

See example specifications that use this module.

MODULE *PetriNet*

from “Formal definition and basic terminology” [https://en.wikipedia.org/wiki/Petri\\_net](https://en.wikipedia.org/wiki/Petri_net)

Definition 1. A net is a tuple  $N = (P, T, F)$  where

1.  $P$  and  $T$  are disjoint finite sets of places and transitions, respectively.
2.  $F \subseteq (P \times T) \cup (T \times P)$  is a set of (directed) arcs (or flow relations).

Definition 4. A Petri net is a net of the form  $PN = (N, M, W)$ , which extends the elementary net so that

1.  $N = (P, T, F)$  is a net.
2.  $M : P \rightarrow Z$  is a place multiset, where  $Z$  is a countable set.  $M$  extends the concept of configuration and is commonly described with reference to Petri net diagrams as a marking.
3.  $W : F \rightarrow Z$  is an arc multiset, so that the count (or weight) for each arc is a measure of the arc multiplicity.

\* firing a transition  $t$  in a marking  $M$  consumes  $W(s, t)$  tokens from each of its input places  $s$ , and produces  $W(t, s)$  tokens in each of its output places  $s$

\* a transition is enabled (it may fire) in  $M$  if there are enough tokens in its input places for the consumptions to be possible, *i.e.* if and only if  $\forall s: M(s) \geq W(s, t)$ .

Instantiate *PetriNet* with (*Places* , *Transitions* , *Arcs* , *InitialMarking* , *ArcWeights* ) constants and (*Marking* ) variable. *Marking* variable should be declared but not assigned by users of this module.

LOCAL INSTANCE *Integers*

LOCAL INSTANCE *Sequences*

LOCAL INSTANCE *FiniteSets*

LOCAL INSTANCE *TLC*

LOCAL INSTANCE *Helpers*

CONSTANTS *Places*, *Transitions*, *Arcs*, *InitialMarking*, *ArcWeights*

*ConstsInvariant*  $\triangleq \wedge Places \in \text{SUBSET STRING}$

$\wedge Transitions \in \text{SUBSET STRING}$

$\wedge \forall k \in \text{DOMAIN } Arcs : \wedge k \in \text{STRING}$

$\wedge Arcs[k] \in \text{SUBSET STRING}$

$\wedge \forall p \in \text{DOMAIN } InitialMarking : \wedge p \in \text{STRING}$

$\wedge InitialMarking[p] \in Int$

An arc weight is a tuple of (from node, to node, weight)

$\wedge \forall i \in 1 \dots Len(ArcWeights) : \wedge Len(ArcWeights[i]) = 3$

$\wedge ArcWeights[i][1] \in \text{STRING}$

$\wedge ArcWeights[i][2] \in \text{STRING}$

$\wedge ArcWeights[i][3] \in Int$

ASSUME *ConstsInvariant*

Marking is a Bag where the domain is *Places* and the range is  $Int \geq 0$ .

VARIABLE *Marking*

*vars*  $\triangleq \langle Marking \rangle$

---

## Invariants

$$\begin{aligned}
TypeInvariant &\triangleq \wedge ConstsInvariant \\
&\wedge \forall p \in DOMAIN \ Marking : \wedge p \in STRING \\
&\wedge Marking[p] \in Int \\
ModelInvariant &\triangleq \wedge Places \cap Transitions = \{\} \\
&\wedge \forall k \in DOMAIN \ Arcs : \vee (k \in Places \wedge Arcs[k] \subseteq Transitions) \\
&\quad \vee (k \in Transitions \wedge Arcs[k] \subseteq Places) \\
&\wedge \forall k \in DOMAIN \ InitialMarking : \wedge k \in Places \\
&\quad \wedge InitialMarking[k] \geq 0 \\
&\wedge \forall i \in 1 \dots Len(ArcWeights) : \\
&\quad \wedge \vee \wedge ArcWeights[i][1] \in Places \\
&\quad \wedge ArcWeights[i][2] \in Transitions \\
&\quad \vee \wedge ArcWeights[i][1] \in Transitions \\
&\quad \wedge ArcWeights[i][2] \in Places \\
&\quad \wedge ArcWeights[i][3] \geq 1 \\
&\wedge \forall k \in DOMAIN \ Marking : k \in Places \wedge Marking[k] \geq 0 \\
&\wedge DOMAIN \ Marking = Places
\end{aligned}$$

$$Invariants \triangleq TypeInvariant \wedge ModelInvariant$$

## Operators

Hydrate a marking bag with all missing *Places* mapped to 0.

$$M^* \triangleq M @@ [p \in Places \mapsto 0]$$

(Input and output) places and transitions for transitions and places respectively.

$$\begin{aligned}
Inputs(v) &\triangleq \{k \in DOMAIN \ Arcs : v \in Arcs[k]\} \\
Outputs(v) &\triangleq \text{IF } v \in DOMAIN \ Arcs \text{ THEN } Arcs[v] \text{ ELSE } \{\}
\end{aligned}$$

Unspecified arc weights default to 1.

$$\begin{aligned}
ArcWeight(from, to) &\triangleq \text{LET} \\
\quad match(w) &\triangleq w[1] = from \wedge w[2] = to \\
\quad ws &\triangleq SelectSeq(ArcWeights, match)
\end{aligned}$$

IN

$$\text{IF } Len(ws) > 0 \text{ THEN } ws[1][3] \text{ ELSE } 1$$

$$\begin{aligned}
Enabled(t) &\triangleq \wedge t \in Transitions \\
&\wedge \forall p \in Inputs(t) : Marking[p] \geq ArcWeight(p, t)
\end{aligned}$$

$$\begin{aligned}
Fire(t) &\triangleq \wedge Enabled(t) \\
&\wedge Marking' = Marking \oplus \\
&\quad [p \in Inputs(t) \mapsto 0 - ArcWeight(p, t)] \oplus \\
&\quad [p \in Outputs(t) \mapsto ArcWeight(t, p)]
\end{aligned}$$

## Properties

$$Reachable(m) \triangleq \Diamond(Marking = m^*)$$

$$FinalMarking(m) \triangleq \Diamond \Box (Marking = m^*)$$

$$Bound(k) \triangleq \Box (\forall p \in \text{DOMAIN } Marking : Marking[p] \leq k)$$

Optional restrictions on the structure of Petri Nets.

$$\begin{aligned} IsStateMachine \triangleq & \quad \wedge \forall t \in Transitions : \wedge Cardinality(Inputs(t)) = 1 \\ & \quad \wedge Cardinality(Outputs(t)) = 1 \\ & \quad \wedge \Box (BagSum(Marking) = 1) \end{aligned}$$

$$\begin{aligned} IsMarkedGraph \triangleq & \quad \forall p \in Places : \wedge Cardinality(Inputs(p)) = 1 \\ & \quad \wedge Cardinality(Outputs(p)) = 1 \end{aligned}$$

$$\begin{aligned} IsFreeChoiceNet \triangleq & \quad \forall k \in \text{DOMAIN } Arcs \cap Places : \\ & \quad \vee Cardinality(Outputs(k)) = 1 \\ & \quad \vee \forall t \in Arcs[k] : Cardinality(Inputs(t)) = 1 \end{aligned}$$

TODO: implement more!

---

Spec

$$Init \triangleq Marking = InitialMarking^*$$

$$Next \triangleq \exists t \in Transitions : Fire(t)$$

$$Spec \triangleq Init \wedge \Box [Next]_{vars} \wedge (\forall t \in Transitions : WF_{vars}(Fire(t)))$$


---