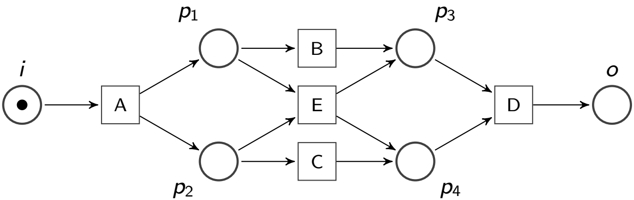
- Adding structurally implicit places to a net preserves its language
- removing places from a net satisfying condition (SWN) cannot invalidate this condition.

Corollary

A workflow net N without short loops and satisfying condition (SWN) is always language equivalent to some α-reconstructible workflow net N'.

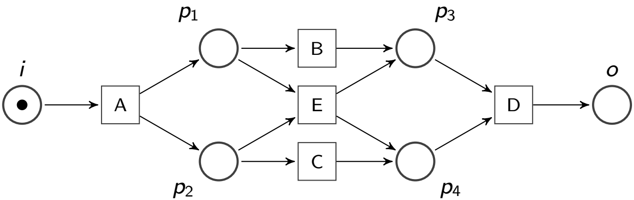
… which is not a necessary condition.

An α reconstructible net which does not satisfy (SWN)

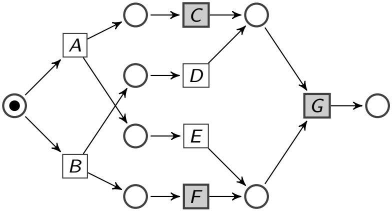


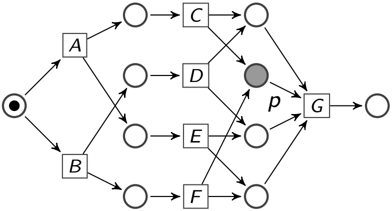
... which is not a necessary condition.

An α reconstructible net which does not satisfy (SWN)



An α-reconstructible workflow net with implicit places





α-reconstructible workflow nets

For every place p of an elementary net

1. $\forall a, a' \in {}^\bullet p \qquad a \sharp_N a'$,
2. $\forall b, b' \in p^\bullet \qquad b \sharp_N b'$,
3. $\forall a \in {}^\bullet p \; \forall b \in p^\bullet \qquad a \rightarrow_N b$

$$\begin{array}{rcl} t \rightarrow_N t' & \Leftrightarrow & t^\bullet \cap {}^\bullet t' \neq \emptyset \\ t \sharp_N t' & \Leftrightarrow & ({}^\bullet t \cap {}^\bullet t') \cup (t^\bullet \cap t'^\bullet) \neq \emptyset \\ t \parallel_N t' & \Leftrightarrow & ({}^\bullet t \cup t'^\bullet) \cap ({}^\bullet t \cup t'^\bullet) = \emptyset \end{array}$$

α-reconstructible workflow nets

For every place p of an elementary net

1. $\forall a, a' \in {}^\bullet p \qquad a \sharp_N a'$,
2. $\forall b, b' \in p^\bullet \qquad b \sharp_N b'$,
3. $\forall a \in {}^\bullet p \, \forall b \in p^\bullet \qquad a \rightarrow_N b$

$$\begin{array}{rcl} t \rightarrow_N t' & \Leftrightarrow & t^\bullet \cap {}^\bullet t' \neq \emptyset \\ t \sharp_N t' & \Leftrightarrow & ({}^\bullet t \cap {}^\bullet t') \cup (t^\bullet \cap t'^\bullet) \neq \emptyset \\ t \parallel_N t' & \Leftrightarrow & ({}^\bullet t \cup t'^\bullet) \cap ({}^\bullet t \cup t'^\bullet) = \emptyset \end{array}$$

Let $W = \mathcal{L}(N)$ be the full log of a workflow net

1. $\rightarrow_W \subseteq \rightarrow_N \qquad \|_W t' \subseteq \|_N$, and $\sharp_N \subseteq \sharp_W$
2. if t and t' are co-enabled, i.e. there exists some reachable marking M such that $M[t\rangle$ and $M[t'\rangle$, then $t \|_N t' \Leftrightarrow t \|_W t'$ and $t \sharp_N t' \Leftrightarrow t \sharp_W t'$

α-reconstructible workflow nets

For every place p of an elementary net

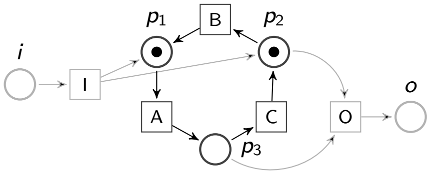
1. $\forall a, a' \in {}^\bullet p \qquad a \sharp_N a'$,
2. $\forall b, b' \in p^\bullet \qquad b \sharp_N b'$,
3. $\forall a \in {}^\bullet p \; \forall b \in p^\bullet \qquad a \rightarrow_N b$

$$
\begin{aligned}
t \rightarrow_N t' &\Leftrightarrow t^\bullet \cap {}^\bullet t' \neq \emptyset \\
t \sharp_N t' &\Leftrightarrow ({}^\bullet t \cap {}^\bullet t') \cup (t^\bullet \cap t'^\bullet) \neq \emptyset \\
t \parallel_N t' &\Leftrightarrow ({}^\bullet t \cup t'^\bullet) \cap ({}^\bullet t \cup t'^\bullet) = \emptyset
\end{aligned}
$$

Let $W = \mathcal{L}(N)$ be the full log of a workflow net

1. $\rightarrow_W \subseteq \rightarrow_N \qquad \|_W t' \subseteq \|_N$, and $\sharp_N \subseteq \sharp_W$
2. if t and t' are co-enabled, i.e. there exists some reachable marking M such that $M[t\rangle$ and $M[t'\rangle$, then $t \|_N t' \Leftrightarrow t \|_W t'$ and $t \sharp_N t' \Leftrightarrow t \sharp_W t'$

The importance of contact-freeness

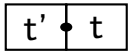
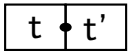


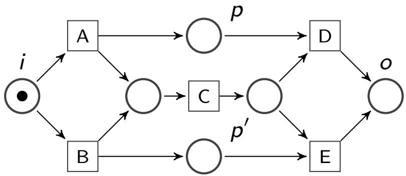
$\mathcal{L}(N) = I(ABC)^* AO$

$A \rightarrow_W B$	**vs**	$B \rightarrow_N A$
$B \rightarrow_W C$		$A \rightarrow_N C$
$C \rightarrow_W A$		$C \rightarrow_N B$

Boundary places

How to ensure $\rightarrow_N \subseteq \rightarrow_W$?

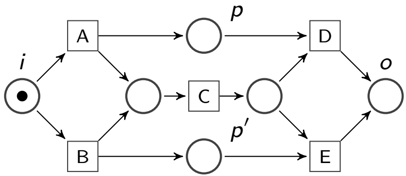
1. N workflow net without short loops: $t^\bullet \cap {}^\bullet t' \Rightarrow \neg$ 
2. Boundary place: inner place s.t. $\forall t \in {}^\bullet p \;\; \forall t' \in p^\bullet$ .
3. $\rightarrow_N = \rightarrow_{\mathcal{L}(N)}$ if N is a workflow net without short loops and all of whose inner places are boundary places



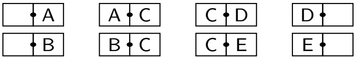
Boundary places

How to ensure $\longrightarrow_N \subseteq \longrightarrow_W$?

1. N workflow net without short loops: $t^\bullet \cap {}^\bullet t' \;\Rightarrow\; \neg\, \boxed{t' \bullet t}$
2. Boundary place: inner place s.t. $\forall t \in {}^\bullet p \;\; \forall t' \in p^\bullet \;\; \boxed{t \bullet t'}$.
3. $\longrightarrow_N = \longrightarrow_{\mathcal{L}(N)}$ if N is a workflow net without short loops and all of whose inner places are boundary places



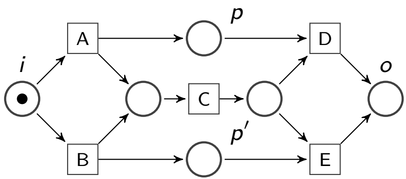
$\mathcal{L}(N) = \{ACD, BCE\}$



Boundary places

How to ensure $\rightarrow_N \subseteq \rightarrow_W$?

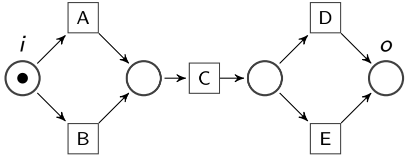
1. N workflow net without short loops: $t^\bullet \cap {}^\bullet t' \Rightarrow \neg\ \boxed{t' \bullet t}$
2. Boundary place: inner place s.t. $\forall t \in {}^\bullet p\ \ \forall t' \in p^\bullet\ \ \boxed{t \bullet t'}$.
3. $\rightarrow_N = \rightarrow_{\mathcal{L}(N)}$ if N is a workflow net without short loops and all of whose inner places are boundary places



$\mathcal{L}(N) = \{ACD, BCE\}$

$\bullet$ A	A $\bullet$ C	C $\bullet$ D	D $\bullet$
$\bullet$ B	B $\bullet$ C	C $\bullet$ E	E $\bullet$

$\alpha(\mathcal{L}(N))$: p and p' are neither (structurally) implit places nor boundary places



α-reconstructibility of workflow nets

Theorem

A workflow net N is α-reconstructible if and only if

1. It has no short loop
2. Every inner place is a boundary place
3. $^\bullet p \subseteq {}^\bullet q \ \wedge\ p^\bullet \subseteq q^\bullet \quad \Rightarrow \quad p = q$
4. There exists places witnessing for relation $\prec = \prec_{\mathcal{L}(N)}$:

$$A \prec B \ \Rightarrow\ \exists p \in P \ \textbf{s.t.}\ A \subseteq {}^\bullet p \ \wedge\ B \subseteq p^\bullet$$

$^\bullet p \prec p^\bullet$ for every place p of a workflow net without short loops and whose inner places are boundary places; thus the pairs $\langle {}^\bullet p, p^\bullet \rangle$ are the maximal elements of $\prec$ (w.r.t; $\sqsubseteq$).

Processes

$\mathcal{R} = (R, \ell)$ process of a workflow net $N = (P, T, F, M_0)$

a net $R = (P_R, T_R, F_R)$ and two labelling functions $\ell : T_R \to T$ and $\ell : P_R \to \wp(P)$ such that:

1. There is a place i_R such that ${}^\bullet i_R = \emptyset$, and $\ell(i_R) = \{i\}$ where i is the input place of the workflow net N.
2. There is a place o_R such that $o_R{}^\bullet = \emptyset$, and $\ell(o_R) = \{o\}$ where o is the output place of the workflow net N.
3. $\forall p_R \in P_R \setminus \{i_R, o_R\} \qquad |{}^\bullet p_R| = 1 \quad \text{and} \quad |p_R{}^\bullet| = 1.$
4. $\forall t_R \in T_R \qquad {}^\bullet t_R \neq \emptyset \quad \text{and} \quad t_R{}^\bullet \neq \emptyset.$
5. The underlying graph of R is acyclic.
6. $\{\ell(p_R) \mid p_R \in {}^\bullet t_R\}$ is a partition of ${}^\bullet \ell(t_R)$.
7. $\{\ell(p_R) \mid p_R \in t_R{}^\bullet\}$ is a partition of $\ell(t_R)^\bullet$.

Processes

$\mathcal{R} = (R, \ell)$ process of a workflow net $N = (P, T, F, M_0)$

a net $R = (P_R, T_R, F_R)$ and two labelling functions $\ell : T_R \to T$ and $\ell : P_R \to \wp(P)$ such that:

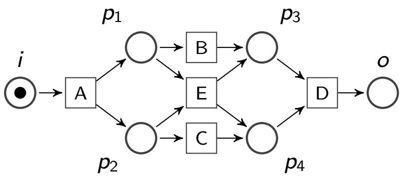
1. There is a place i_R such that ${}^\bullet i_R = \emptyset$, and $\ell(i_R) = \{i\}$ where i is the input place of the workflow net N.
2. There is a place o_R such that $o_R{}^\bullet = \emptyset$, and $\ell(o_R) = \{o\}$ where o is the output place of the workflow net N.
3. $\forall p_R \in P_R \setminus \{i_R, o_R\} \quad |{}^\bullet p_R| = 1 \quad \text{and} \quad |p_R{}^\bullet| = 1$.
4. $\forall t_R \in T_R \quad {}^\bullet t_R \neq \emptyset \quad \text{and} \quad t_R{}^\bullet \neq \emptyset$.
5. The underlying graph of R is acyclic.
6. $\{\ell(p_R) \mid p_R \in {}^\bullet t_R\}$ is a partition of ${}^\bullet \ell(t_R)$.
7. $\{\ell(p_R) \mid p_R \in t_R{}^\bullet\}$ is a partition of $\ell(t_R)^\bullet$.

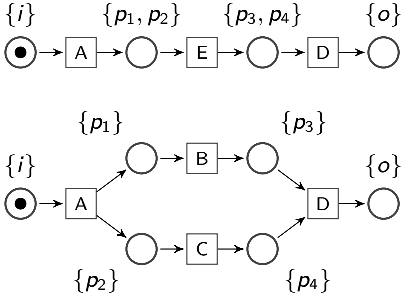
Proposition

Processes $\mathcal{R} = (R, \ell)$ of a workflow net N are in bijective correspondence with the equivalences classes of complete execution sequences of N modulo permutation of concurrent transitions.

Characterization of boundary places

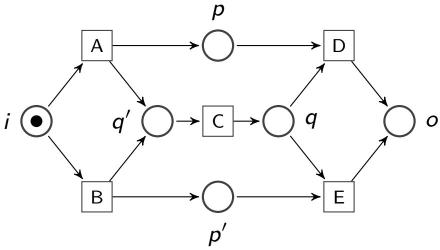
An inner place p of a workflow net is a boundary place if and only if for every pair of transitions $t \in {}^\bullet p$ and $t' \in p^\bullet$ there exists a process $\mathcal{R} = (R, \ell)$ of N and a non (structurally) implicit place $p_R \in P_R$ in this process with $p_R \in {t_R}^\bullet \cap {}^\bullet t'_R$ such that $\ell(t_R) = t$, $\ell(t'_R) = t'$ and $p \in \ell(p_R)$.

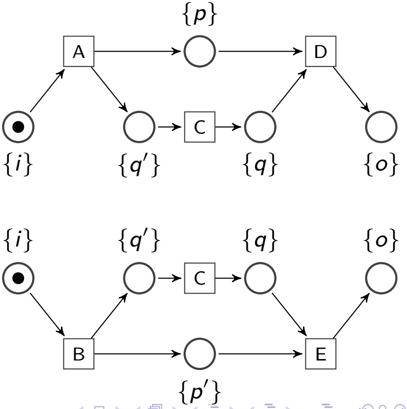




Characterization of boundary places

An inner place p of a workflow net is a boundary place if and only if for every pair of transitions $t \in {}^\bullet p$ and $t' \in p^\bullet$ there exists a process $\mathcal{R} = (R, \ell)$ of N and a non (structurally) implicit place $p_R \in P_R$ in this process with $p_R \in {t_R}^\bullet \cap {}^\bullet t'_R$ such that $\ell(t_R) = t$, $\ell(t'_R) = t'$ and $p \in \ell(p_R)$.

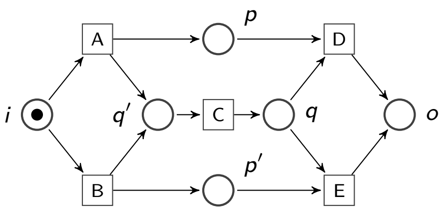




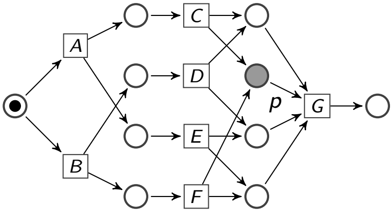
Boundary places vs non implicit places

An inner place of a structured workflow net is a boundary place if and only it is a non implicit place.

These two notions differs in general



Places p are p' are neither boundary nor implicit places.



A place which is both a boundary place and an implicit place

Conclusion

- We presented a characterization of the class of α-reconstructible workflow nets.
  - Limited interest from a practical point of view: this class is not given by structural properties.
  - It may however pave the way to the discovery of interesting classes of α-reconstructible workflow nets larger than the class of structured workflow nets.
- The variant mining algorithm based on regions (ω-algorithm) is more expressive
  - The two algorithms do not solve the same problem: recovery for α versus approximation of a log by a workflow net for ω.
  - For a fixed log ω is computationally much more costy but it can me made incremental (we refine the workflow approximation when the log increases).
  - There is potentially room for the design of intermediate mining algorithms.