AI Meets Declarative Process Mining: A Concrete Initiative to Move from Theory to Practice

Fabrizio M. Maggi
maggi@inf.unibz.it

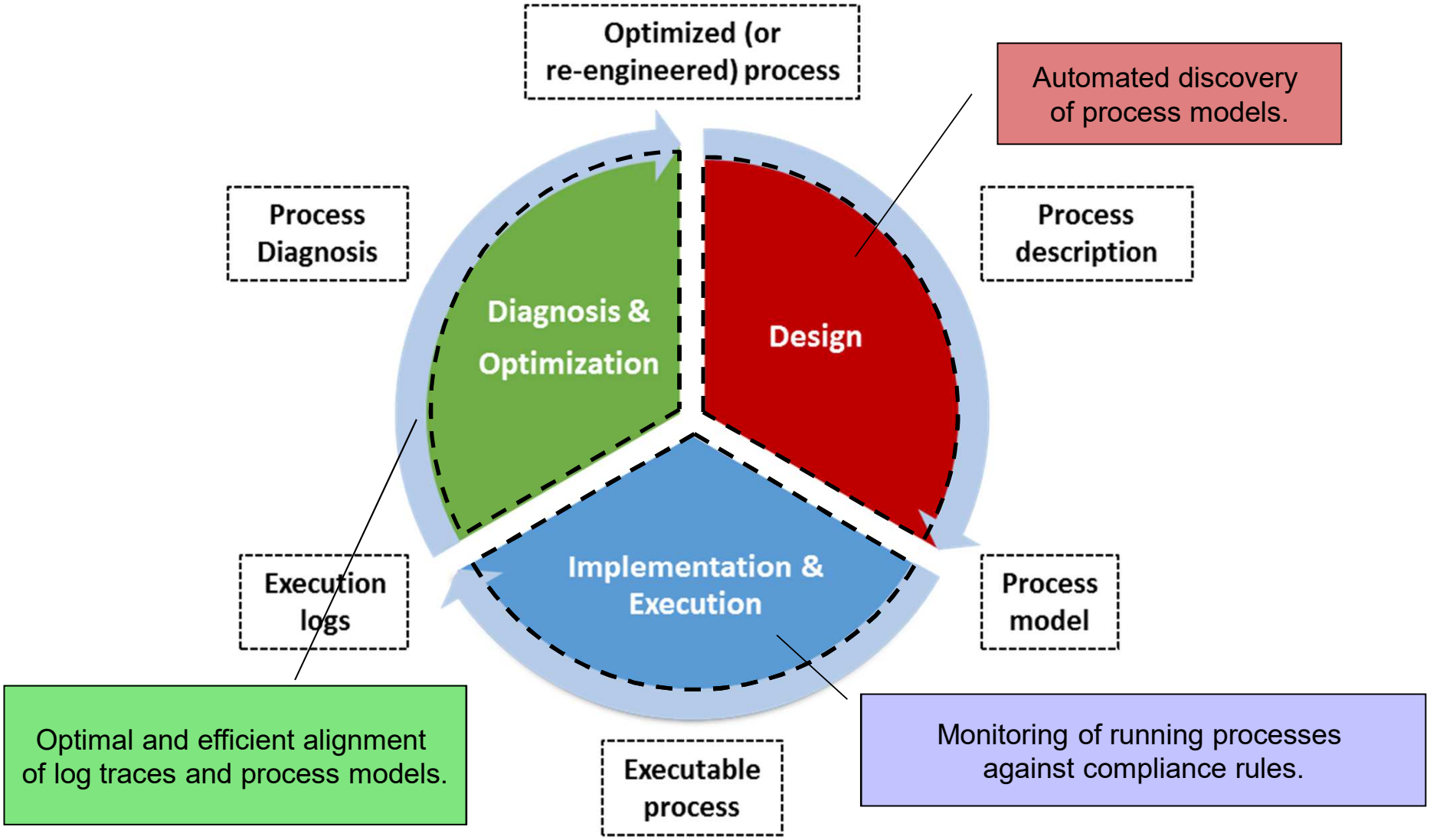
Process Intelligence

Process mining: Application of **data science** to discover, validate and improve processes. **By combining data mining and process analytics**, organizations can mine log data from their information systems to **understand the performance** of their processes, **revealing bottlenecks** and other **areas of improvement**.

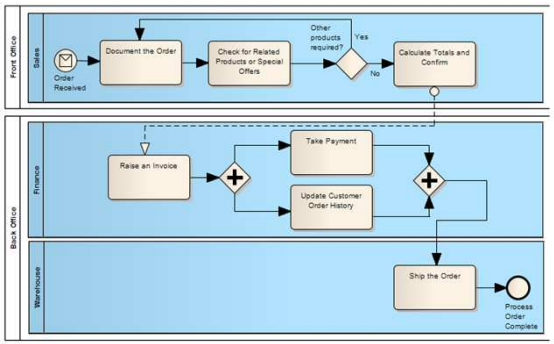
Wil M. P. van der Aalst: ***Process Mining - Data Science in Action, Second Edition***. Springer 2016, ISBN 978-3-662-49850-7

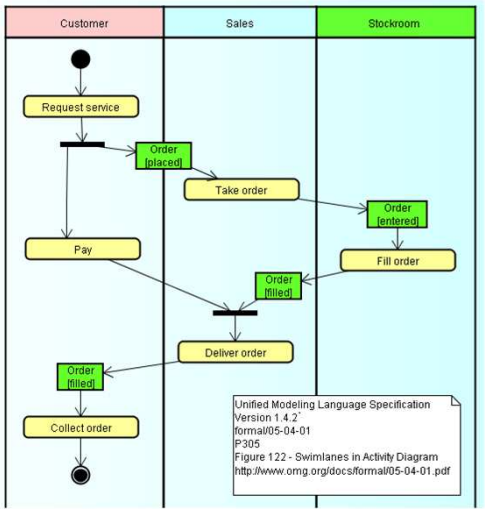
Process Intelligence: Application of **AI for developing process mining techniques** to **strengthen** *designing*, *monitoring* and *diagnosis* of BPs during their **life-cycle**.

Process Intelligence in the BPM Lifecycle

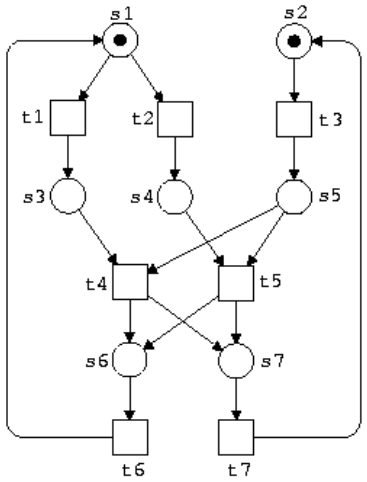


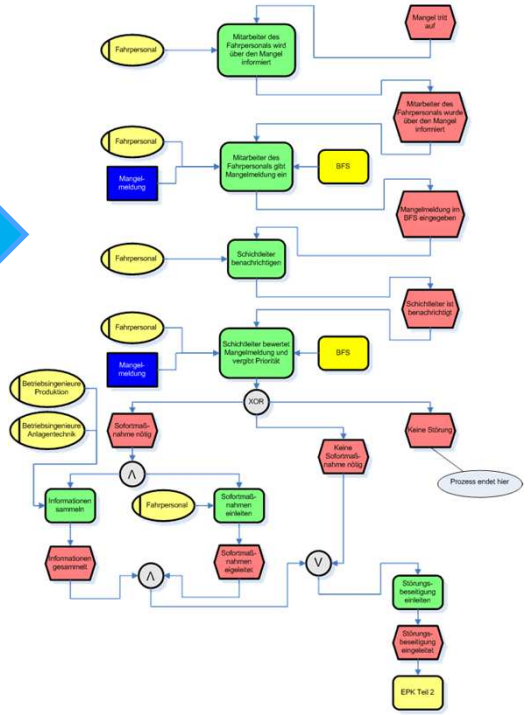
Imperative Process Models





Process Models





Declare

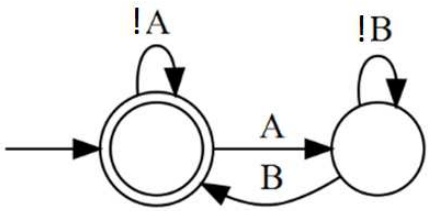
- *A is always eventually followed by B*
- RESPONSE
- Graphical representation

Maja Pesic, Helen Schonenberg, Wil M. P. van der Aalst: ***DECLARE: Full Support for Loosely-Structured Processes***. EDOC 2007: 287-300

- Semantics specified through LTL (for finite traces)

$$\Box(A \Rightarrow \Diamond B)$$

- LTL rules can be translated into automata



Giuseppe De Giacomo, Moshe Y. Vardi: ***Linear Temporal Logic and Linear Dynamic Logic on Finite Traces***. IJCAI 2013: 854-860

Roadmap

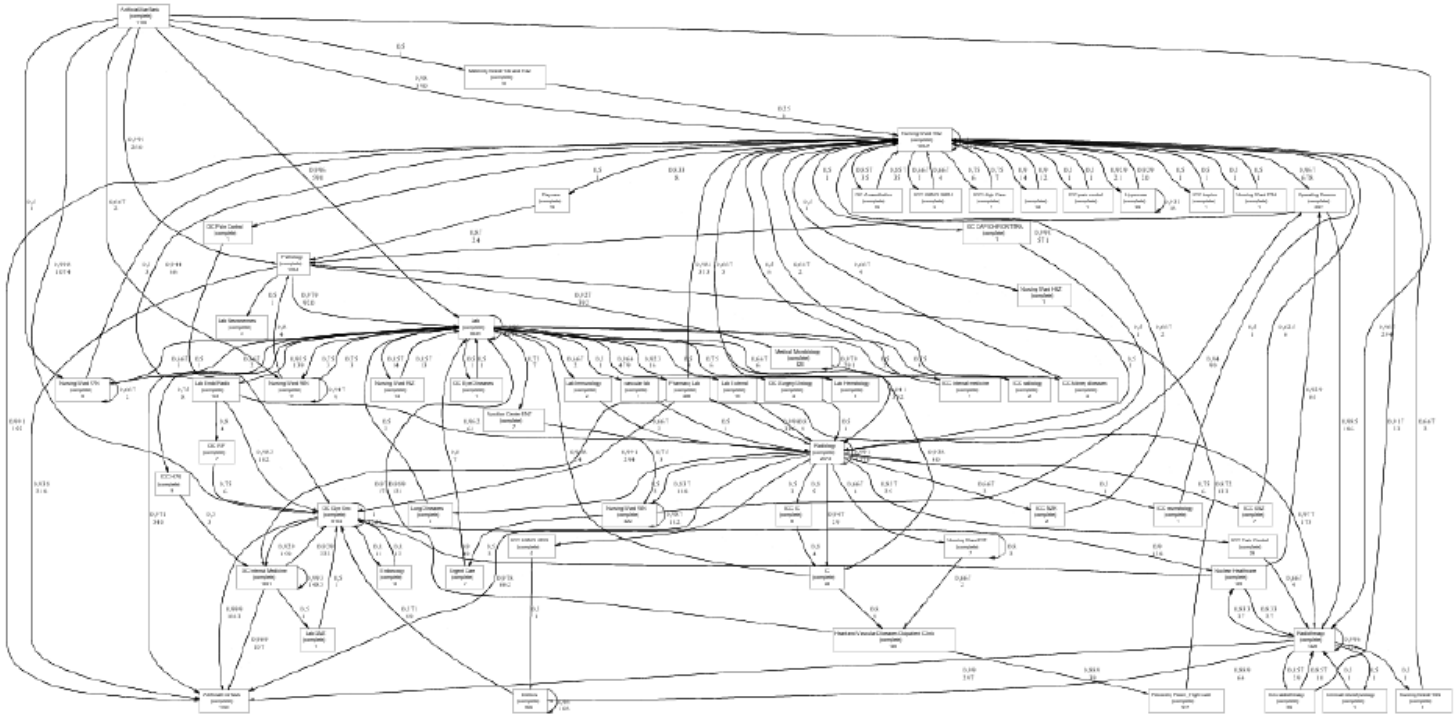
Automated ***Process Discovery***

Compliance Monitoring

Conformance Checking

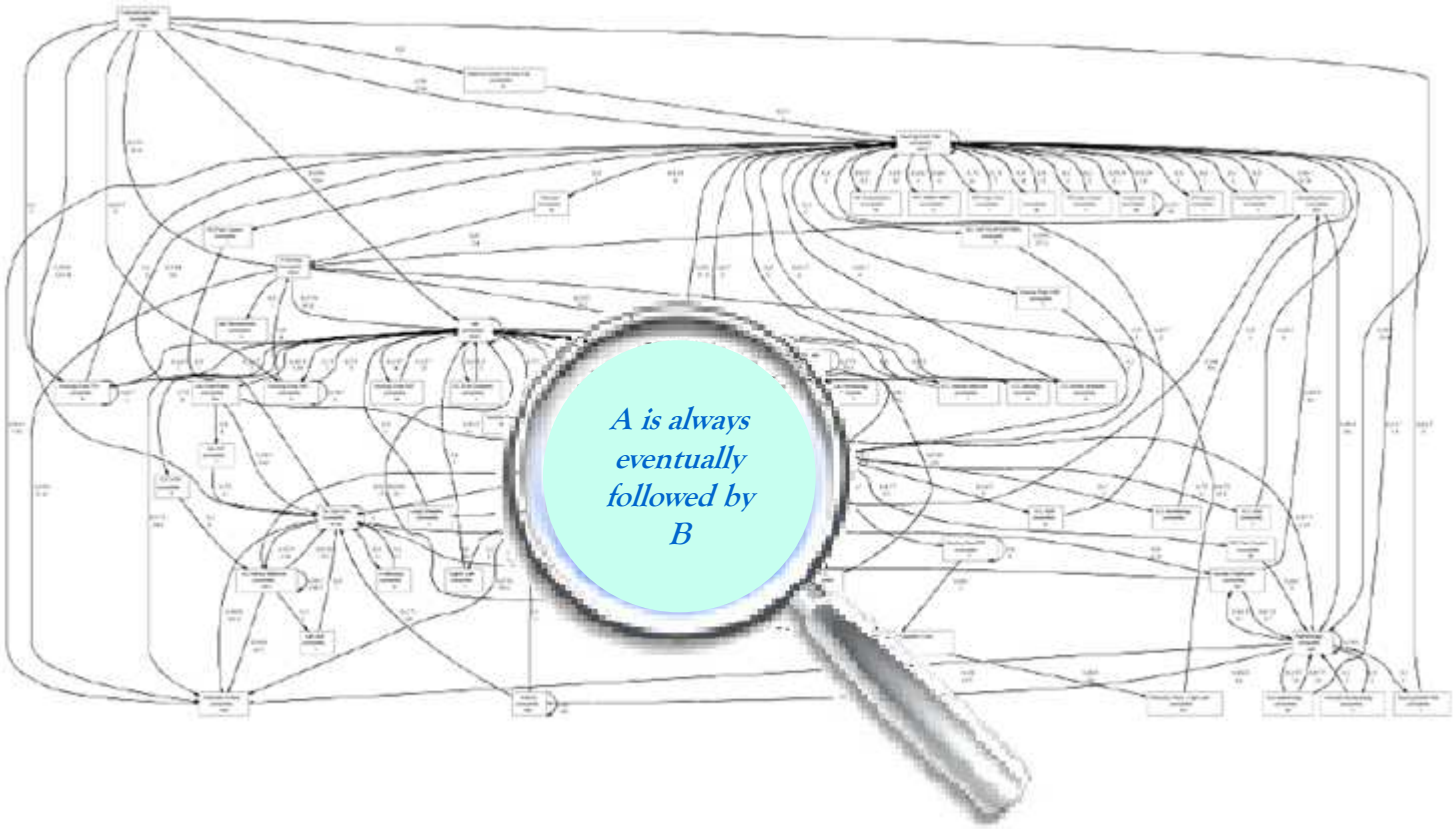
Log Generation

Imperative Process Discovery

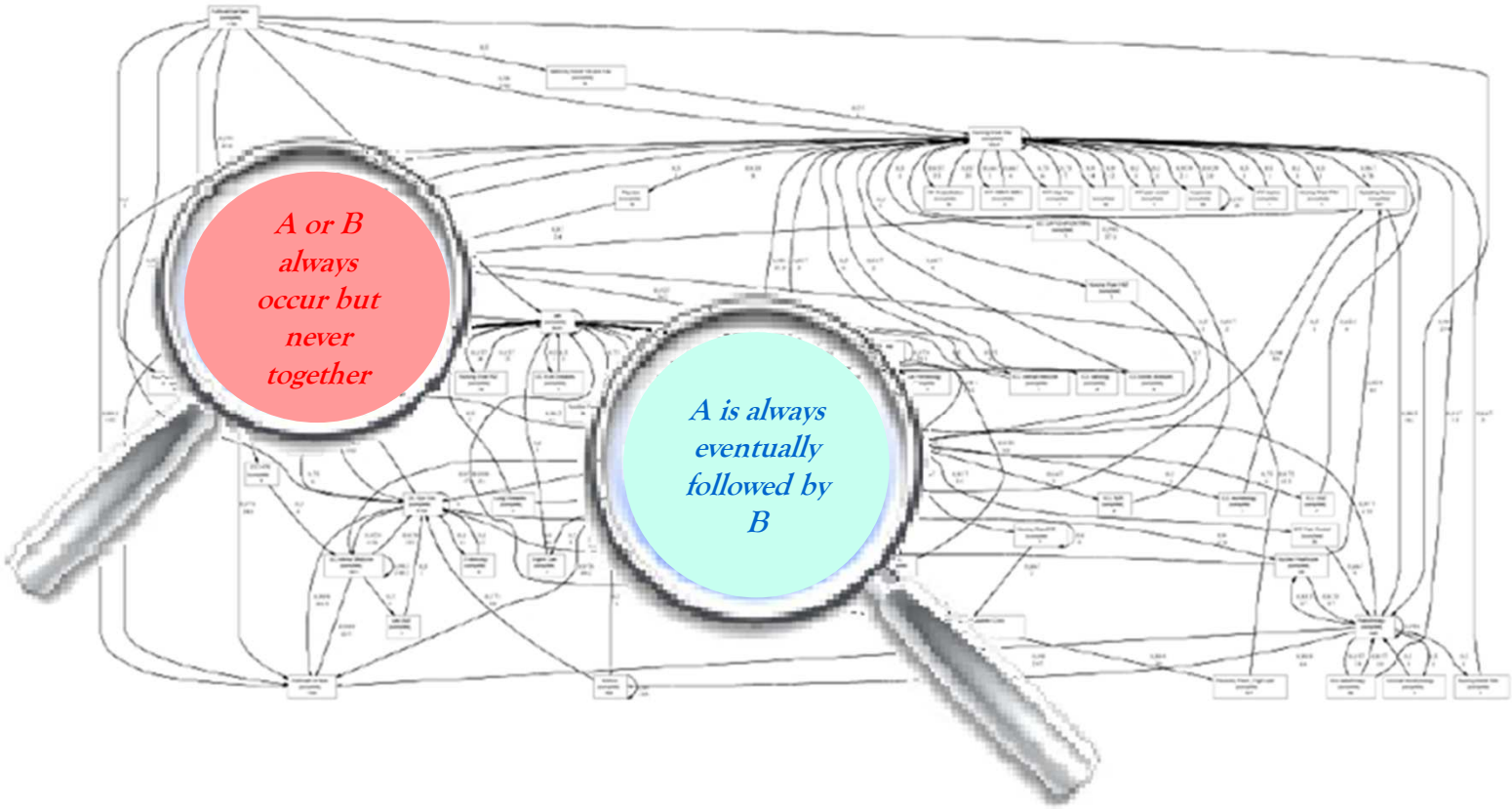


Imperative Process Discovery

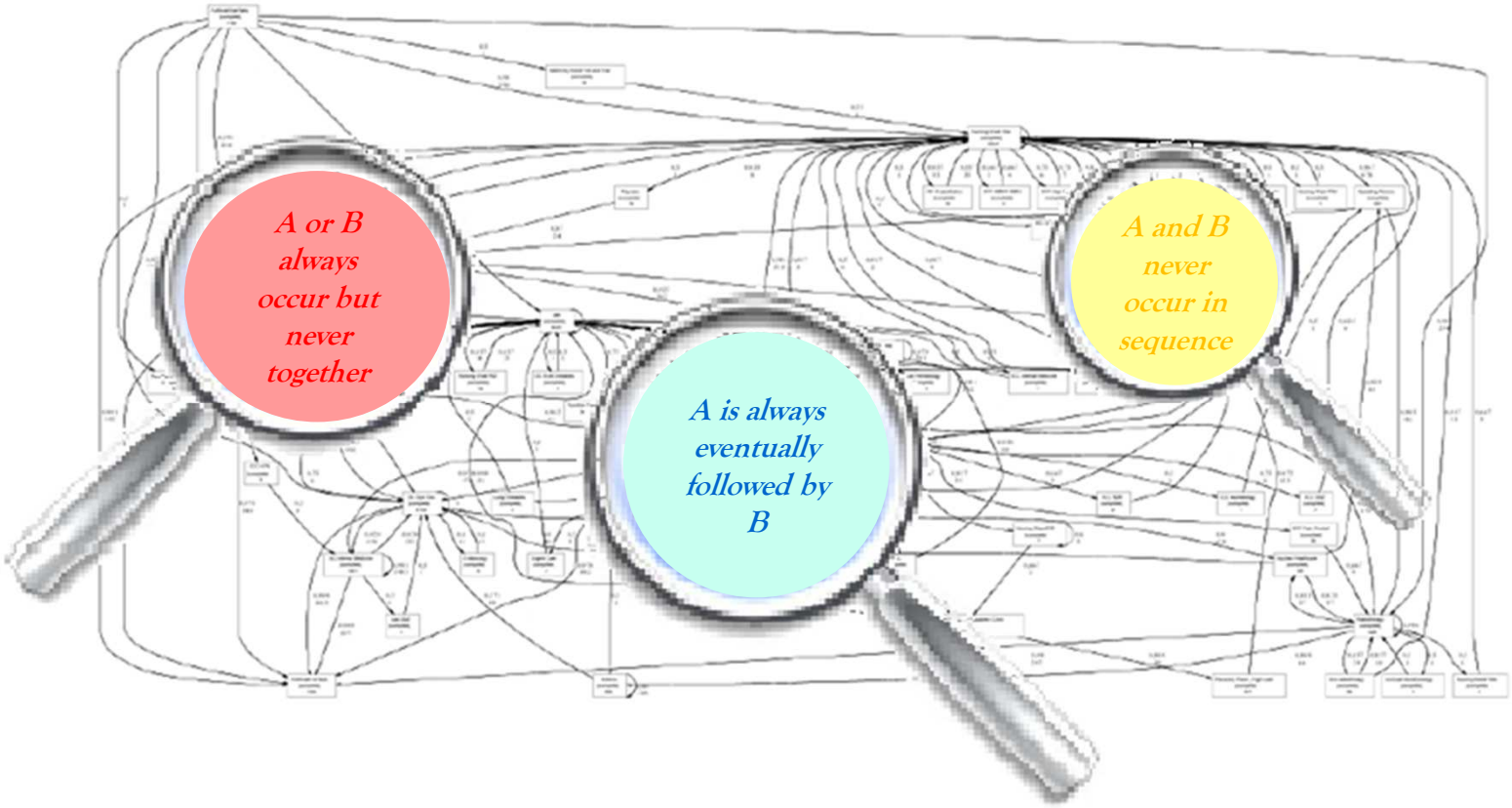
Declarative Process Discovery



Declarative Process Discovery

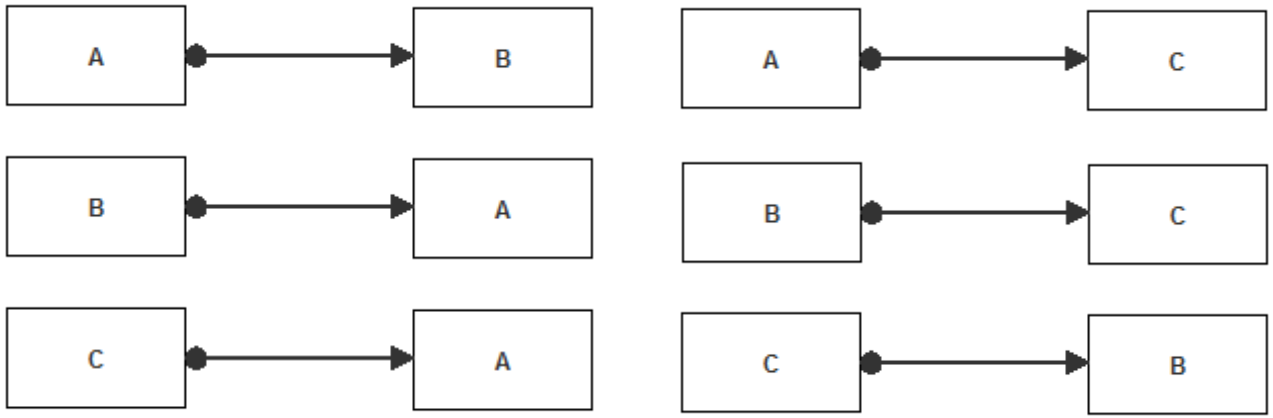


Declarative Process Discovery



Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]

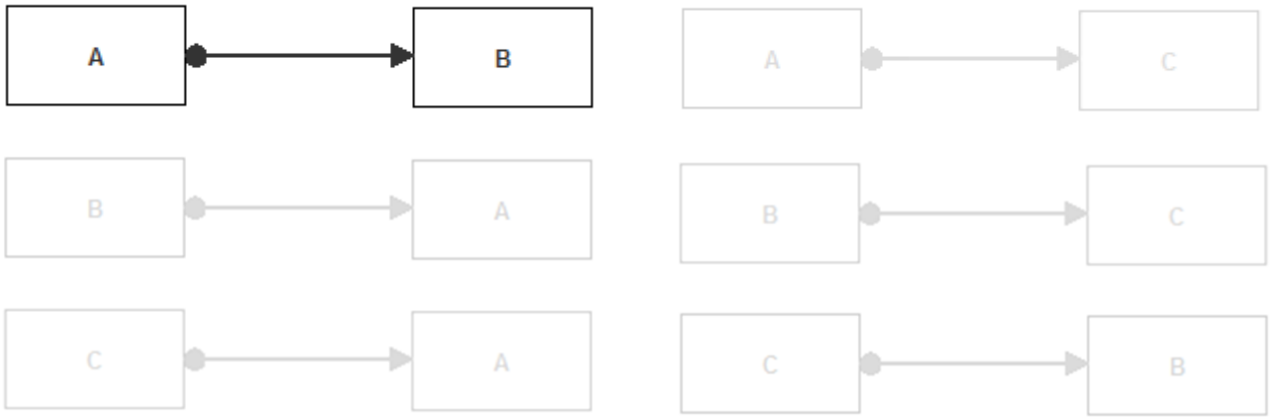


- finite number of constraint types
- finite set of activities

Fabrizio M. Maggi, R. P. Jagadeesh Chandra Bose, Wil M. P. van der Aalst: ***Efficient Discovery of Understandable Declarative Process Models from Event Logs***. CAiSE 2012: 270-285

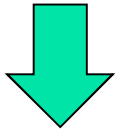
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



Discovery algorithm

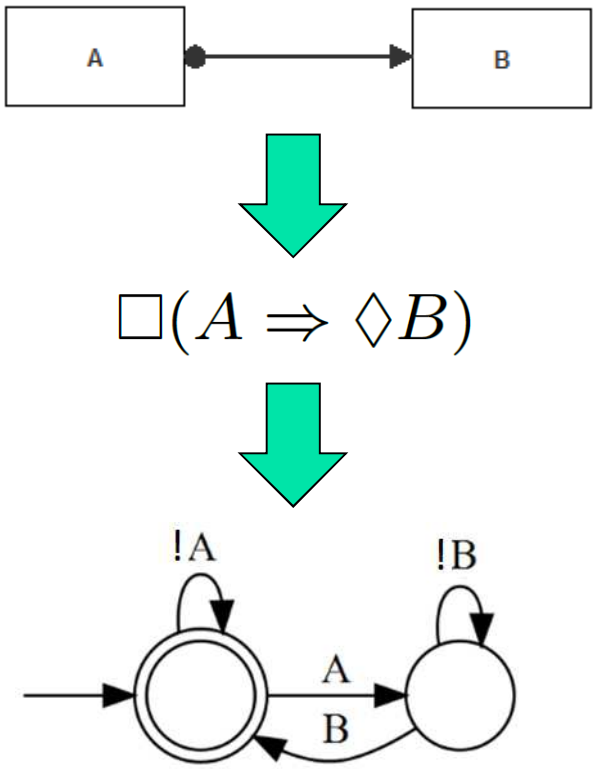
W = [<A C B C>, <C B A C>, <A C A C A C B>]



$$\Box(A \Rightarrow \Diamond B)$$

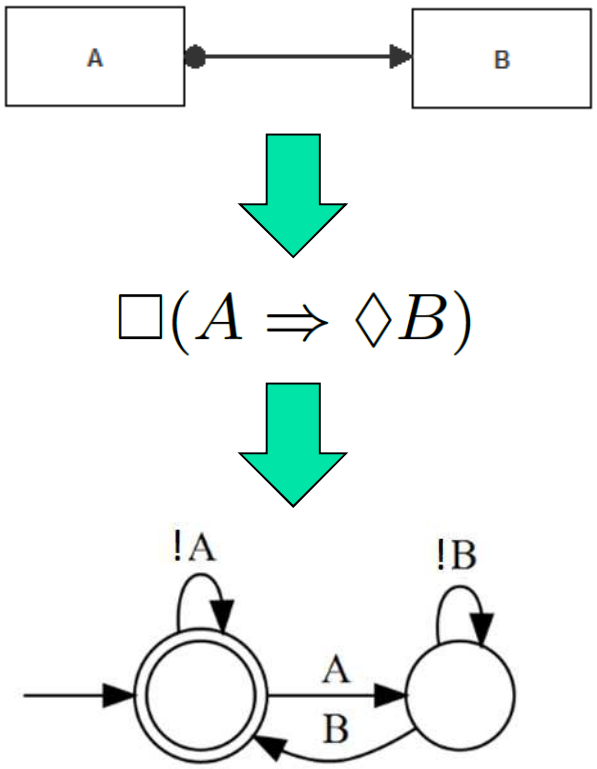
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



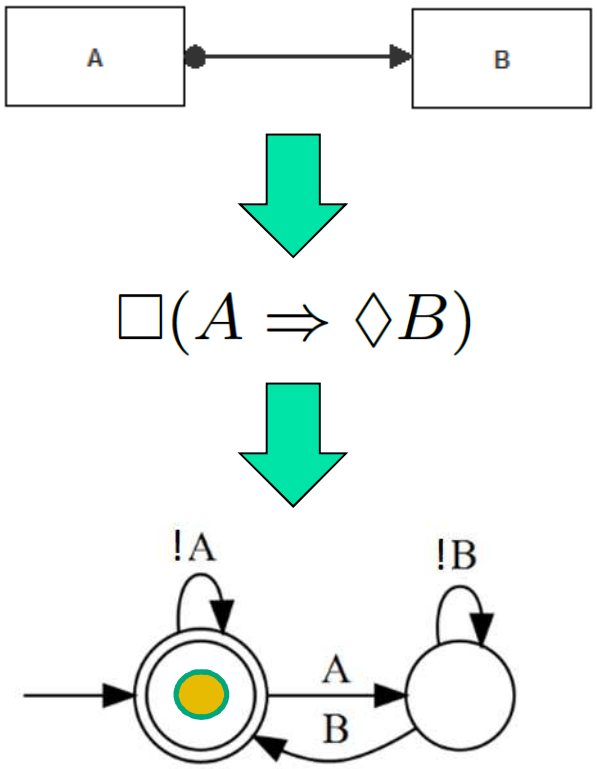
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



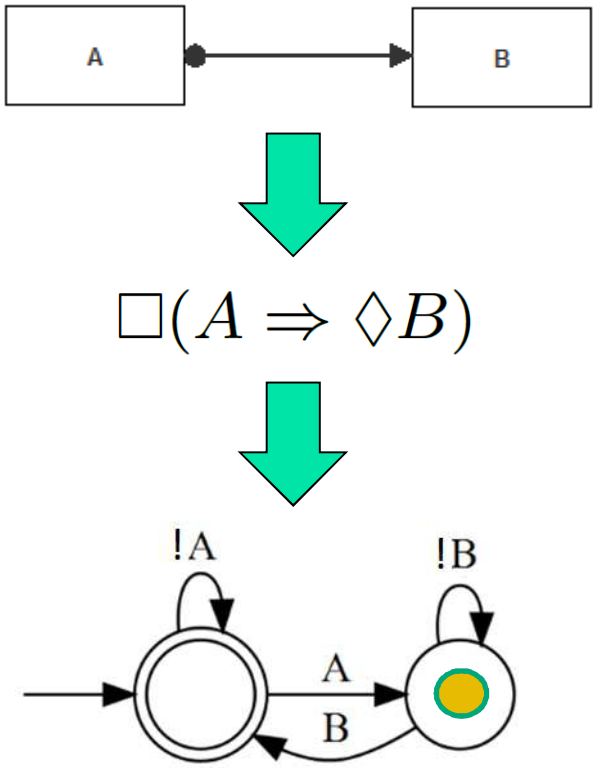
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



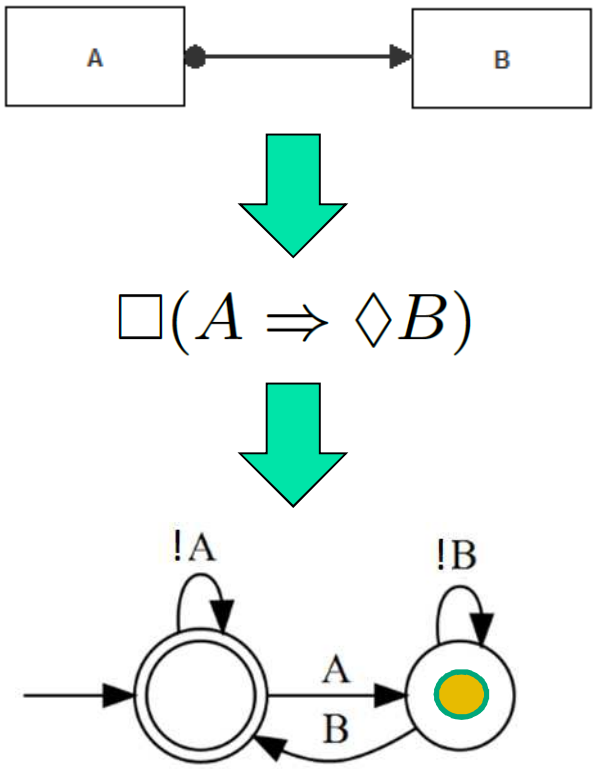
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



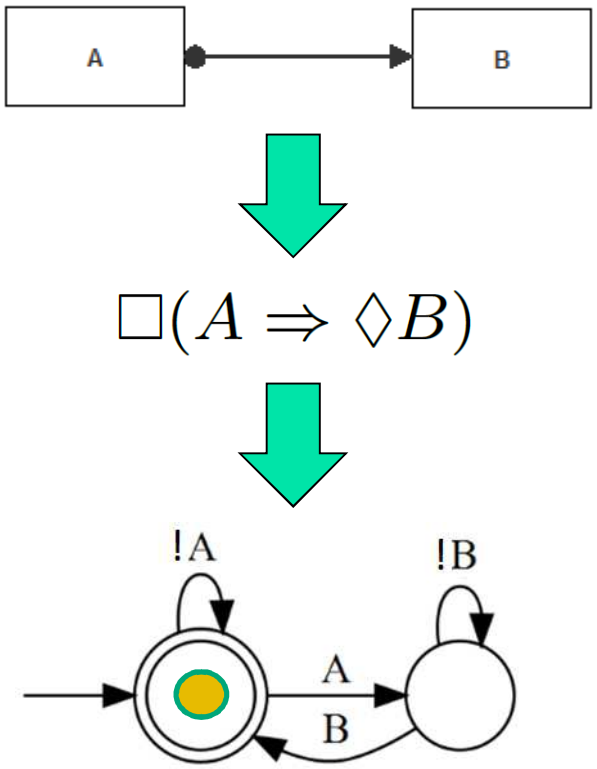
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



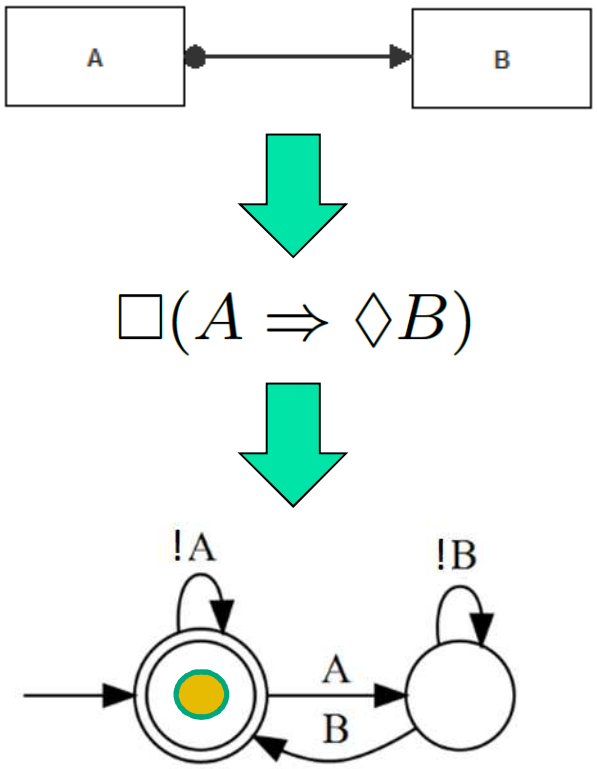
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



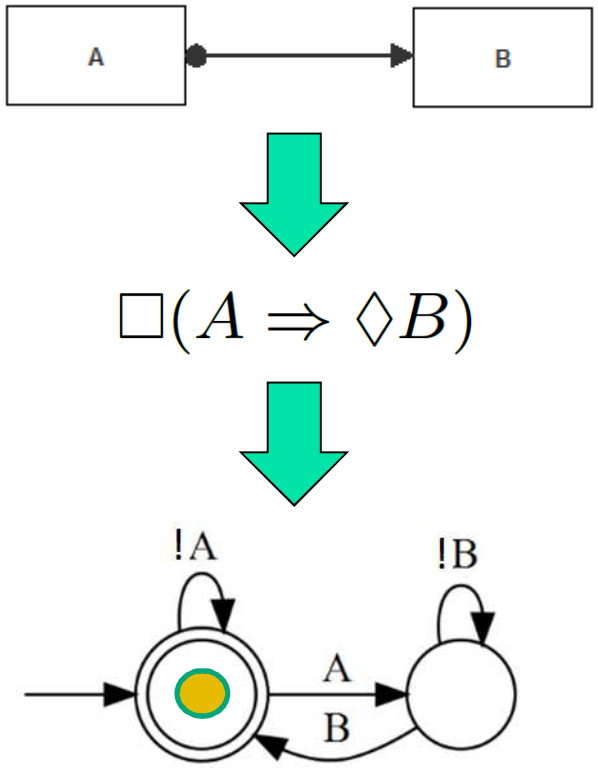
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



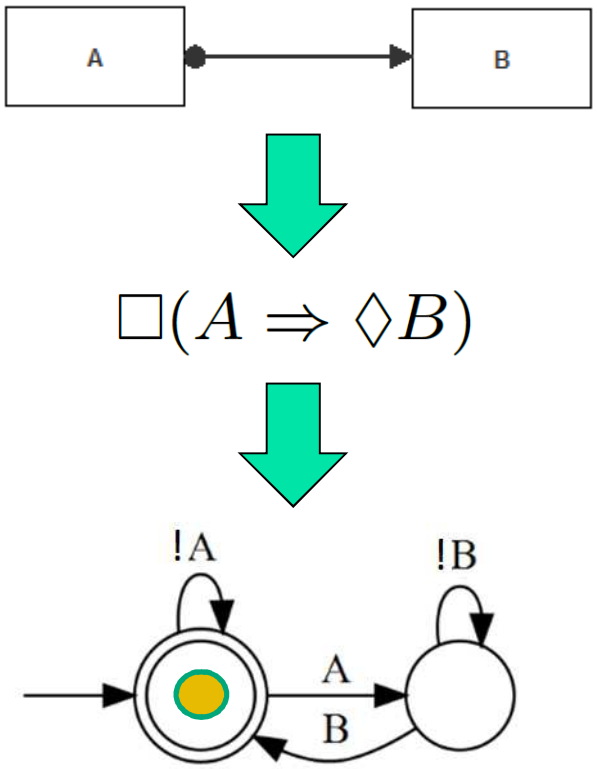
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



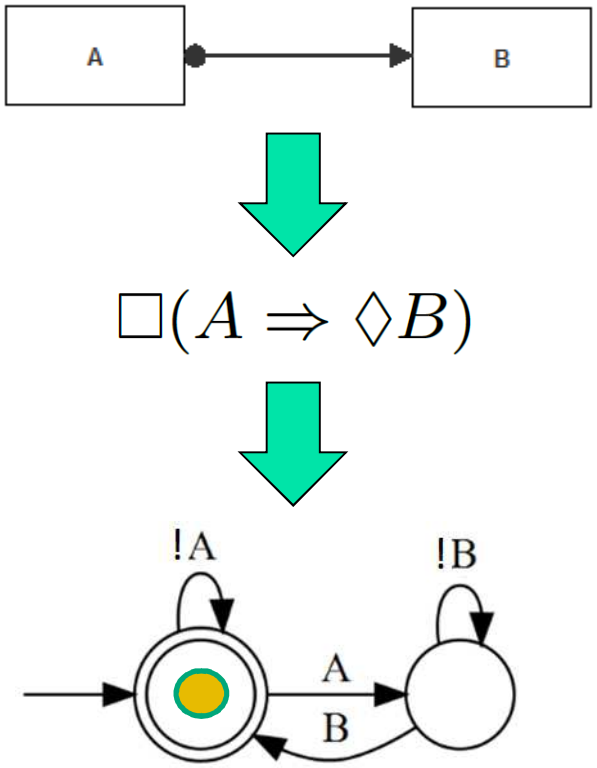
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



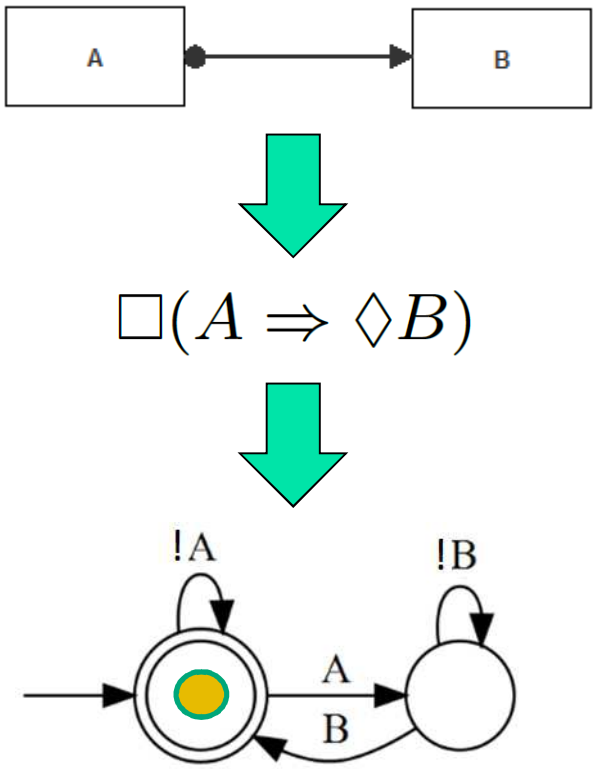
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



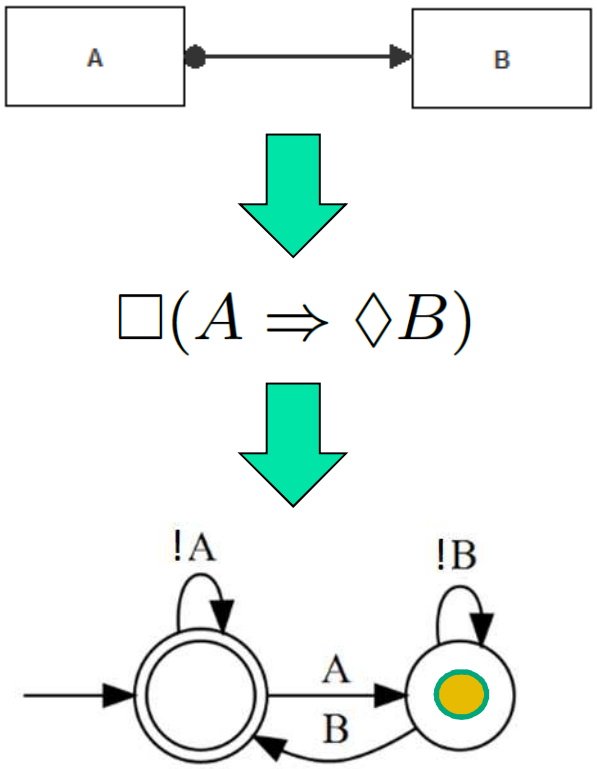
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



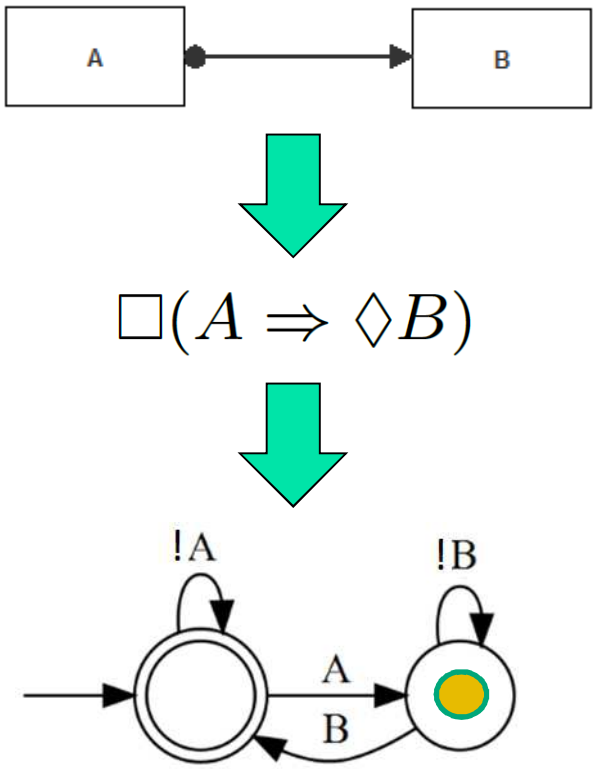
Discovery algorithm

W = [<A C B C>, <C B A C>, <A C A C A C B>]



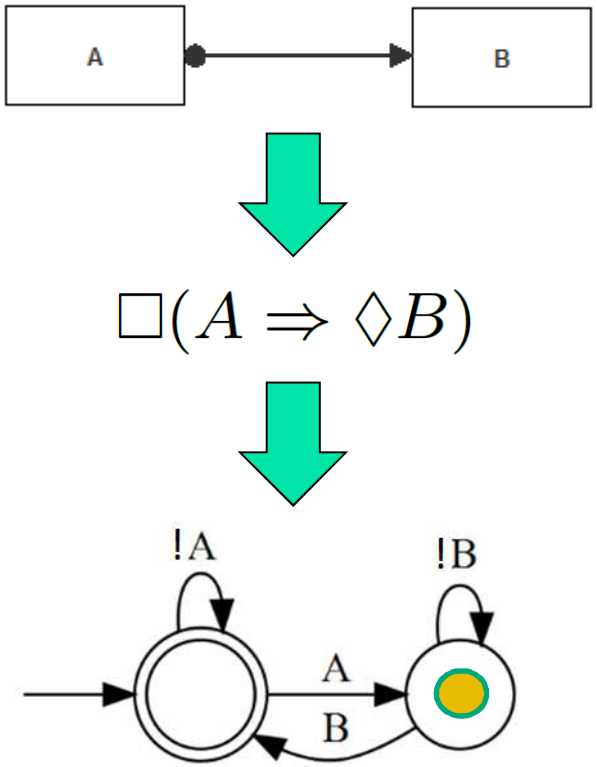
Discovery algorithm

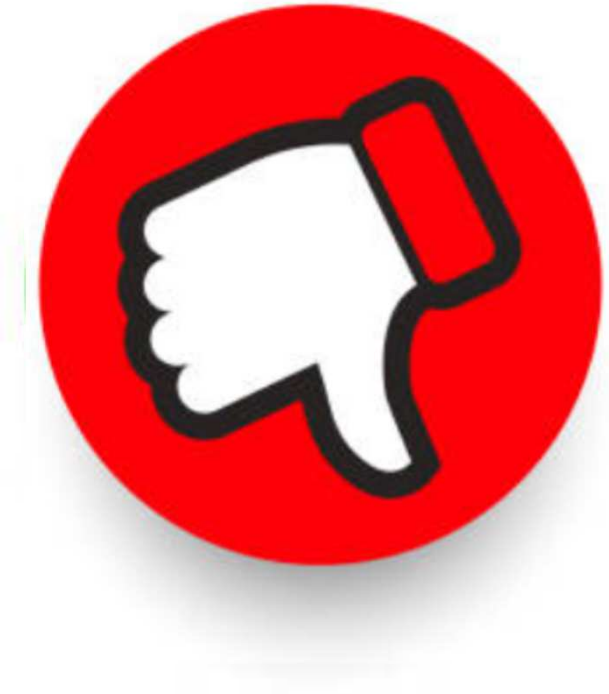
W = [<A C B C>, <C B A C>, <A C A C A C B>]



Discovery algorithm

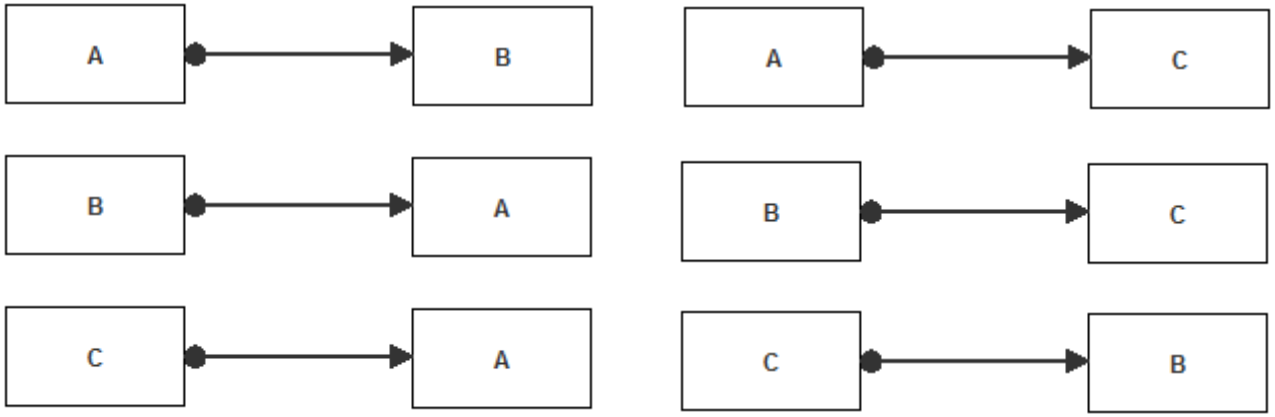
W = [<A C B C>, <C B A C>, <A C A C A C B>]





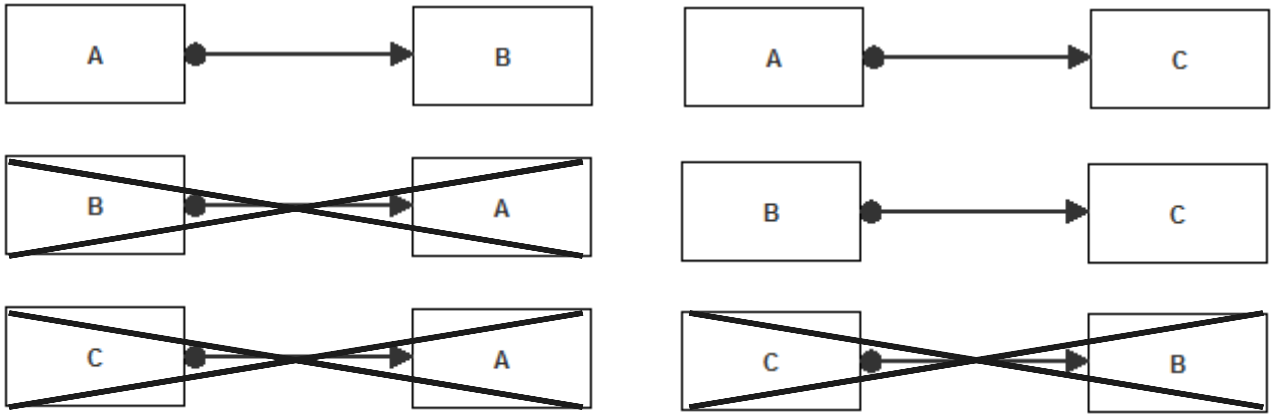
Discovery algorithm: Pruning

W = [<A C B C>, <C B A C>, <A C A C A C B>]



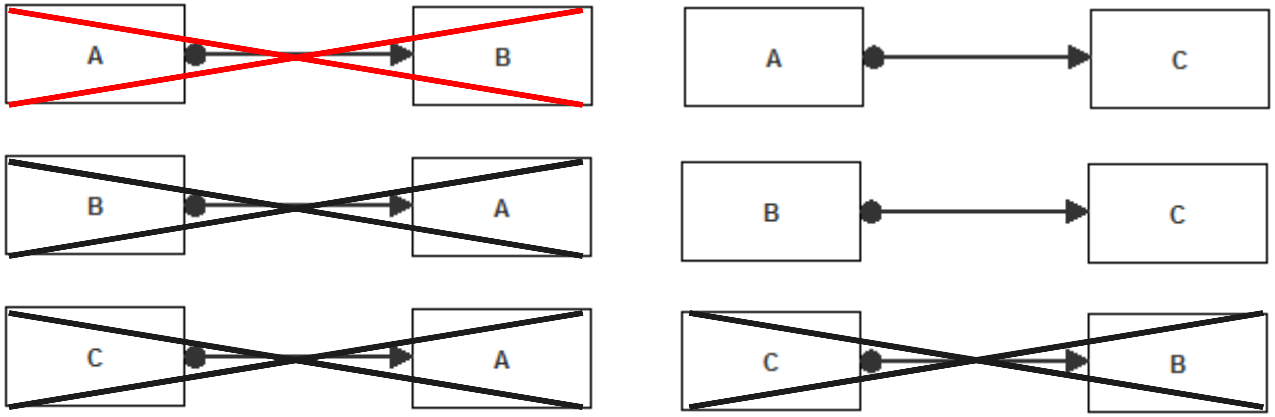
Discovery algorithm: Pruning

W = [<A C B C>, <C B A C>, <A C A C A C B>]



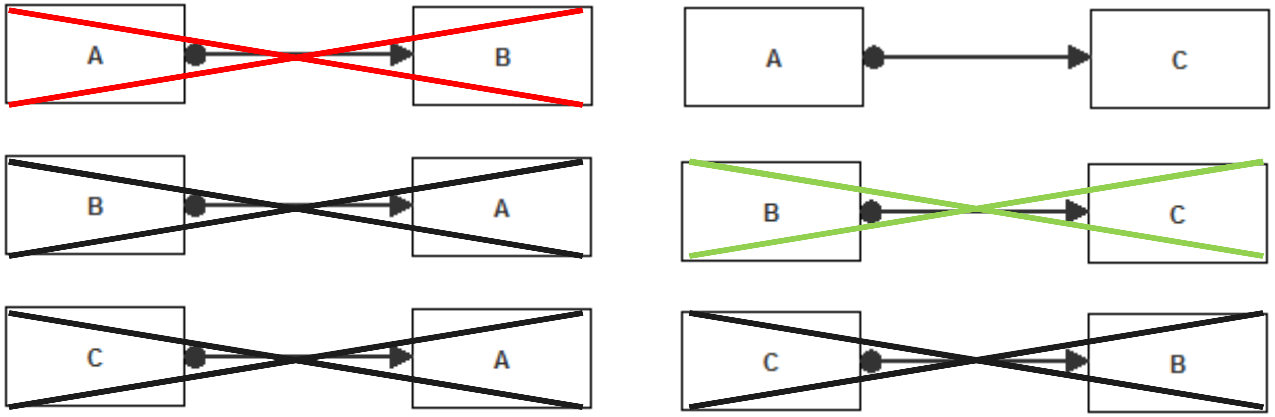
Discovery algorithm: Pruning

W = [<A C B C>, <C B A C>, <A C A C A C B>]



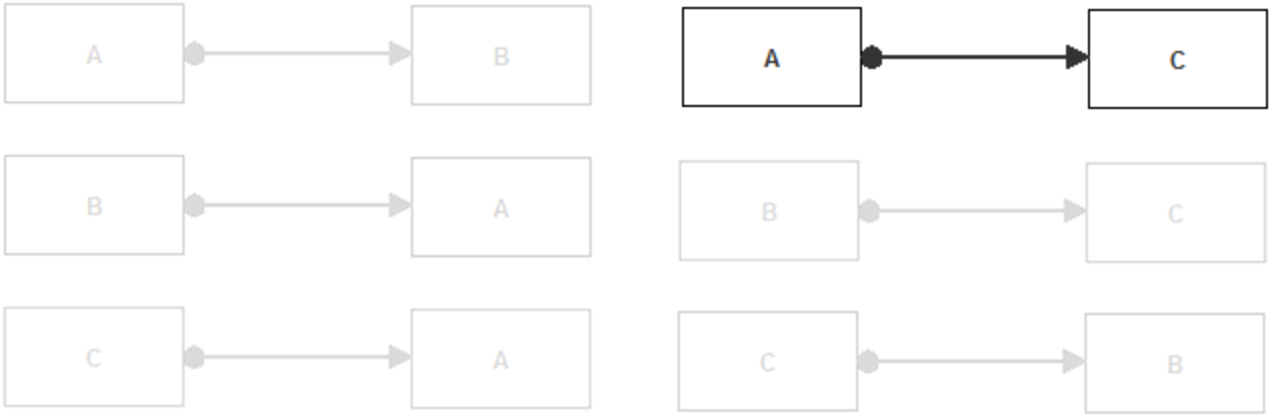
Discovery algorithm: Pruning

W = [<A C B C>, <C B A C>, <A C A C A C B>]



Discovery algorithm: Pruning

W = [<A C B C>, <C B A C>, <A C A C A C B>]



Process Discovery

rulemining.org

Roadmap

Automated Process Discovery

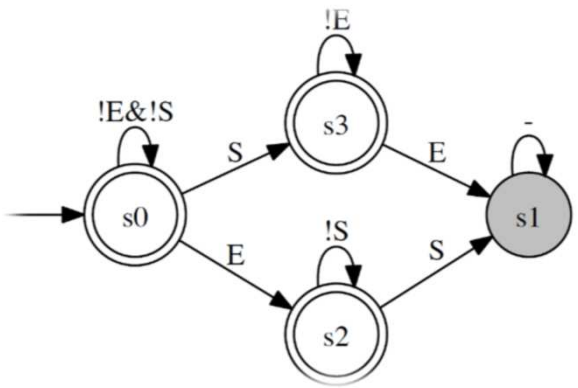
Compliance Monitoring

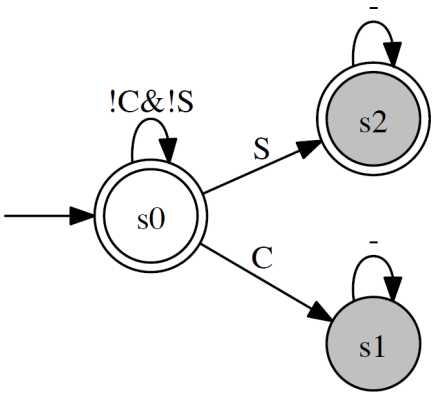
Conformance Checking

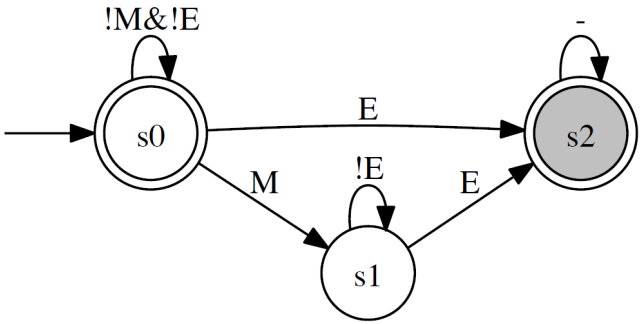
Log Generation

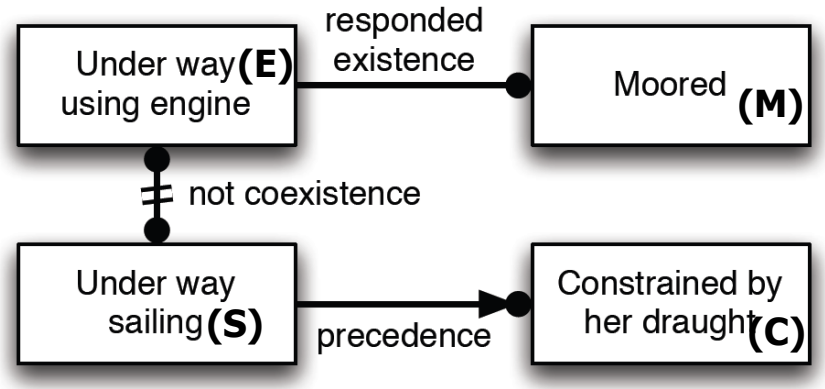
Compliance

Compliance Models

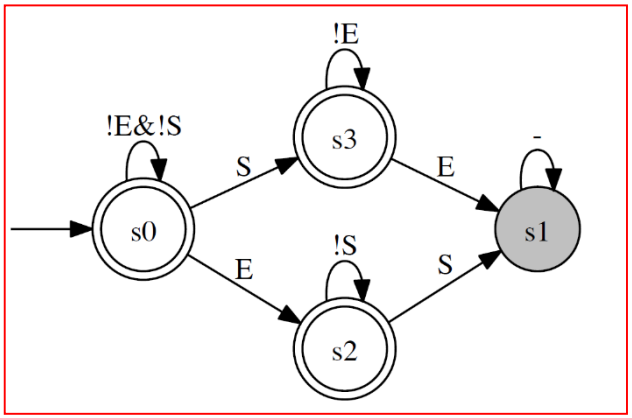


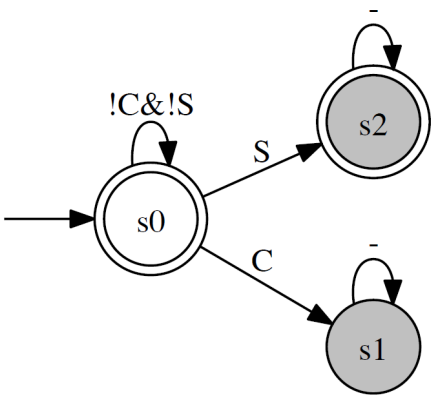


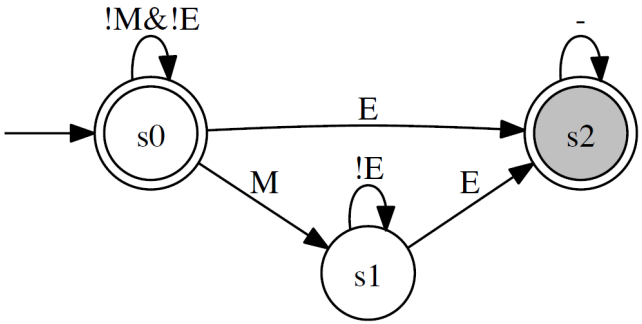


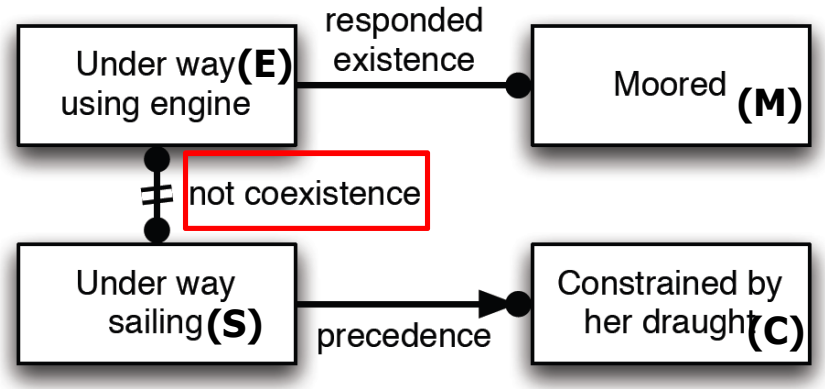


Compliance Models

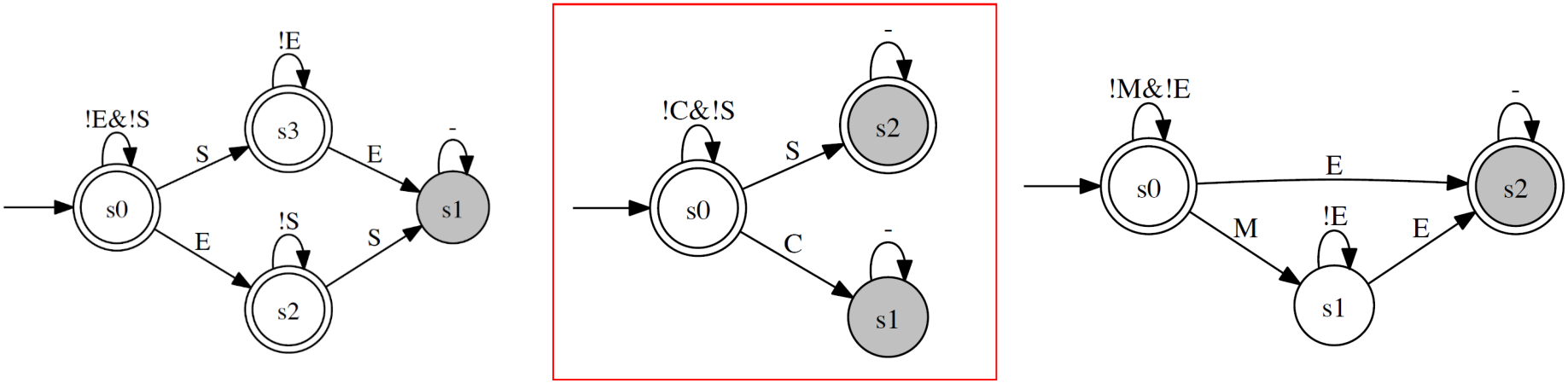




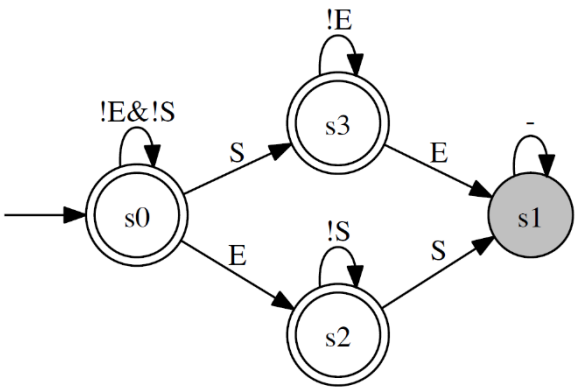


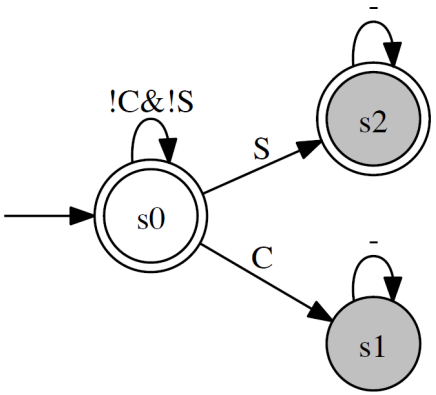


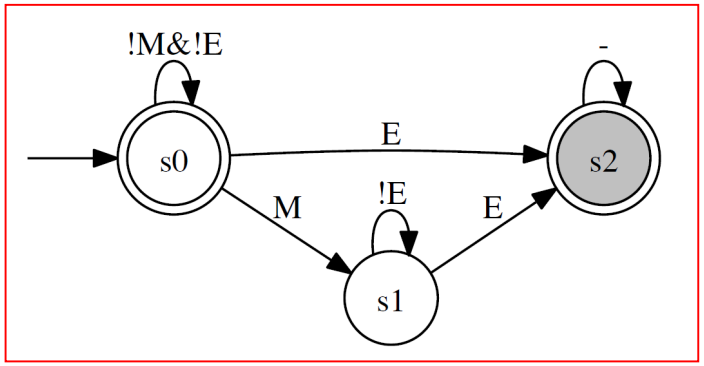
Compliance Models

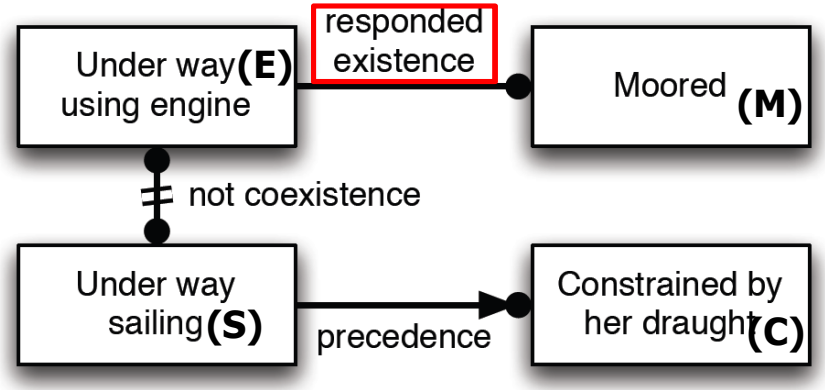


Compliance Models





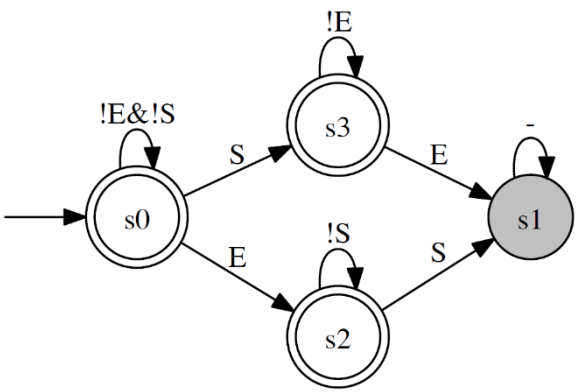


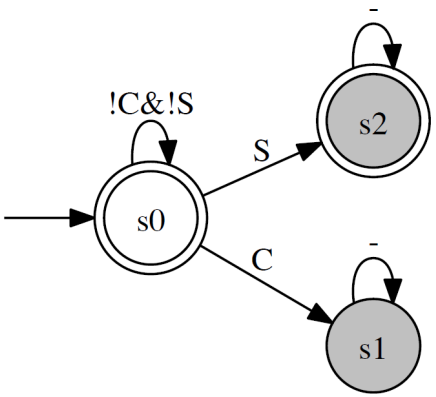


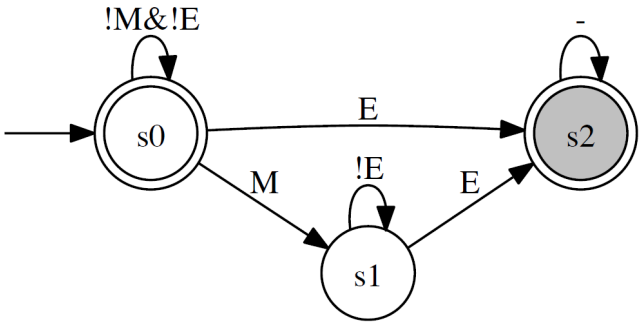
RV-LTL Semantics

Andreas Bauer, Martin Leucker, Christian Schallhart: ***The Good, the Bad, and the Ugly, But How Ugly Is Ugly?*** RV 2007: 126-138

RV-LTL Semantics

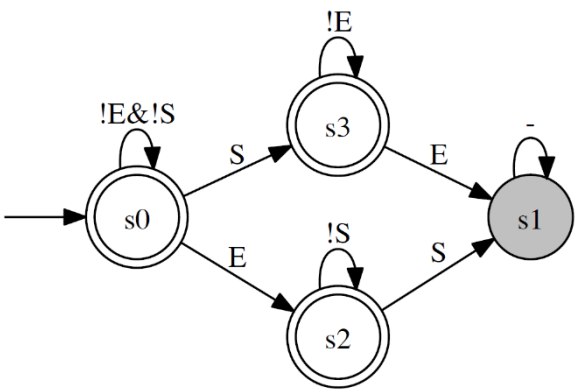


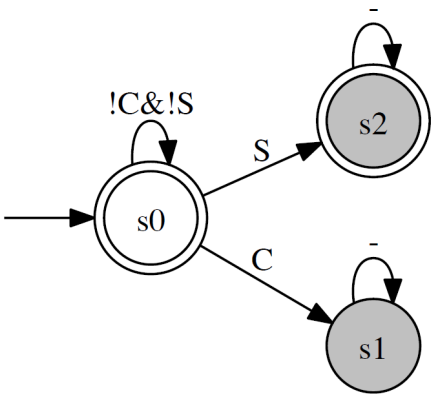


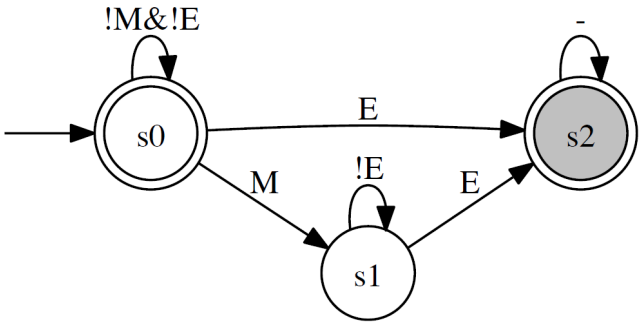


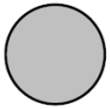
"the good" (permanently sat)

RV-LTL Semantics



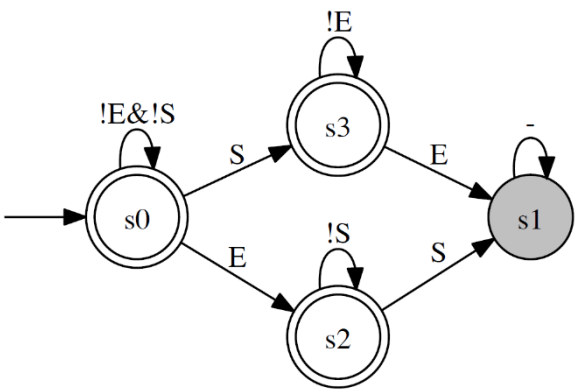


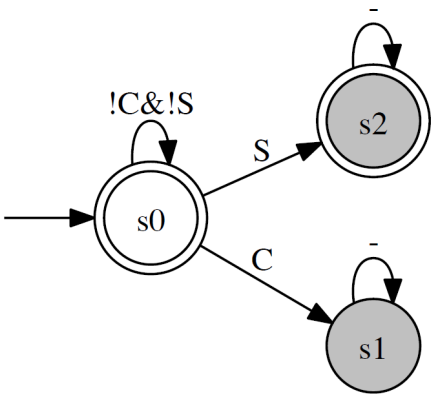


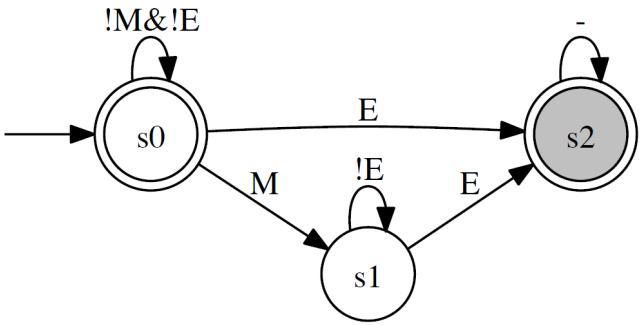


"the bad" (permanently viol)

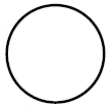
RV-LTL Semantics







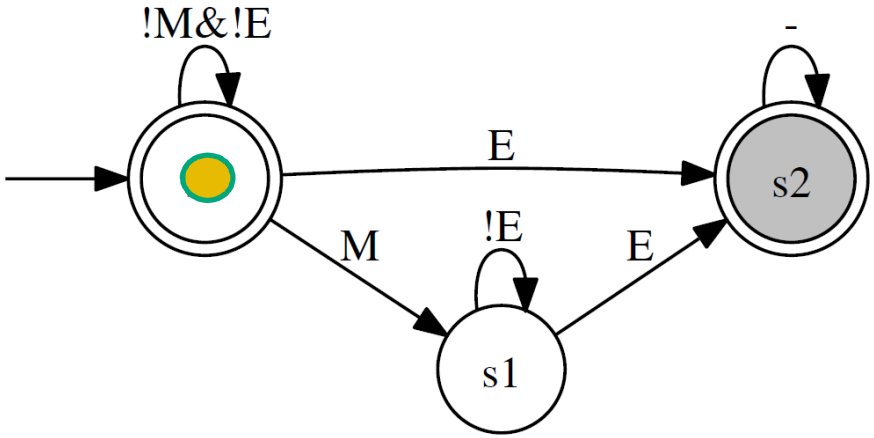
“inconclusive” (possibly sat)

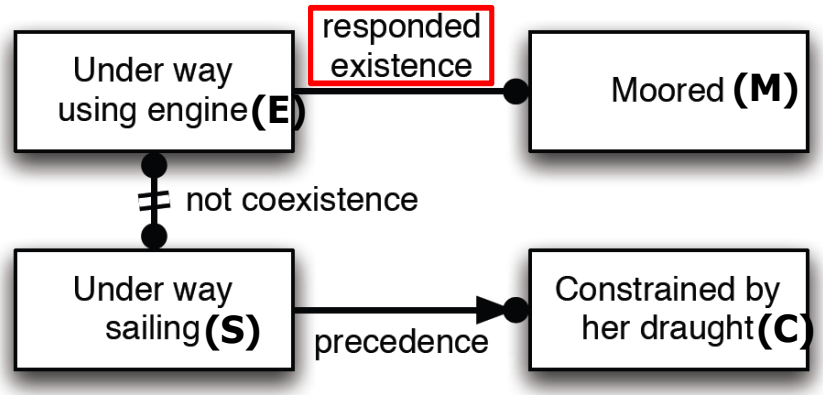


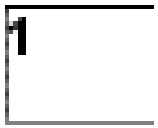
“inconclusive” (possibly viol)

Monitoring Approach

Fabrizio M. Maggi, Marco Montali, Michael Westergaard, Wil M. P. van der Aalst: ***Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata***. BPM 2011: 132-147



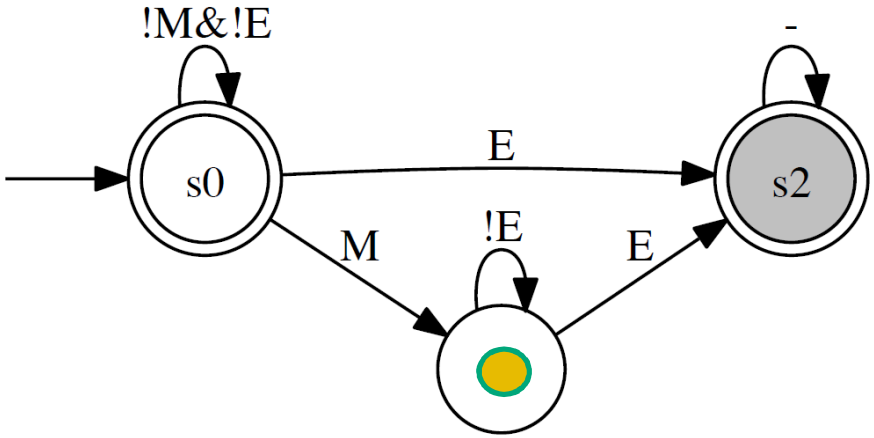


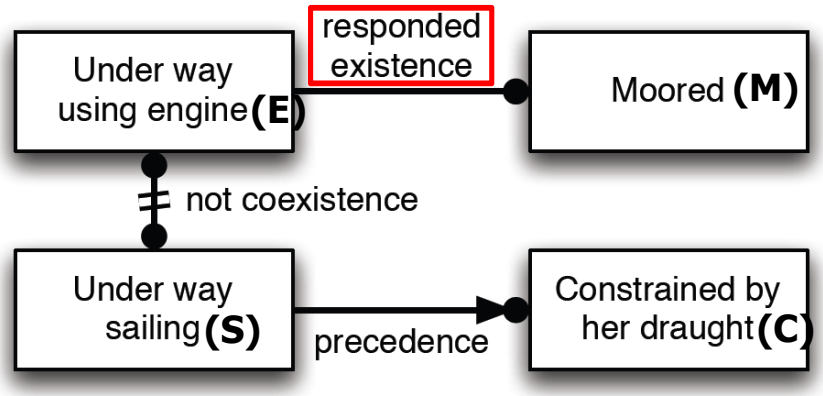


start M S E

Monitoring Approach

Fabrizio M. Maggi, Marco Montali, Michael Westergaard, Wil M. P. van der Aalst: ***Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata***. BPM 2011: 132-147

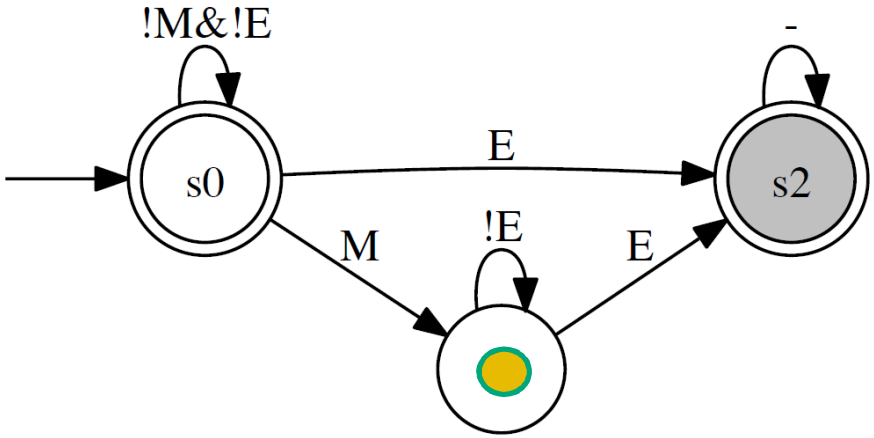


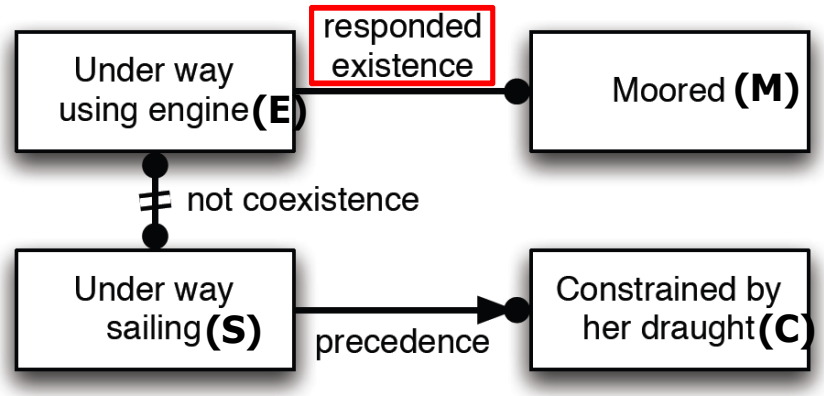


start M S E

Monitoring Approach

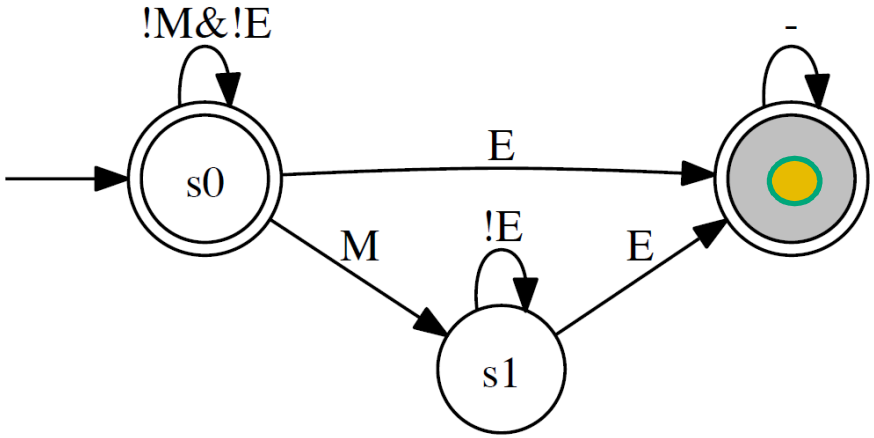
Fabrizio M. Maggi, Marco Montali, Michael Westergaard, Wil M. P. van der Aalst: ***Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata***. BPM 2011: 132-147

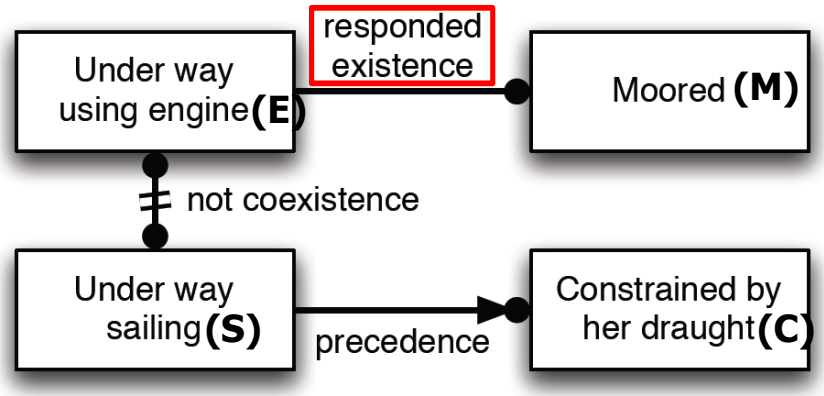




Monitoring Approach

Fabrizio M. Maggi, Marco Montali, Michael Westergaard, Wil M. P. van der Aalst: ***Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata***. BPM 2011: 132-147

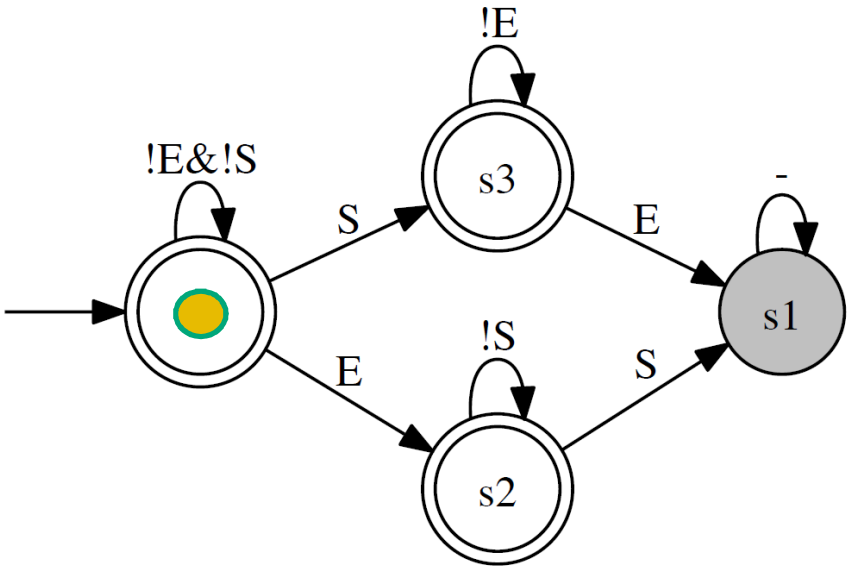


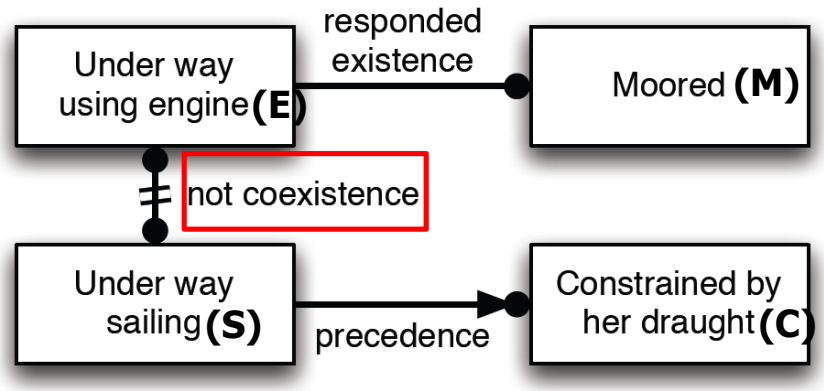


start M S E

Monitoring Approach

Fabrizio M. Maggi, Marco Montali, Michael Westergaard, Wil M. P. van der Aalst: ***Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata***. BPM 2011: 132-147

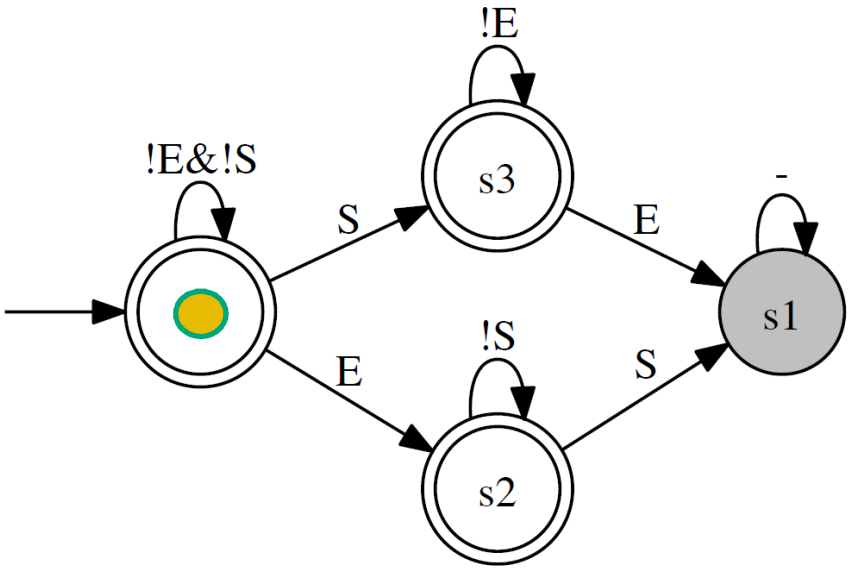


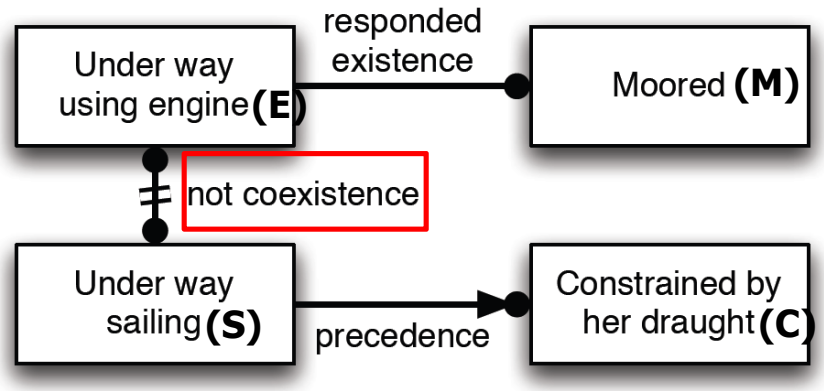


start M S E

Monitoring Approach

Fabrizio M. Maggi, Marco Montali, Michael Westergaard, Wil M. P. van der Aalst: ***Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata***. BPM 2011: 132-147

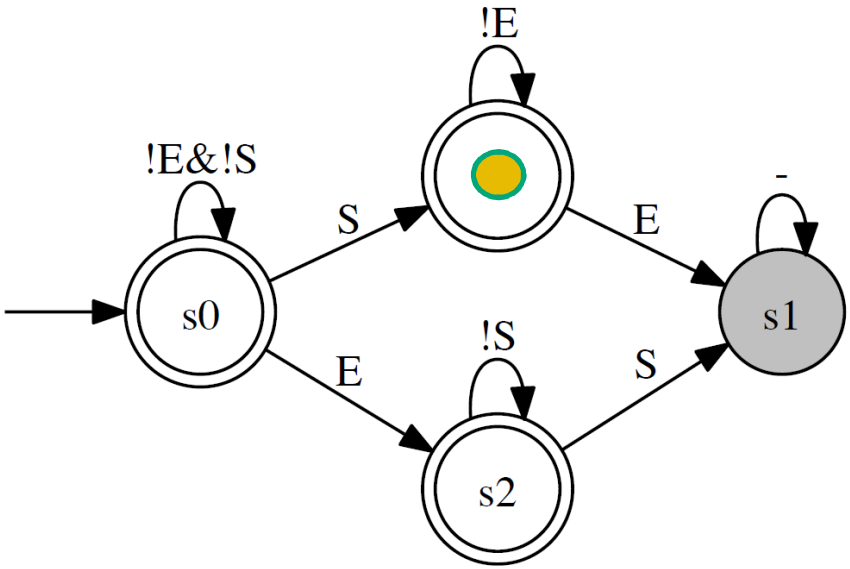


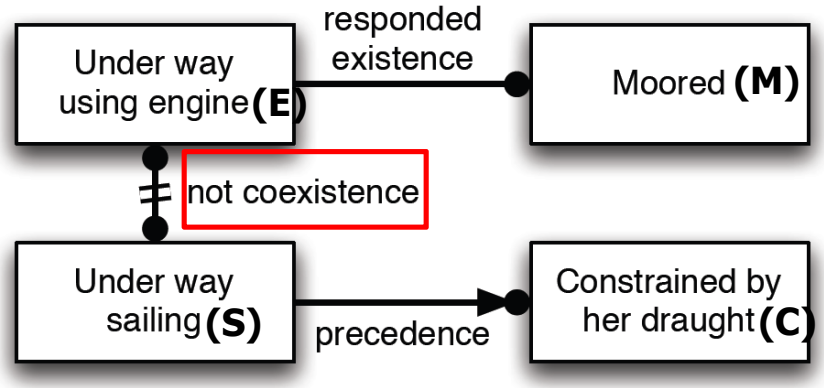


start M S E

Monitoring Approach

Fabrizio M. Maggi, Marco Montali, Michael Westergaard, Wil M. P. van der Aalst: ***Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata***. BPM 2011: 132-147

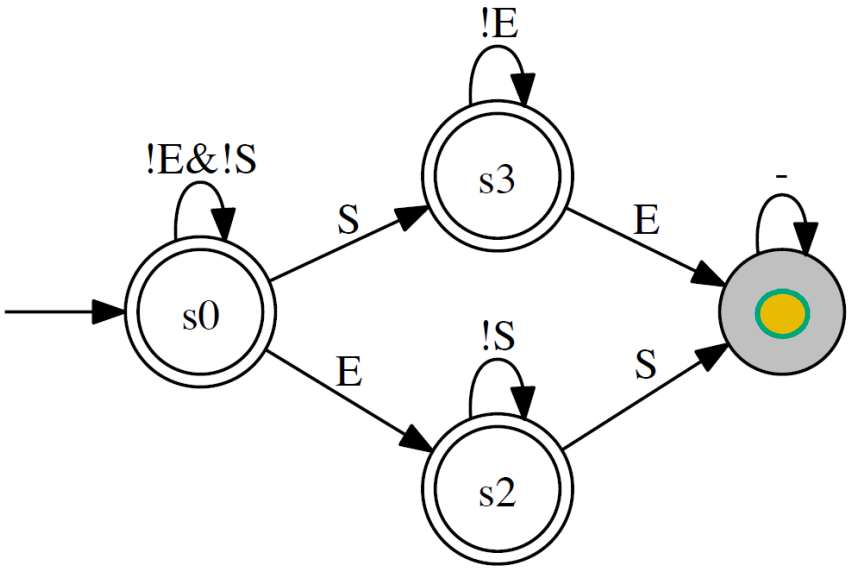


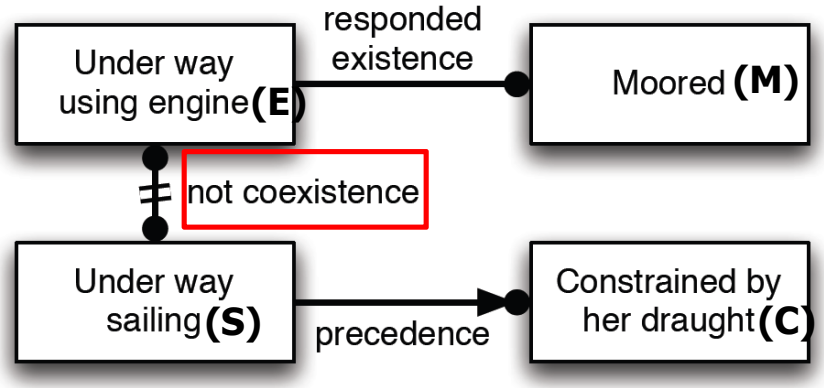


start M S E

Monitoring Approach

Fabrizio M. Maggi, Marco Montali, Michael Westergaard, Wil M. P. van der Aalst: ***Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata***. BPM 2011: 132-147





start M S E

Compliance Monitoring

rulemining.org

Roadmap

Automated Process Discovery

Compliance Monitoring

Conformance Checking

Log Generation

Conformance Checking with Trace Alignment

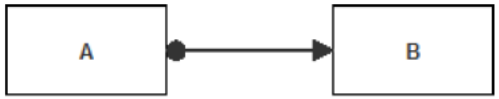
out of Alignment

in Alignment

Trace Alignment

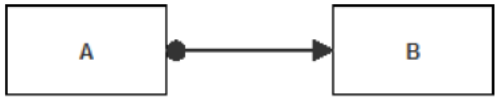
Massimiliano de Leoni, Fabrizio M. Maggi, Wil M. P. van der Aalst: ***Aligning Event Logs and Declarative Process Models for Conformance Checking***. BPM 2012: 82-97

<A A D F C D E>



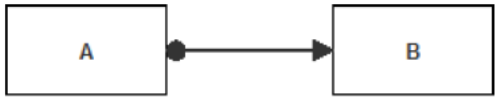
Trace Alignment

<A A D F C D E>



Trace Alignment

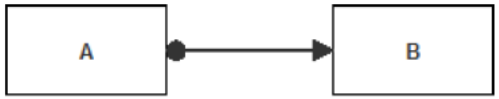
<A A D F C D E>

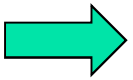


<A A D F C D E B>

Trace Alignment

<A A D F C D E>



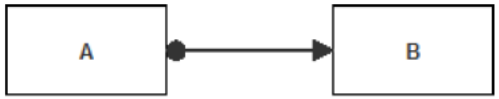


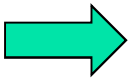
<A A D F C D E B>

in Alignment

Trace Alignment

<A A D F C D E>



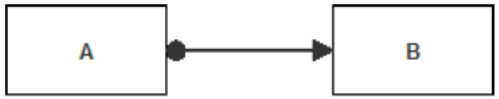


<A A D F C D E B>

in Alignment

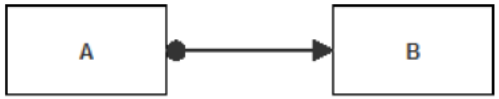
Trace Alignment

<A A D F C D E>



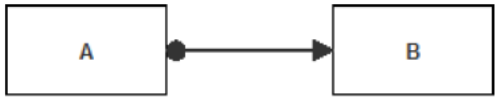
Trace Alignment

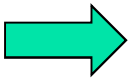
<A A D F C D E>



Trace Alignment

<A A D F C D E>



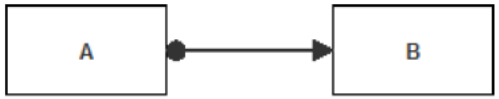


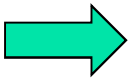
<A A D F C D E>

in Alignment

Trace Alignment

<A A D F C D E>



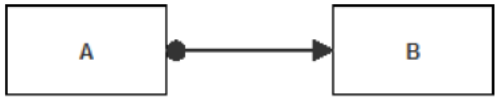


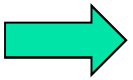
<A A D F C D E>

in Alignment

Trace Alignment

<A A D F C D E>



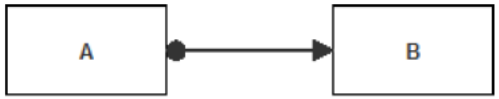


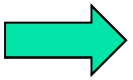
<A A D F C D E>

in Alignment

Trace Alignment

<A A D F C D E>





<A A D F C D E>

in Alignment

Optimal Alignments

Modifications have a cost!

Trace Alignment as a Planning Problem

- Trace Alignment can be formulated as a cost-optimal planning problem
  - **Domain**:
    - ✓ Addition and Deletion modeled by actions with positive costs
    - ✓ Domain dynamics encodes synchronous execution of a trace and all constraint automata
  - **Problem**:
    - ✓ Initial state: all automata in their initial state
    - ✓ Goal state: all automata in a final state
  - **Solution**:
    - ✓ Optimal (i.e., minimal-cost) plan to reach the goal state

Giuseppe De Giacomo, Fabrizio M. Maggi, Andrea Marrella, Fabio Patrizi: ***On the Disruptive Effectiveness of Automated Planning for LTLf-Based Trace Alignment***. AAAI 2017: 3555-3561

- To find the minimum cost trace alignment, any planner can be used
  - e.g., **Fast-Downward**, **SymBA*-2**

Conformance Checking

rulemining.org

Roadmap

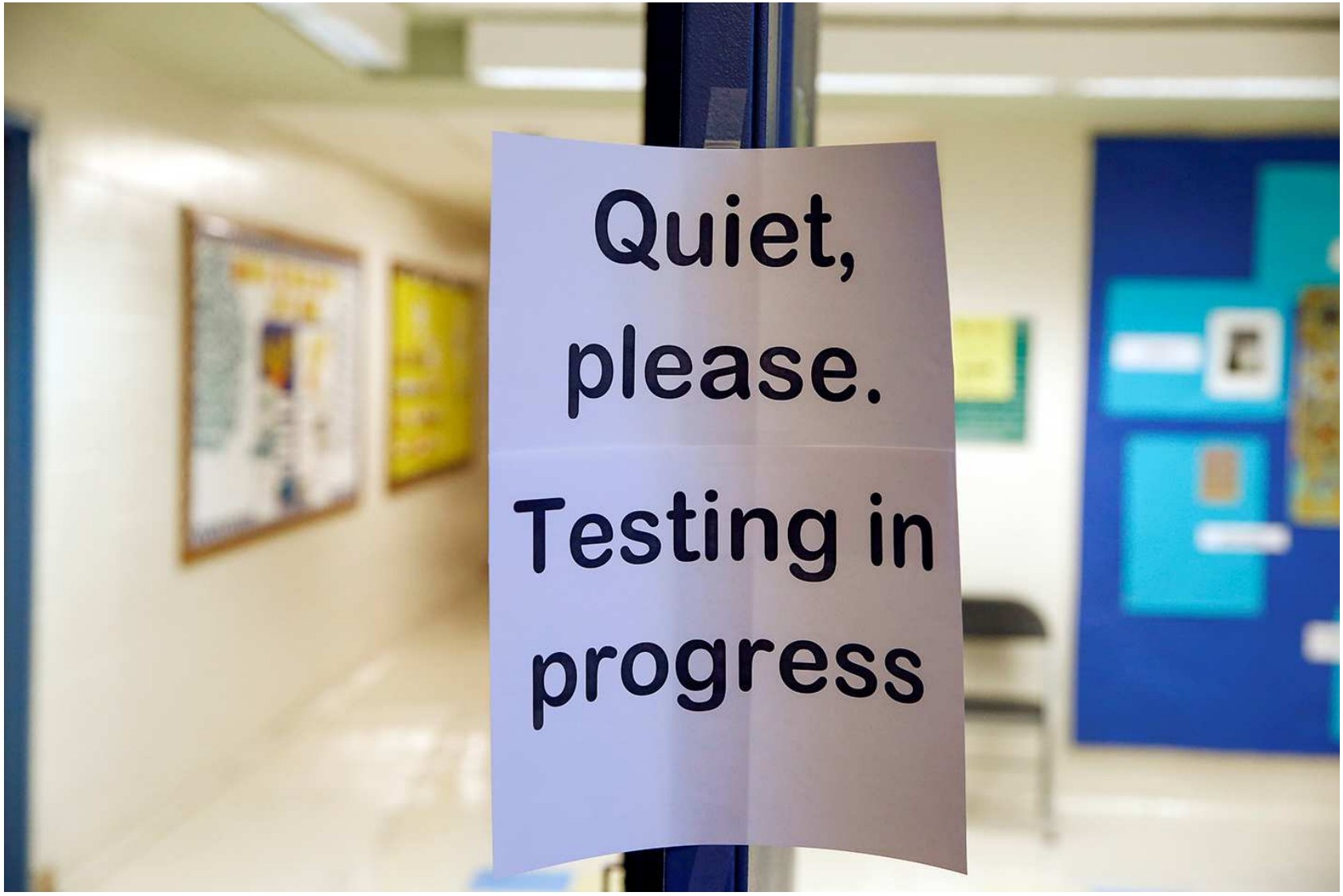
Automated Process Discovery

Compliance Monitoring

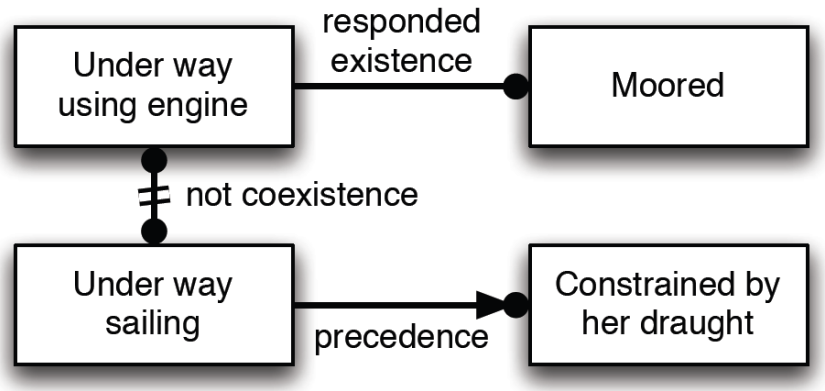
Conformance Checking

Log Generation

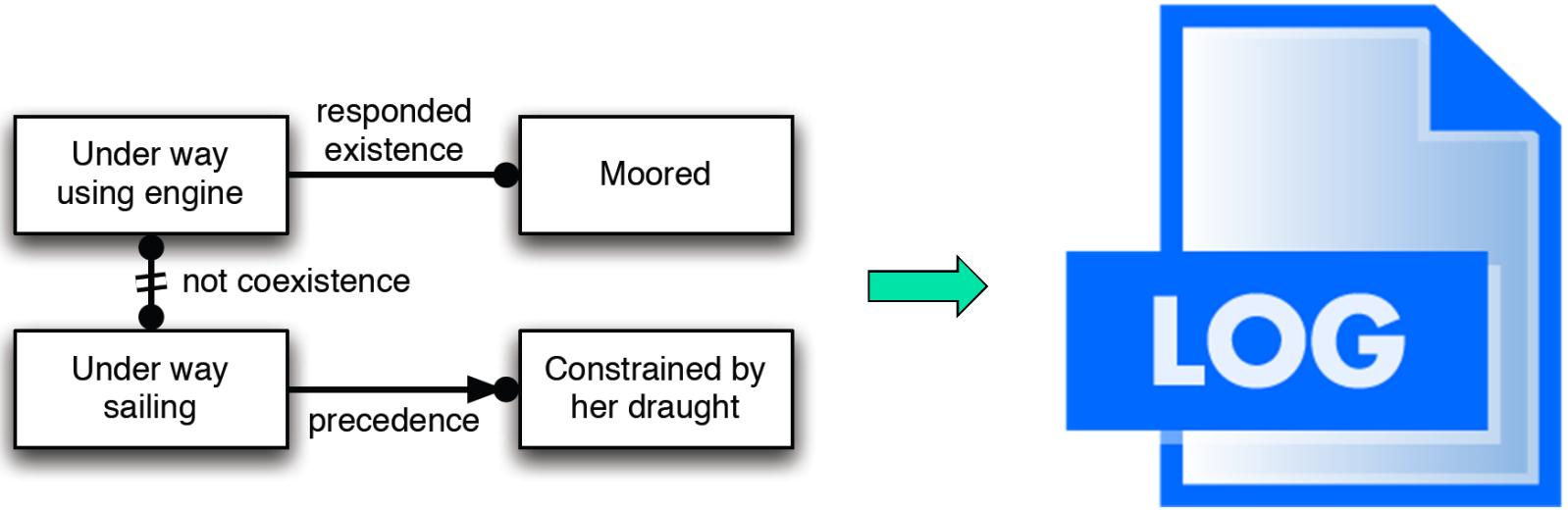
Log Generation



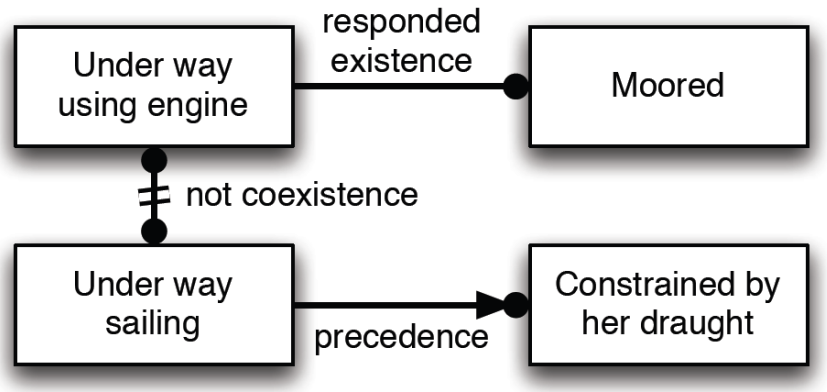
Log Generation



Log Generation



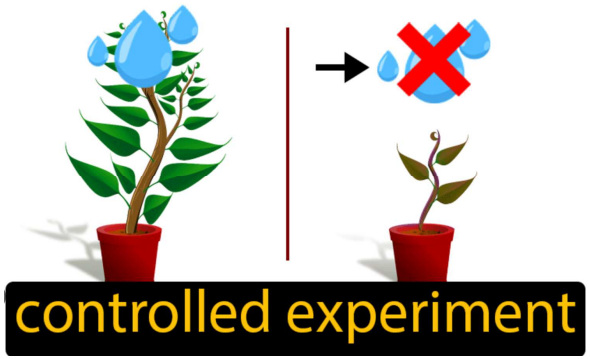
Log Generation



#traces in the log

#events in a trace

…



Log Generation with ASP

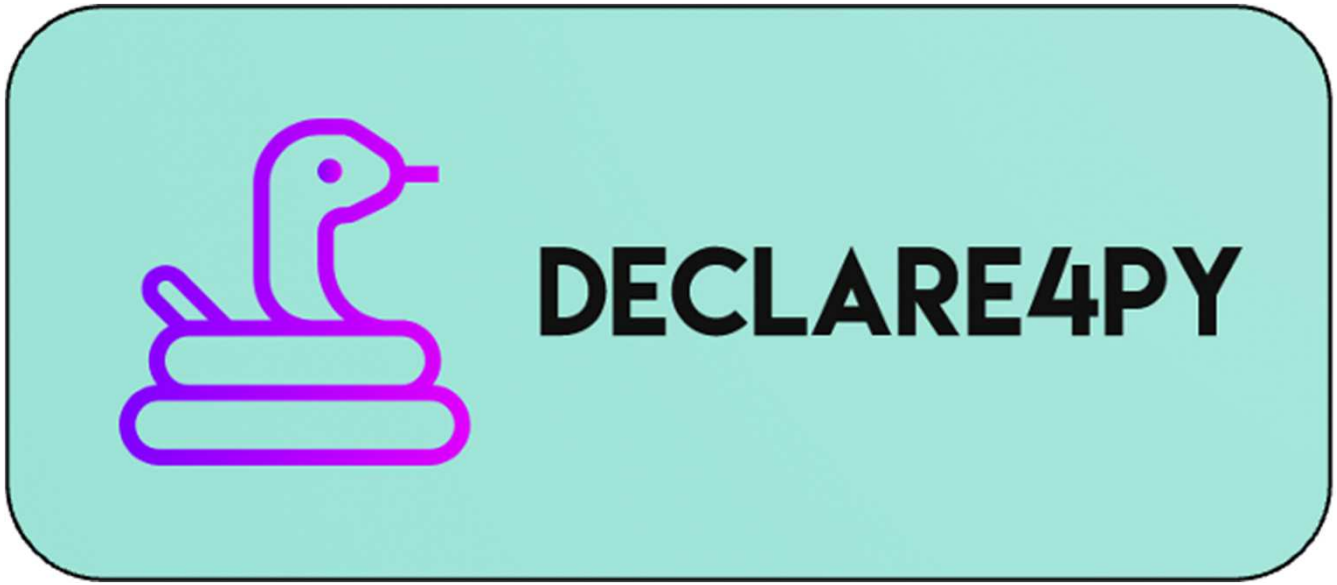
- Log generation can be encoded in Answer Set Programming (ASP) using predicates and rules to define:
  - Each constraint automaton of the input model
  - The length of a trace to be generated
  - The requirement that every automaton ends up in a final state in the last time point of the generated trace
- Predicate ***trace*** is the guessed predicate and contains a sequence of activities satisfying all the input constraints
- To generate a solution for the guessed predicate, an ASP solver can be used
  - e.g., **Clingo**

Francesco Chiariello, Fabrizio M. Maggi, Fabio Patrizi: ***ASP-Based Declarative Process Mining***. AAAI 2022: 5539-5547

Log Generation

rulemining.org

For Python Users…



https://github.com/francxx96/declare4py

Thank you
for your attention!