Process Model Discovery: A Method Based on Transition System Decomposition

Anna A. Kalenkova[1], Irina A. Lomazova[1] and Wil M.P.van der Aalst[2,1]

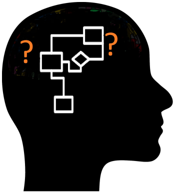
[1]Higher School of Economics — PAIS Lab,
[2]Eindhoven University of Technology

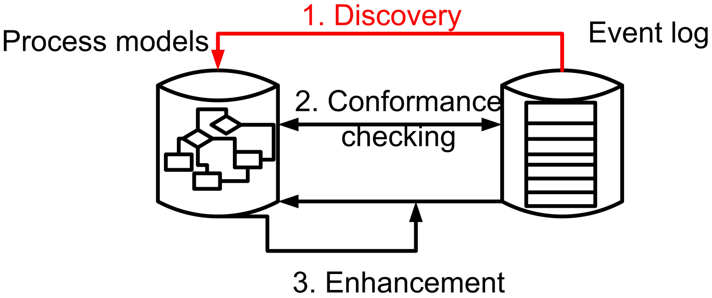
35th International Conference on Application and Theory of Petri Nets and Concurrency. Tunis, Tunisia

Outline

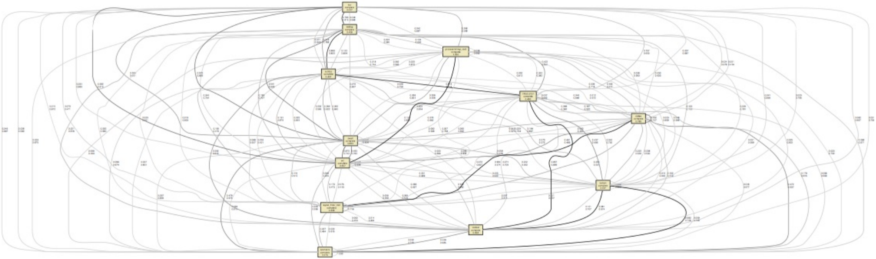
- Two words about Process mining
- Motivating example
- Basic definitions
- Discovery algorithm based on transition system decomposition
- Structural and behavioral properties preserved by decomposed discovery
- Region based algorithms
- Conclusion

Process mining





Quality dimensions. Simplicity



Motivating example. Log and transition system

$$L = \{\langle start\_booking, book\_flight, get\_insurance, send\_email, choose\_payment\_type, pay\_by\_card, complite\_booking\rangle,$$

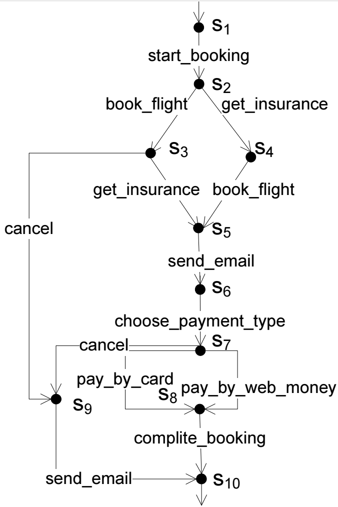
$$\langle start\_booking, get\_insurance, book\_flight, send\_email, choose\_payment\_type, pay\_by\_card, complite\_booking\rangle,$$

$$\langle start\_booking, get\_insurance, book\_flight, send\_email, choose\_payment\_type, pay\_by\_web\_money, complite\_booking\rangle,$$

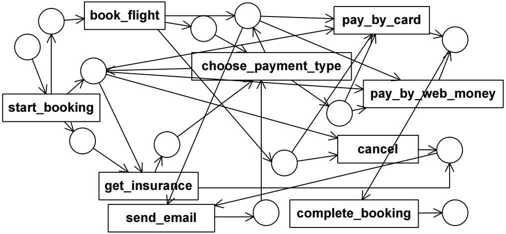
$$\langle start\_booking, book\_flight, get\_insurance, send\_email, choose\_payment\_type, , pay\_by\_web\_money, complite\_booking\rangle,$$

$$\langle start\_booking, book\_flight, cancel, send\_email\rangle,$$

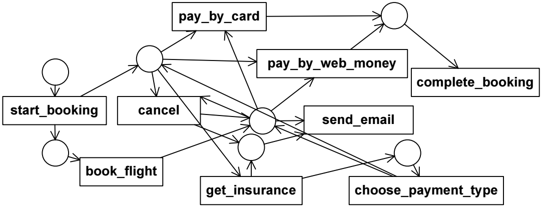
$$\langle start\_booking, book\_flight, get\_insurance, send\_email, choose\_payment\_type, cancel, send\_email\rangle\}.$$



Motivating example. Discovered models

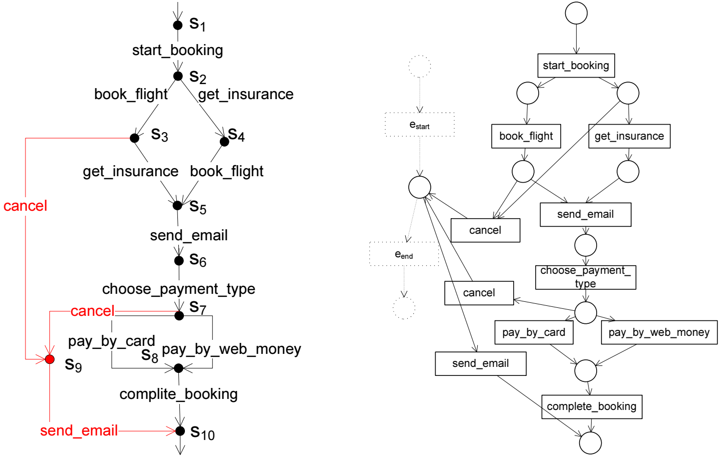


State-based regions miner



ILP miner

Motivating example. Decomposition approach



Basic definitions

Definition (Event log). Let E be a set of events. A *trace* σ (over E) is a sequence of events, i.e., $\sigma \in E^*$. An *event log* L is a multiset of traces.

Definition (Transition system). A *transition system* is a tuple $TS = (S, E, B, s_{init}, S_{fin})$, where

- S - is a set of states;
- E - is a set of events;
- $B \subseteq S \times E \times S$ is a *transition relation*;
- $s_{init} \in S$ is an initial state;
- and $S_{fin} \subseteq S$ — a set of final states.

Basic definitions

We write $s \xrightarrow{e} s'$, when $(s, e, s') \in B$.

A state s is *reachable* from a state s' iff there is a possibly empty sequence of transitions leading from s to s' (denoted by $s \xrightarrow{*} s'$).

A transition system must satisfy the following basic axioms: every state is reachable from the initial state: $\forall s \in S : s_{init} \xrightarrow{*} s$, for every state there is a final state, which is reachable from it: $\forall s \in S \exists s_{fin} \in S_{fin} : s \xrightarrow{*} s_{fin}$.

Definition (Language accepted by transition system).
A trace $\sigma = \langle e_1, \ldots, e_n \rangle$ is called *feasible* in a transition system TS iff $s_{init} \xrightarrow{*} s_n$, and $s_n \in S_{fin}$, i.e. a *feasible* trace leads from the initial state to some final state. A *language accepted by* TS is defined as the set of all traces feasible in TS, and is denoted by $L(TS)$.

Basic definitions

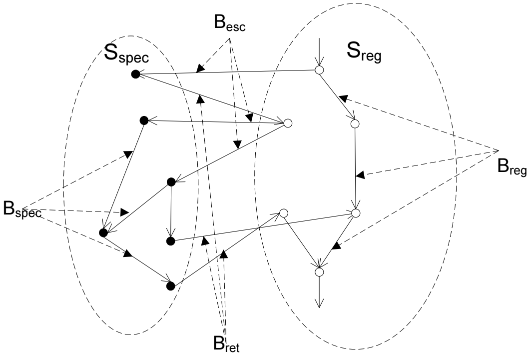
Definition (Reachability graph)

A *reachability graph* for a marked Petri net $(N, m_0), N = (P, T, F, \lambda)$ labeled with events from E is a transition system $TS = (S, E, B, s_{init}, S_{fin})$, such that

- the set of states S defined as a set of all markings reachable in N from the marking m_0,
- transition relation B defined by $(m, e, m') \in B$ iff $m \xrightarrow{t} m'$, where $e = \lambda(t)$,
- the initial state in TS is the initial marking m_0,
- if some reachable markings in (N, m_0) are distinguished as final markings, they are defined as final elements in TS.

Decomposition approach

Definition (Decomposed transition system).
A set of states S can be partitioned over S_{reg} and S_{spec} and then the tuple $TS_{dec} = (TS, S_{reg}, S_{spec})$ is called a *decomposed transition system* if the following additional conditions hold: $s_{in} \in S_{reg}$ and $S_{fin} \subseteq S_{reg}$.



Region algorithm. Properties

1. There is a homomorphism ω from a transition system TS to the reachability graph RG of a target Petri net N, i.e. for every $s \in S$ there is a corresponding node - $\omega(s)$ in RG (every state in TS has a corresponding N marking), such that $\omega(s_{in}) = m_0$ and for every transition $(s_1 \xrightarrow{e} s_2) \in B$ there is an arc $(\omega(s_1), \omega(s_2))$ in RG labeled with e.

2. The target Petri net is safe.

Decomposition algoritm. Step 1

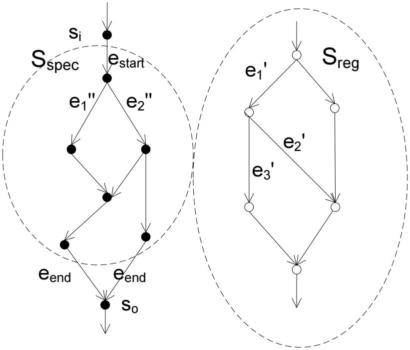
Step 1. For the decomposed transition system:

$TS_{dec} = ((S, E, B, s_{in}, S_{fin}), S_{reg}, S_{spec})$ construct two transition systems:

$TS_{reg} = (S_{reg}, E_{reg} \cup E', B_{reg} \cup B', s_{in}, S_{fin})$

and

$TS_{spec} = (S_{spec} \cup \{s_i\} \cup \{s_o\}, E_{spec} \cup E'' \cup \{e_{start}, e_{end}\}, B_{spec} \cup B'' \cup B_{start} \cup B_{end}, s_i, \{s_o\})$

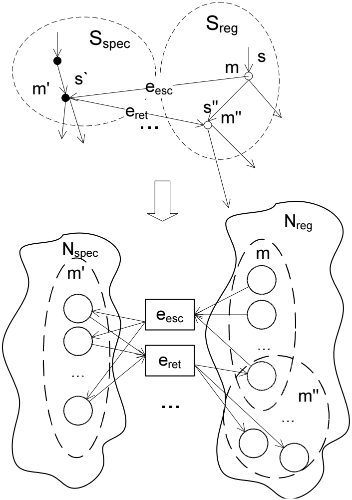


Decomposition algoritm. Steps 2, 3 and 4

Step 2. Apply a region-based algorithm to retrieve a regular (N_{reg}) and a special (N_{spec}) process flow from TS_{reg} and TS_{spec} respectively.

Step 3. Restore the connections between N_{reg} and N_{spec} to create a unified Petri net N: $\forall(s \overset{e_{esc}}{\rightarrow} s') \in B_{esc}, s \in S_{reg}, s' \in S_{spec}$ add a novel Petri net transition labeled with e_{esc}. Similarly, transitions from B_{ret} are restored.

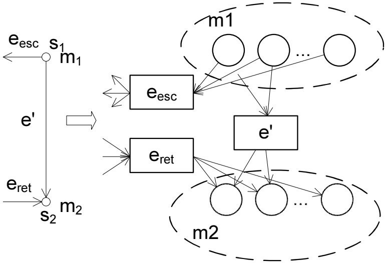
Step 4. Delete the following nodes along with their incident arcs: transitions with labels from E' and E'', all the transitions labeled with e_{start} along with the places from their presets, and all the transitions labeled with e_{end} along with the places from their postsets.



Structural properties

Lemma (Connectivity properties)

If there is a *directed path* between a pair of nodes (i.e., places or transitions) within subprocess model N_{reg} (N_{spec}) and these nodes were not deleted during the construction of the unified Petri net model $N = (P, T, F, \lambda)$, then there is a directed path between them within N.



Theorem (WF-nets)

Assume that subprocess models N_{reg} and N_{spec} are WF-nets. Then the unified process model N is a WF-net.

Behavioural properties

Theorem (Bisimulation)

Suppose there are bisimulation relations between transition systems TS_{reg} and TS_{spec}, corresponding to subprocess models N_{reg} and N_{spec}, and reachability graphs RG_{reg}, RG_{spec} of these subprocess models. Suppose also that there are no states in TS_{reg} and TS_{spec}, which correspond to markings dominating one another. Then there is a bisimulation relation between a decomposed transition system and the reachability graph RG of the unified process model N.

Theorem (Language inclusion)

Let $TS_{dec} = (TS, S_{reg}, S_{spec})$ be a decomposed transition system. Assume that N_{reg} and N_{spec} are subprocess models constructed from transition systems TS_{reg} and TS_{spec} with corresponding reachability graphs RG_{reg} and RG_{spec}. Let Petri net N be the unified process model with a reachability graph RG. If $L(TS_{reg}) \subseteq L(RG_{reg})$ and $L(TS_{spec}) \subseteq L(RG_{spec})$, then there is a language inclusion: $L(TS) \subseteq L(RG)$.

A language based region algorithm. Properties

A (language-based) region is defined as a $(2|T|+1)$-tuple of $\{0,1\}$, representing an initial marking and a number of tokens each transition consumes and produces in a place.

1. There is a bijection between Petri net transitions T and events in the initial transition system E, such that every transition is labeled with a corresponding event.

2. There is a homomorphism ω from a transition system TS to the reachability graph RG of a target Petri net N.

3. The target Petri net is safe.

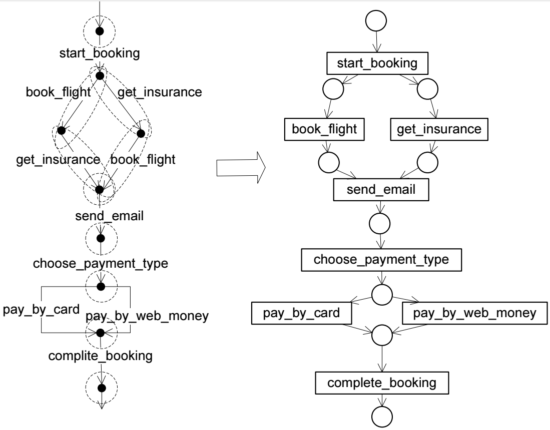
State-based region algorithm

Let $TS = (S, E, T, s_{in}, S_{fin})$ and $S' \subseteq S$. S' is a *region* iff $\forall e \in E$ one of the following conditions hods:

- all the transitions $s_1 \xrightarrow{e} s_2$ **enter** S', i.e. $s_1 \notin S'$ and $s_2 \in S'$,

- all the transitions $s_1 \xrightarrow{e} s_2$ **exit** S', i.e. $s_1 \in S'$ and $s_2 \notin S'$,

- all the transitions $s_1 \xrightarrow{e} s_2$ **do not cross** S', i.e. $s_1, s_2 \in S'$ or $s_1, s_2 \notin S'$.



State-based region algorithm. Properties

1. Every Petri net transition $t \in T$ corresponds to an event in the initial transition system $e \in E$ (the transition t is labeled with e), the opposite is not true (events of the initial transition system might be split).

2. There is a bisimulation between a transition system and a reachability graph of the target Petri net, this implies that every state in TS corresponds to a Petri net marking. Bisimulation relation defined by the state-based region algorithm specifies the only one state in a reachability graph for each state in a transition system, in such a case bisimulation implies homomorphism.

3. The target Petri net is safe, i.e. no more than one token can appear in a place.

Conclusions. Future plans

1. The method presented can be applied to construct a hierarchy of subprocesses.
2. The decomposition method gives a starting point for more advanced mining techniques.
3. Some improvements (including reset arcs and parallel executions) can be proposed.
4. A method for automatization of a decomposition should be proposed.
5. Methods for transition system repair could be developed.
6. The method can applied for the construction subprocesses of a hight-level process model (e.g. BPMN process model).

Thank you!