Process Modeling and Analysis

Andrea Polini

Process Mining
MSc in Computer Science (LM-18)
University of Camerino

Summary

1. Petri Nets (PN)
2. Transition Systems
3. Workflow Nets
4. YAWL
5. BPMN
6. Causal Nets
7. Process Trees
8. Model Based Analysis

Summary

1. Petri Nets (PN)
2. Transition Systems
3. Workflow Nets
4. YAWL
5. BPMN
6. Causal Nets
7. Process Trees
8. Model Based Analysis

Petri Nets
Algebraic definition

A Petri Net is an operational formalism conceived in the 60s by the mathematician Adam Petri. It is particularly suitable to model concurrent, asynchronous, distributed, parallel, non deterministic, and stochastic systems. They can be formally defined using the following tuple: $<\mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{W}, \mathcal{M}_0>$:

- $\mathcal{P}$ finite set of places
- $\mathcal{T}$ finite set of transitions
- $\mathcal{F} \subseteq \{\mathcal{P} \times \mathcal{T}\} \cup \{\mathcal{T} \times \mathcal{P}\}$ represents the flow relation for the network
- $\mathcal{W} : \mathcal{F} \rightarrow \mathbb{N}^+$ is the weight function. It permits to associate a positive value to the elements of $\mathcal{F}$
- $\mathcal{M}_0 : \mathcal{P} \rightarrow \mathbb{N}$ is the initial marking and it is needed to represent the initial state of the net.

Moreover the following relations must hold: $\mathcal{P} \cup \mathcal{T} \neq \oslash$ and $\mathcal{P} \cap \mathcal{T} = \oslash$

Petri Nets
Algebraic definition

A Petri Net is an operational formalism conceived in the 60s by the mathematician Adam Petri. It is particularly suitable to model concurrent, asynchronous, distributed, parallel, non deterministic, and stochastic systems. They can be formally defined using the following tuple: $<\mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{W}, \mathcal{M}_0>$:

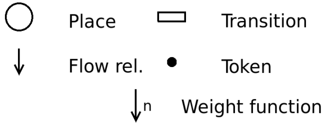
- $\mathcal{P}$ finite set of places
- $\mathcal{T}$ finite set of transitions
- $\mathcal{F} \subseteq \{\mathcal{P} \times \mathcal{T}\} \cup \{\mathcal{T} \times \mathcal{P}\}$ represents the flow relation for the network
- $\mathcal{W} : \mathcal{F} \rightarrow \mathbb{N}^+$ is the weight function. It permits to associate a positive value to the elements of $\mathcal{F}$
- $\mathcal{M}_0 : \mathcal{P} \rightarrow \mathbb{N}$ is the initial marking and it is needed to represent the initial state of the net.

Moreover the following relations must hold: $\mathcal{P} \cup \mathcal{T} \neq \oslash$ and $\mathcal{P} \cap \mathcal{T} = \oslash$

The status of the net is given by the marking function that associate to any place a natural number representing the number of tokens in a place at a given time. : $\mathcal{M} : \mathcal{P} \rightarrow \mathbb{N}$

Petri Net

Graphical Notation and behaviour



Evolution of a Petri Net :

- Given a transition in $t \in \mathcal{T}$ we call input place (or output place), a place $p_i \in \mathcal{P}$ connected to the transition as input flow, i.e. $(p_i, t) \in \mathcal{F}$ (a place $p_o \in \mathcal{P}$ connected to the transition as output flow – $(t, p_o) \in \mathcal{F}$)
- A transition is enabled, and then it can fire, iff **all** the input places contains a number of tokens greater or equal to the weight associated to each corresponding input flow. Formally a transition t_e is enabled if $\forall p \in \mathcal{P}.(p, t_e) \in \mathcal{F} \implies \mathcal{M}(p) \geq \mathcal{W}(p, t_e)$
- When a transition fires a number of token corresponding to the weight associated to the corresponding incoming flow is removed from each incoming place. At the same time a number of token corresponding to the weight of the outgoing flow is added to each outgoing place.
- A transition for which there is no input place is always enabled.

Petri Net
Modeling guidelines

Generally places are used to represent resources and the availability of a resource is represented by the presence of at least one token on the corresponding place. The state of a process is represented by the marking funtions and transitions are used to let the state evolve.

Policies for conflic resolutions are not intrinsic in the formalism instead they have to be implemented. For instance the formalism is not *fair*. For instance....

A Petri Net is in deadlock if there are not enabled transitions. In case the net is in a deadlock situation the corresponding marking will represent a possible final state.

Exemplificative models

- Traffic light
- Two position Buffer
- Reader/Writer
- Dining philosophers

Some additional notation

Let $\mathcal{PN} = \langle \mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{W}, \mathcal{M}_0 \rangle$ a Petri net and $x \in \mathcal{P} \cup \mathcal{T}$ a node of $\mathcal{PM}$:

- $\bullet x = \{(y|(y,x) \in \mathcal{F}\}$ – y is an input node of x
- $x\bullet = \{(y|(x,y) \in \mathcal{F}\}$ – y is an output node of x
- A transition $t \in \mathcal{T}$ is enabled – $(\mathcal{PN}, \mathcal{M})[t\rangle$ – iff $\forall y \in \bullet t.M(y) \geq \mathcal{W}(y,t)$
- The firing rule ...and we write $(\mathcal{PN}, \mathcal{M}_n)[t\rangle(\mathcal{PN}, \mathcal{M}_{n+1})$
- A sequence $\sigma \in T^*$ is called a firing sequence iff for some $n \in \mathbb{N}$ there exists markings $\mathcal{M}_1, \ldots, \mathcal{M}_n$ and transitions $t_1, \ldots, t_n \in \mathcal{T}$ s.t. $(\mathcal{PN}, \mathcal{M})[t_i\rangle \wedge (\mathcal{PN}, \mathcal{M})[t_i\rangle(\mathcal{PN}, \mathcal{M}_{i+1})$ for each $1 \leq i \leq n$
- A marking $\mathcal{M}$ is reaceable if there is a firing sequence from $\mathcal{M}_0$ to $\mathcal{M}$. $[\mathcal{PN}\rangle$ represents the set of reacheable markings.
- A labeled Petri Net is a tuple $\mathcal{PNL} = \langle \mathcal{PN}, \mathcal{A}, \mathcal{L} \rangle$ where $\mathcal{PN}$ is a Petri Net, $\mathcal{A} \in \mathbb{A}$ is a set of activity labels, and $\mathcal{L} : \mathcal{T} \rightarrow \mathcal{A}$ is a labeling function.
  - τ is used to represent silent activities

Limitations and possible extensions

The formalism focuses on control related aspects. It does not easily support the representation of data. In particular:

- it is not possible to direct the flow on the base of data conditions
- It is not possible to select one transition when more of them are active
- It is not possible to specify time related aspects and deadlines.

Several extensions available to model more complex systems:

- Tokens with value associated. Typed Tokens and then content management and conditions (Coloured Petri Nets - CPN)
- Priorities associated to transitions $pri : \mathcal{T} \to \mathbb{N}$.
- Timed Petri Nets. To each transition two value are associated to represent minimal and maximal time to fire when enabled.
- Timed Petri Nets with probability distributions associated to transitions (Stochastic Petri Nets - SPN)

Summary

1. Petri Nets (PN)
2. Transition Systems
3. Workflow Nets
4. YAWL
5. BPMN
6. Causal Nets
7. Process Trees
8. Model Based Analysis

Transition Systems

Definition

A transition system is a triplet $\mathcal{TS} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T} \rangle$ where $\mathcal{S}$ is the set of states, $\mathcal{A} \subseteq \mathbb{A}$ is the set of activities (often referred to as actions), and $\mathcal{T} \subseteq \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ is the set of transitions. $\mathcal{S}^{start} \subseteq \mathcal{S}$ is the set of initial states (sometimes referred to as "start" states), and $\mathcal{S}^{end} \subseteq \mathcal{S}$ is the set of final states (sometimes referred to as "accept" states).

Qualities

In principle $\mathcal{S}$ could be an infinite set but for most pratical applications the set is finite (FSM).

Any path in the graph starting in an initial state corresponds to a possible execution sequence.

- A path terminates successfully if it ends in one of the final states.
- A path deadlocks if it reaches a non-final state without any outgoing transitions.
- The transition system may livelock, i.e., some transitions are still enabled but it is impossible to reach one of the final states.

Qualities

In principle $\mathcal{S}$ could be an infinite set but for most pratical applications the set is finite (FSM).

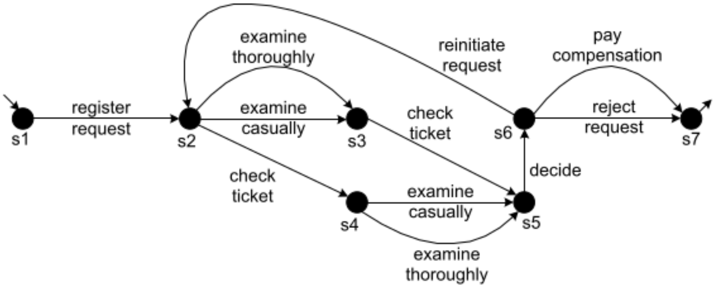
Any path in the graph starting in an initial state corresponds to a possible execution sequence.

- A path terminates successfully if it ends in one of the final states.
- A path deadlocks if it reaches a non-final state without any outgoing transitions.
- The transition system may livelock, i.e., some transitions are still enabled but it is impossible to reach one of the final states.

State explosion phenomenon

Transition systems are simple but have problems expressing concurrency succinctly. Suppose that there are n parallel activities, i.e., all n activities need to be executed but any order is allowed. There are n! possible execution sequences. The transition system requires 2^n states and $n \times 2^{n-1}$ transitions.

Example



Do you recognize which activities could be run in parallel?

(Look for a diamond structure)

Summary

1. Petri Nets (PN)
2. Transition Systems
3. Workflow Nets
4. YAWL
5. BPMN
6. Causal Nets
7. Process Trees
8. Model Based Analysis

Workflow Nets

Definition

Let $\mathcal{PNL} = \langle \mathcal{PN}, \mathcal{A}, \mathcal{L} \rangle$ be a (labeled) Petri net and t' a fresh identifier not in $\mathcal{P} \cup \mathcal{T}$. $\mathcal{PNL}$ is a workflow net (WF-net) iff

- $\mathcal{P}$ contains an input place i (source place) s.t. $\bullet i = \emptyset$
- $\mathcal{P}$ contains an output place o (sink place) s.t. $o \bullet = \emptyset$
- $\mathcal{PN}' = \langle \mathcal{P}, \mathcal{T} \cup t', \mathcal{F} \cup \{(o, t'), (t', i)\}, \mathcal{W} \cup \{(o, t') \rightarrow 1, (t', i) \rightarrow 1\}, \mathcal{M}_0 \rangle$ and $\mathcal{PNL}' = \langle \mathcal{PN}', \mathcal{A} \cup \tau, \mathcal{L} \cup \{t' \rightarrow \tau\} \rangle$ is strongly connected, i.e., there is a directed path between any pair of nodes in $\mathcal{PNL}'$.

WF-Net are then a restricted form of PN, and they are widely used in process mining

Soundness of a WF-Net

Soundness is a very important property for BP models.

Definition

Let $\mathcal{WN} = \langle \mathcal{PN}, \mathcal{A}, \mathcal{L} \rangle$ a WF-net with input place *i* and output place *o*. $\mathcal{WN}$ is sound iff:

- safeness – $(\mathcal{WN}, [i])$ is safe, i.e., places cannot hold multiple tokens at the same time;
- proper completion – for any marking $\mathcal{M} \in [\mathcal{WN}, [i]\rangle$, $o \in \mathcal{M}$ implies $\mathcal{M} = [o]$
- option to complete – for any marking $\mathcal{M} \in [\mathcal{WN}, [i]\rangle$, $[o] \in [\mathcal{WN}, \mathcal{M}\rangle$;
- absence of dead parts – $(\mathcal{WN}, [i])$ contains no dead transitions (i.e., for any $t \in \mathcal{T}$, there is at least a firing sequence enabling *t*).

Summary

1 Petri Nets (PN)

2 Transition Systems

3 Workflow Nets

4 YAWL

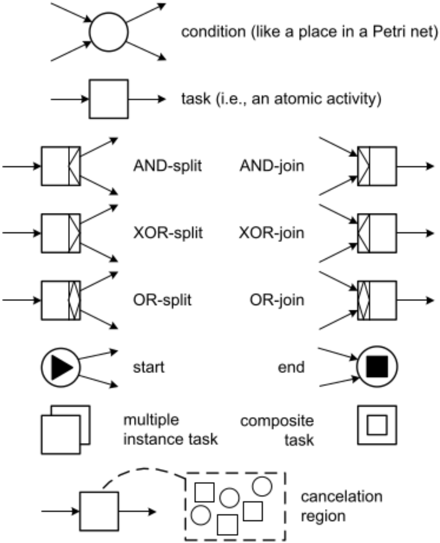
5 BPMN

6 Causal Nets

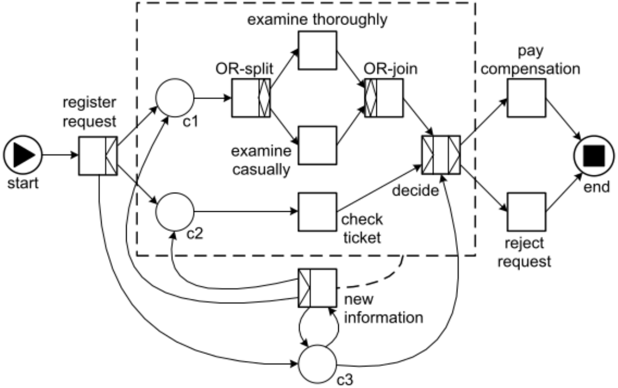
7 Process Trees

8 Model Based Analysis

YAWL - Yet Another Workflow Language



Model example in YAWL

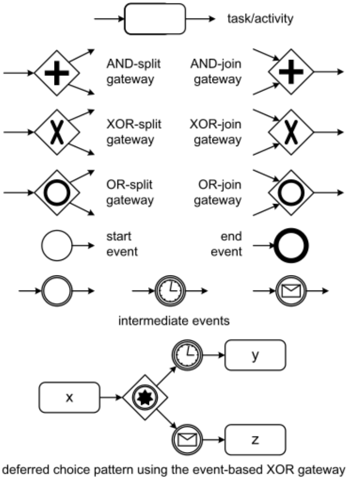


Summary

1. Petri Nets (PN)
2. Transition Systems
3. Workflow Nets
4. YAWL
5. BPMN
6. Causal Nets
7. Process Trees
8. Model Based Analysis

BPMN - Business Process Modeling Language

OMG standard - version 2.0 released in 2011



Summary

1. Petri Nets (PN)
2. Transition Systems
3. Workflow Nets
4. YAWL
5. BPMN
6. Causal Nets
7. Process Trees
8. Model Based Analysis

Summary

1. Petri Nets (PN)
2. Transition Systems
3. Workflow Nets
4. YAWL
5. BPMN
6. Causal Nets
7. Process Trees
8. Model Based Analysis

Summary

1. Petri Nets (PN)
2. Transition Systems
3. Workflow Nets
4. YAWL
5. BPMN
6. Causal Nets
7. Process Trees
8. Model Based Analysis

Quality related aspects

For Process Mining there are many interesting questions to answer to better understand the results and the quality of mining activities. In particular:

- How can we relate process models?
- Do the models satisfy behavioural requirements?

Reacheability graphs for PN

Definition

Let $\mathcal{PNL} = \langle \mathcal{PN}, \mathcal{A}, \mathcal{L} \rangle$ a labeled Petri Net with $\mathcal{PN} = \langle \mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{W}, \mathcal{M}_0 \rangle$ be a Petri net, $\mathcal{A}$ a set of activities, and $\mathcal{L}$ a labeling function. $\langle \mathcal{PN}, \mathcal{A}, \mathcal{L} \rangle$ defines a transition system $\mathcal{TS} = (\mathcal{S}, \mathcal{A}', \mathcal{T}')$ with $\mathcal{S}$ = $[\mathcal{PN}\rangle$, $\mathcal{S}^{start} = \{\mathcal{M}_0\}, \mathcal{A}' = \mathcal{A}$, and $\mathcal{T}' = \{(\mathcal{M}, \mathcal{L}(t), \mathcal{M}') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ s.t. $\exists t \in \mathcal{T}(\mathcal{PNL}, \mathcal{M})[t\rangle(\mathcal{PNL}, \mathcal{M}')\}$.
TS is often referred to as the reachability graph of $\mathcal{PNL}$

Reacheability graphs for PN

Definition

Let $\mathcal{PNL} = \langle \mathcal{PN}, \mathcal{A}, \mathcal{L} \rangle$ a labeled Petri Net with $\mathcal{PN} = \langle \mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{W}, \mathcal{M}_0 \rangle$ be a Petri net, $\mathcal{A}$ a set of activities, and $\mathcal{L}$ a labeling function. $\langle \mathcal{PN}, \mathcal{A}, \mathcal{L} \rangle$ defines a transition system $\mathcal{TS} = (\mathcal{S}, \mathcal{A}', \mathcal{T}')$ with $\mathcal{S}$ = $[\mathcal{PN}\rangle$, $\mathcal{S}^{start} = \{\mathcal{M}_0\}, \mathcal{A}' = \mathcal{A}$, and $\mathcal{T}' = \{(\mathcal{M}, \mathcal{L}(t), \mathcal{M}') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ s.t. $\exists t \in \mathcal{T}(\mathcal{PNL}, \mathcal{M})[t\rangle(\mathcal{PNL}, \mathcal{M}')\}$.
TS is often referred to as the reachability graph of $\mathcal{PNL}$

Reacheability Graph – exercise

Derive the RG for the traffic lights network

PN relevant property

There are some general property worthy to be checked on a $\mathcal{PN}$:

- A Petri net $\mathcal{PN}$ is k-bounded if no place ever holds more that k tokens. Formally, for any $p \in \mathcal{P}$ and any $M \in [\mathcal{PN}\rangle : \mathcal{M}(p) \leq k$.
- A Petri net is safe if and only if it is 1-bounded.
- A marked Petri net is bounded if and only if there exists a $k \in \mathbb{N}$ such that it is k-bounded.
- A Petri net $\mathcal{PN}$, is deadlock free if at every reachable marking at least one transition is enabled. Formally, for any $\mathcal{M} \in [\mathcal{N}\rangle$ there exists a transition $t \in \mathcal{T} s.t. (\mathcal{PN}, \mathcal{M})[t\rangle$.
- A transition $t \in \mathcal{T}$ in a Petri net $\mathcal{PN}$ is live if from every reachable marking it is possible to enable t. Formally, for any $\mathcal{M} \in [\mathcal{PN}\rangle$ there exists a marking $\mathcal{M}' \in [\mathcal{PN}, \mathcal{M}\rangle$ s.t. $(\mathcal{PN}, \mathcal{M}')[t\rangle$. A marked Petri net is live if each of its transitions is live. Note that a deadlock-free Petri net does not need to be live.

TS and behavioural equivalences

When TS are considered an interesting aspect to investigate refers to the comparison of two TS to discover when they are "behaviourally equivalent".
Obviously this aspect is relevant for process mining

TS equivalence

Many different definitions of equivalence can be given.

- Trace equivalence considers two transition systems to be equivalent if their execution sequences are the same
- Notions like branching bisimilarity also take the moment of choice into account