Petri net challenge

A digraph (directed graph) consists of a set N of *nodes* and a relation $\rightarrow \subseteq N \times N$. We write $n \rightarrow m$ if $(n,m) \in \rightarrow$ and $n \not\rightarrow m$ if $(n,m) \notin \rightarrow$. We want to find out for a given digraph $(N, \rightarrow)$ whether it is possible to partition N, i.e. find A, B, c, d such that $N = A \cup B \cup \{c, d\}$ with A, B nonempty and moreover

$\forall n, m : n \in A \wedge m \in B : n \not\rightarrow m \wedge m \not\rightarrow n$
$\forall n : n \in A : d \not\rightarrow n$
$\forall m : m \in B : c \not\rightarrow m.$

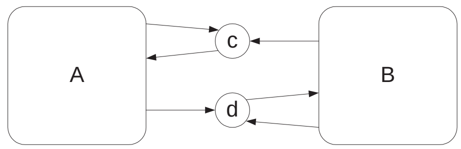

Figure 1: A digraph with partitioning

Figure 1 gives a graphical representation of the above conditions. The challenge is to find (reasonably) efficient algorithms to decide whether a digraph can be partitioned and, if so, to find them.

Hint: It may be handy to use distances between nodes as an auxiliary notion. We set $d(n, n) = 0$ and $d(n, m) = 1 + \text{MIN}\{d(n, m') \mid m' \rightarrow m\}$ if $n \neq m$.

Algorithms that meet the challenge may be self-developed or may be obtained from literature; there is extensive literature on graph algorithms.

We want to apply these graph algorithms to Petri nets in the tool Yasper (www.yasper.org). Petri nets are digraphs with two classes of *nodes*, namely *places* and *transitions*. The graph is *bipartite*; places are only connected to transitions and vice versa. So, a Petri net can be represented as a triple $(P, T, \rightarrow)$, where P is the set of places, T the set of transitions and $\rightarrow \subseteq (P \times T) \cup (T \times P)$ is called the set of arcs. Places are represented as circles and transitions as squares like in Figure 2. The left hand diagram is a Petri net with $P = \{p, q, r\}, T = \{t, u, v\}, \rightarrow = \{(t, p), (p, v), (u, p), (u, q), (q, w), (w, r), (r, v)\}$. A partitioning of that net is $A = \{t, u, v, p\}, B = \{w\}, c = r, d = q$.

The right-hand diagram is not a Petri net, because a transition is connected to a transition.

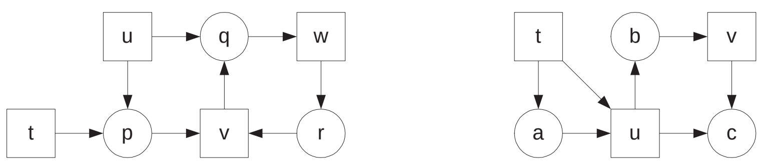

Figure 2: Examples related to Petri nets

Petri nets are used to model dynamic processes (e.g. systems for railway control). These models are analyzed by LaQuSo (www.laquso.nl) in order to find design errors. Partitioning a net can help to speed up analysis. It is allowed to make use of the bipartite character of Petri nets to find or speed up your algorithm. It is also possible to develop algorithms for certain subclasses of nets.