

## Petri Nets.

Petri nets are one of the most common formalism to express a process model. A Petri net is a directed bipartite graph, in which the nodes represent transitions and places. Arcs are connecting places to transitions and transitions to places, and have an associated weight. A transition can fire if each of its input places contains a number of tokens that is at least equal to the weight of the arc connecting the place to the transition. When a transition is fired, then tokens are removed from the input places according to the weight of the input arc, and are added to the output places according to the weight of the output arc. A marking is a state in the Petri net that associates each place to a number of tokens and is uniquely associated to a set of enabled transitions that could be fired according to the marking. Process Discovery algorithms implemented in pm4py returns a Petri net along with an initial marking and a final marking. An initial marking is the initial state of execution of a process, a final marking is a state that should be reached at the end of the execution of the process.

### Importing and Exporting

```
import os
from pm4py.objects.petri.importer import importer as pnml_importer
net, initial_marking, final_marking = pnml_importer.apply(os.path.join("

from pm4py.visualization.petrinet import visualizer as pn_visualizer
gviz = pn_visualizer.apply(net, initial_marking, final_marking)
pn_visualizer.view(gviz)

from pm4py.objects.petri.exporter import exporter as pnml_exporter
pnml_exporter.apply(net, initial_marking, "petri.pnml")

pnml_exporter.apply(net, initial_marking, "petri_final.pnml", final_mark
```

Figure 1

Petri nets, along with their initial and final marking, can be imported/exported from the PNML file format. The code on the above i.e, figure 1 can be used to import a Petri net along with the initial and final marking.

First, we have to import the log. Subsequently, the Petri net is visualized by using the Petri Net visualizer. In addition, the Petri net is exported with its initial marking or initial marking and final marking

### Petrinet Properties

This section is about how to get the properties of a Petri Net. A property of the pet is, for example, a the enabled transition in a particular marking. However, also a list of places, transitions or arcs can be inspected.

The list of transitions enabled in a particular marking can be obtained using the below i.e, figure 2 and figure 3

```
from pm4py.objects.petri import semantics
transitions = semantics.enabled_transitions(net, initial_marking)
```

Figure 2

```
places = net.places
transitions = net.transitions
arcs = net.arcs
```

Figure 3

The function `print(transitions)` reports that only the transition register request is enabled in the initial marking in the given Petri net. To obtain all places, transitions, and arcs of the Petri net, the code which can be obtained on the right-hand side can be used.

```
places = net.places
transitions = net.transitions
arcs = net.arcs
```

Each place has a name and a set of input/output arcs (connected at source/target to a transition). Each transition has a name and a label and a set of input/output arcs (connected at source/target to a place). The code on the right-hand side prints for each place the name, and for each input arc of the place the name and the label of the corresponding transition. However, there also exists `trans.name`, `trans.label`, `arc.target.name`.

```
for place in places:
    print("\nPLACE: "+place.name)
    for arc in place.in_arcs:
        print(arc.source.name, arc.source.label)
```

Figure 4