Process Mining

Kristian Wichmann

January 17, 2018

1 Event logs

An *event log* is a collection of data where each row contains at least the following:

- A *case id* corresponding to a particular unit whose processes we wish to discover.
- An *event id*, to denote a specific kind of activity.
- A *timestamp*, which allows us to order the events.

2 Model: Petri nets

2.1 Petri net graphs

A *Petri net graph* is a tuple (P, T, F). Here P is a collection of *places* and T a collection of *transitions*, P and T disjoint. The *flow relations* $F \subseteq (P \times T) \cup (T \times P)$ corresponds to place inputs ($I \subseteq P \times T$) and outputs ($O \subseteq T \times P$) to transitions.

The graphical representation of a petri net graph is to draw places as circles, transitions as thick lines or squares, and flow relations as arrows from places to transitions. As an example, figure 1 is the graphical representation of the following Petri net graph:

- $P = \{P1, P2, P3, P4\}$
- $T = \{T_1, T_2\}$
- $F = I \cup O$, where:

$$I = \{\{P1, T1\}, \{P2, T2\}, \{P3, T2\}\} \tag{2.1}$$

$$O = \{\{T1, P2\}, \{T1, P3\}, \{T2, P1\}, \{T2, P4\}\} \tag{2.2}$$

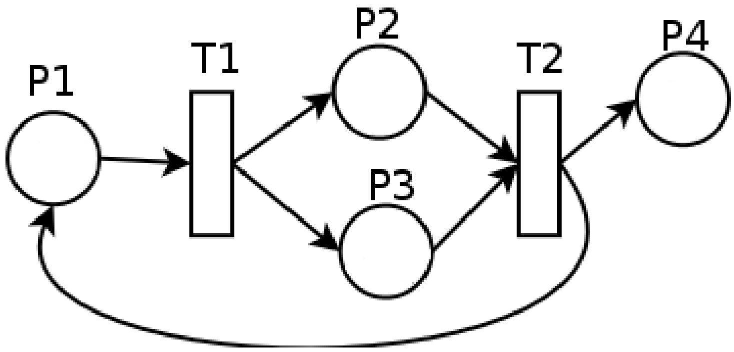

Figure 1: An example of a Petri net graph.

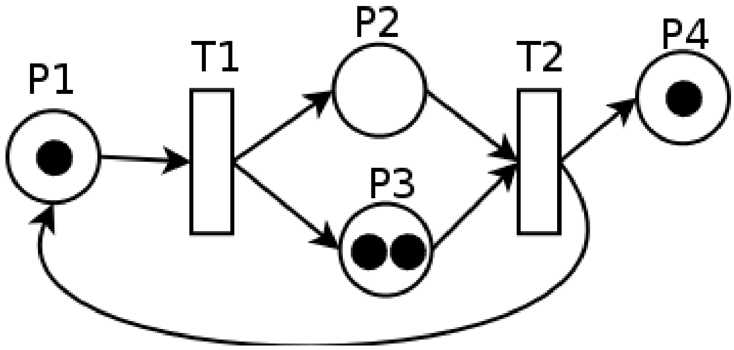

Figure 2: An example of a Petri net.

Sometimes, we may wish to allow more than one arrow between a place-transition/transition-place pair. In this case, F instead becomes a function:

$$F : (P \times T) \cup (T \times P) \to \mathbb{N}_0 \tag{2.3}$$

The value of a pair is then the number of arrows. Again, this could be split into input and output functions:

$$I : P \times T \to \mathbb{N}_0, \quad O : T \times P \to \mathbb{N}_0 \tag{2.4}$$

2.2 Markings and Petri nets

A *marking* M on a Petri net graph, is a function:

$$M : P \to \mathbb{N}_0 \tag{2.5}$$

This is interpreted as the number of *tokens* present at each place in the Petri net graph. For instance, figure 2 shows the following marking on the example

Petri net graph from the section above:

$$P1 \mapsto 1, \quad P2 \mapsto 0, \quad P3 \mapsto 2, \quad P4 \mapsto 1 \tag{2.6}$$

A collection of a Petri net graph (P, T, F) and an initial marking M_0 on it is called a *Petri net*.

2.3 Rules of the "game"

From the initial marking, a Petri net can undergo the transitions in T under certain circumstances.

A transition $t \in T$ is *enabled*, if all the places which has an input flow relation pointing to it all has at least one token[1].

An enabled transition may *fire*, which means that the current marking changes, such that each token from the input places are removed, while all the output places gets a token[2].

Transitions always happens in a specific order, so two transitions cannot fire at the same time. This means that we may write a *trajectory* of the Petri net as:

$$M_0, M_1, M_2, \ldots, \quad \langle t_1, t_2, \cdots \rangle \tag{2.7}$$

Here, all the M_i's are markings, and between M_i and M_{i+1} transition t_{i+1} fires. Such a trajectory may or may not be finite in length.

The set of all markings that are *reachable* for a Petri net N is denoted $R(N)$. The elements of $R(N)$ forms a directed *reachability graph*, with transitions as edges.

2.4 Properties of Petri nets

A marking where no transitions are enable is called *deadlocked*. A Petri net with a deadlocked marking in its reachability graph is said to have a *potential deadlock*.

A transition in a Petri net that can never fire, i.e. is not an edge anywhere in the reachability graph is called *dead* or L_0-*live* (a designation that will become clear in a while). A transition that is not dead has some degree of being alive, as shown below:

[1] If several arrows are allowed, each place has to hold at least a number of tokens corresponding to the number or arrows.

[2] Again, if several arrows are allowed, several tokens are removed/added according to the multiplicity of arrows.

- A transition is L_1-*live* if it is not dead, i.e. if is occurs as an edge somewhere in the reachability graph. This is also called **potentially fireable**.
- A transition is L_2-*live* if for any positive integer k there is a reachable marking in which it occurs at least k times on a trajectory to the marking. (The trajectories can be different for each k).
- A transition is L_3-*live* if there is a trajectory in which the transition occurs infinitely often.
- A transition is L_4-*live* - or simply *live*, if for any reachable marking, the transition is L_1-live. I.e. no matter where in the reachability graph we are, it is always possible to fire the transition some time in the future.

These conditions are progressively stronger, so usually only the highest index is used to refer to a given transition.

A place p in a Petri net is called k-*bounded* if all markings in the reachability graph has as most k tokens on it. A place that is 1-bounded is called *safe*. A place that is k-bounded for any positive value of k is called *bounded*.

A Petri net in which all places are (k-)bounded is called (k-)*bounded*. A Petri net in which all places are safe is called *safe*. A Petri net graph in which all possible initial markings are bounded is called *structurally bounded*.

2.5 Workflow nets

A *workflow net* (or WF net for short) is a Petri net which has a start or *source* place and an ending or *sink* place, so that the source has no inputs, and the sink no outputs. It is used to model processes with transitions representing events.

3 The alpha algorithm

4 Model: Dependency graph

A *dependency graph* is a series of nodes - representing events - in which there are causal arrows between events. So as such, this is simply a directed graph.

5 Heuristics mining