Process Modeling and Analysis

Andrea Polini

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

Summary

1. Petri Nets (PN)

Summary

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: $<\mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{W}, \mathcal{M}_0>$:

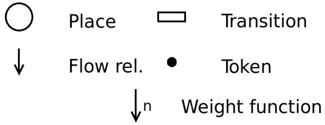
- $\mathcal{P}$ finite set of places
- $\mathcal{T}$ finite set of transitions
- $\mathcal{F} \subseteq \{\mathcal{P} \times \mathcal{T}\} \cup \{\mathcal{T} \times \mathcal{P}\}$ represents the flow relation for the network
- $\mathcal{W} : \mathcal{F} \rightarrow \mathbb{N}^+$ is the weight function. 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

Limitations and possible estensions

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 one 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 $pri : \mathcal{T} \rightarrow \mathbb{N}$.
- 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)