FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Real Time Systems – SS2016

Prof. Dr. Karsten Weronek

Faculty 2

Computer Science and Engineering

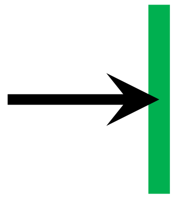
Petri Nets 3:

Variants

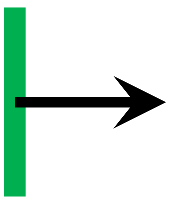
Properties

Examples

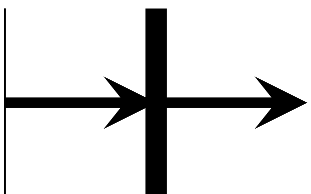
Elements



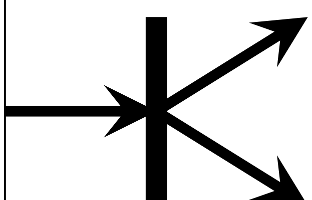
Deletion of Objects (Löschung)



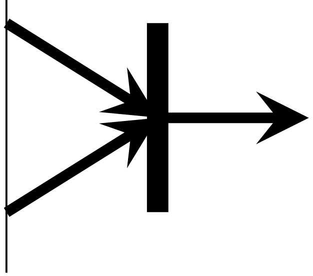
Generation of Objects (Erzeugung)



Transfer of Objects (Weitergabe)

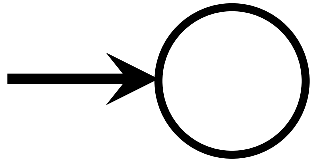


Split of Objects (Aufspaltung)
Begin of concurrency (Nebenläufigkeit)

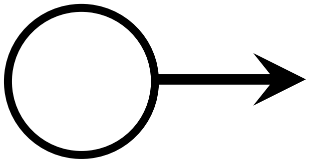


Join/Rendevouz of Objects (Verschmelzung)
End of concurrency

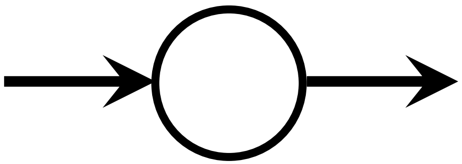
Elements



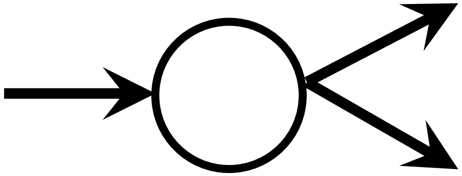
Sink, archiving of Objects (Senke)



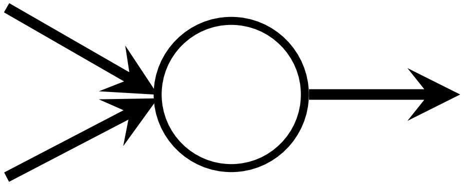
Source of Objects (Reservoir)



Transfer of Objects (Zwischenablage)

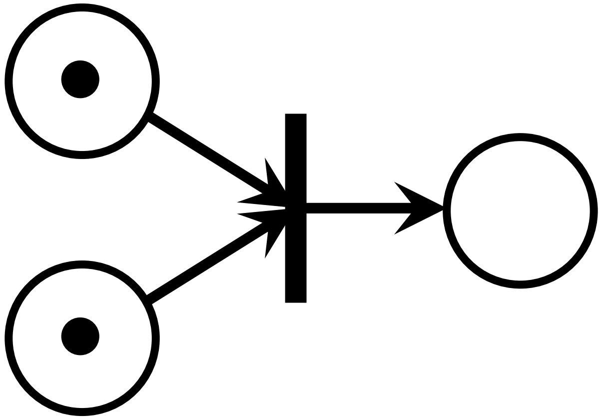


Non-deterministic Fork/Branch (willkürliche Verzweigung)

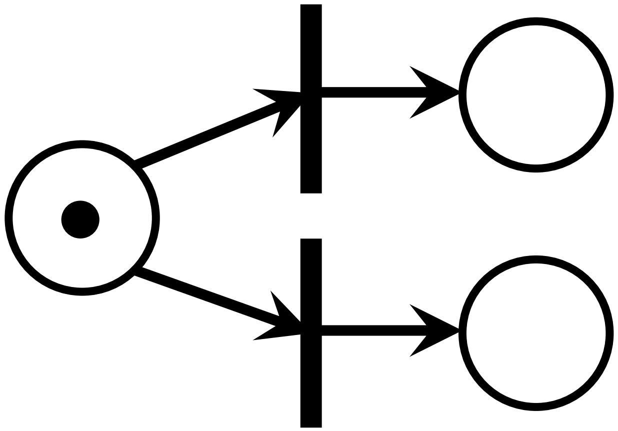


Common Meet for Objects,
Synchonising point of Objects (Gemeinsamer Speicher)

Elements

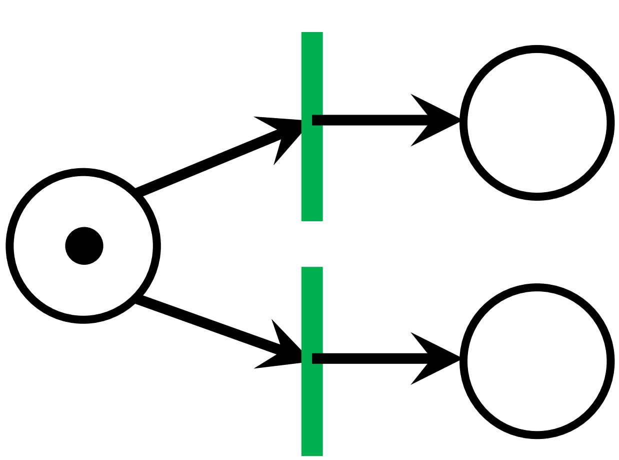


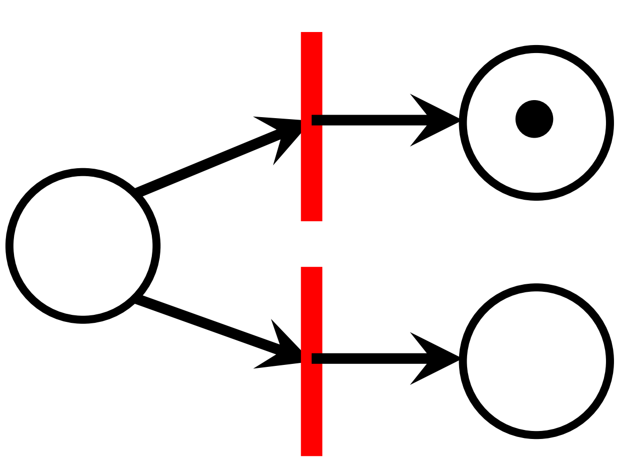
Synchronisation

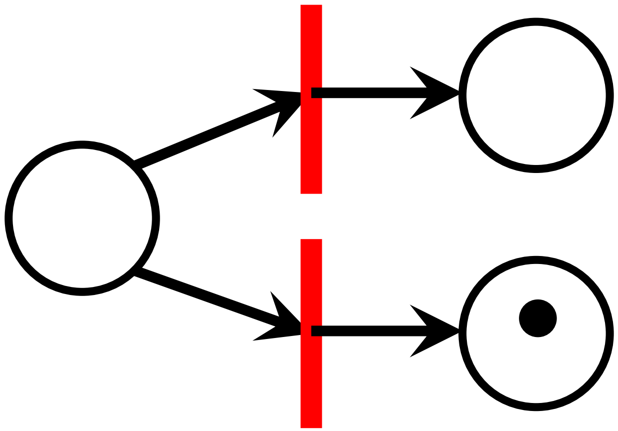


Alternative (non-deterministic)

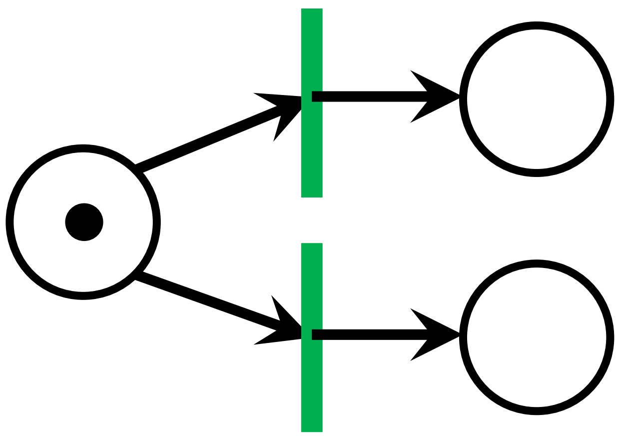
Conflicts (1a/2): Alternative

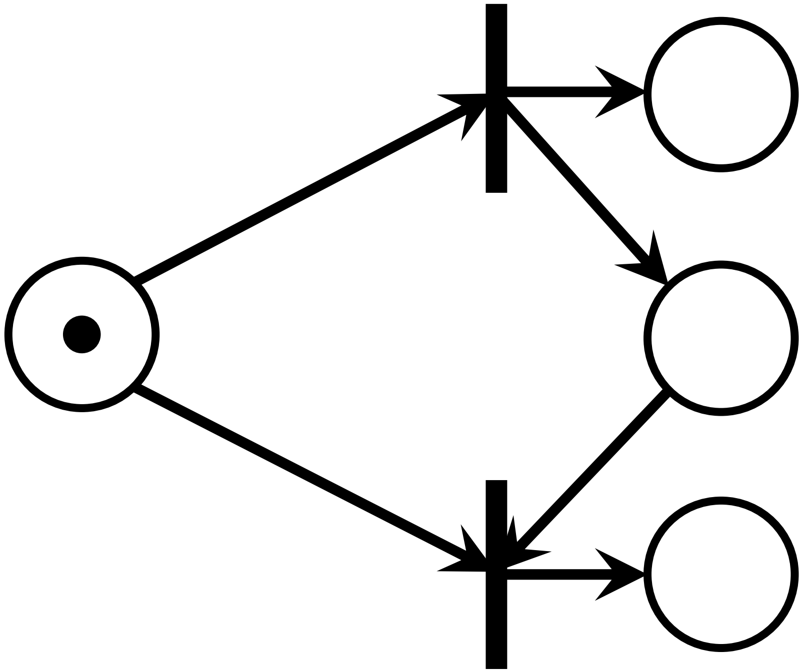




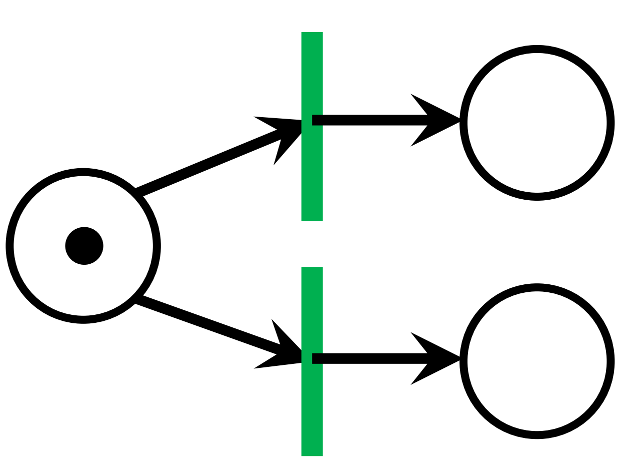


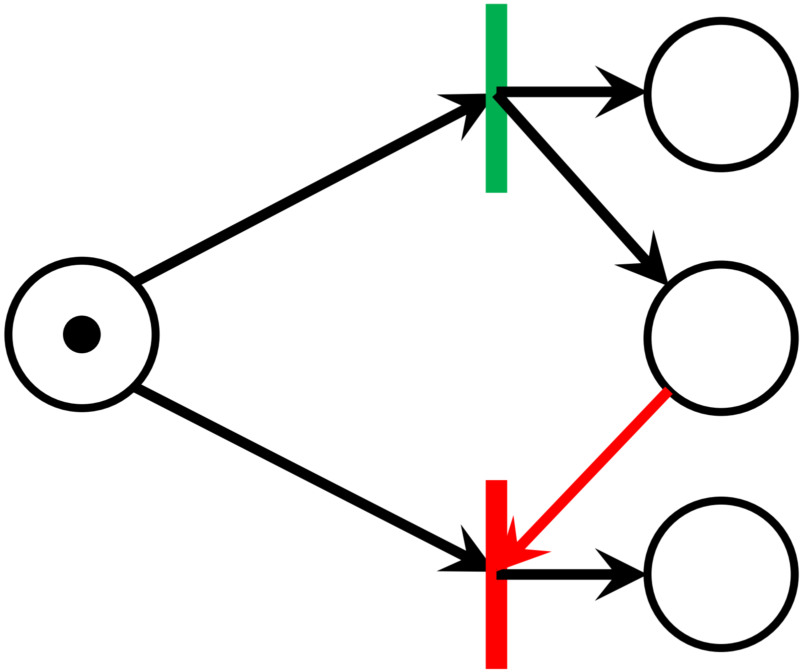
Conflicts (1a/2): Alternative: Prioritisation



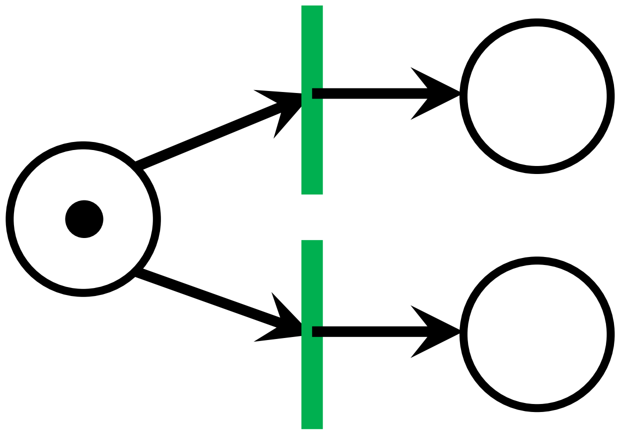


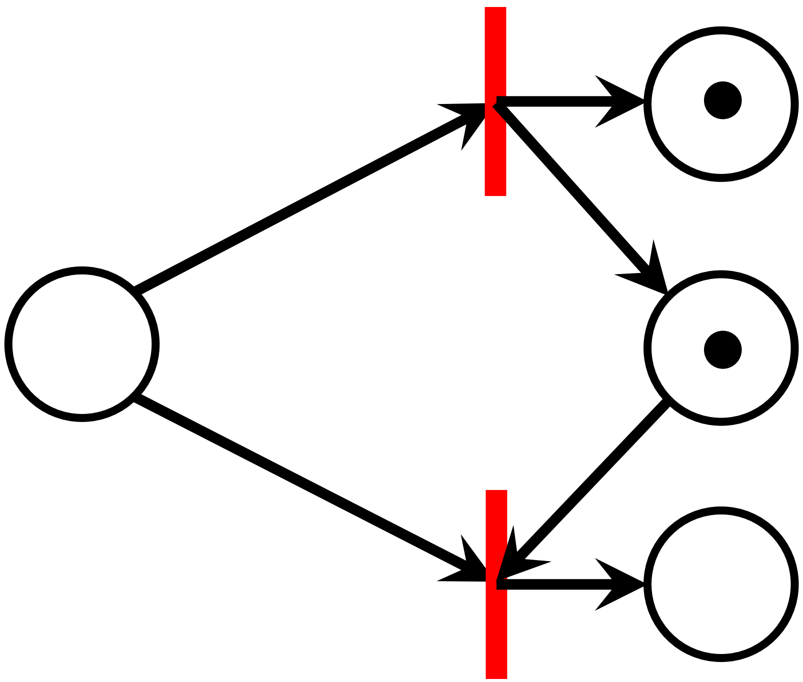
Conflicts (1a/2): Alternative: Prioritisation



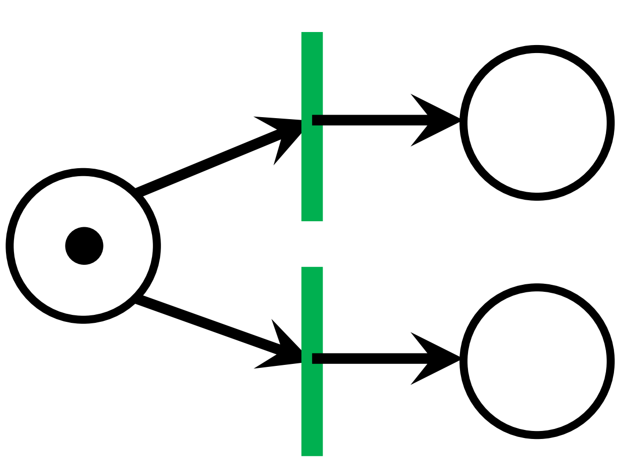


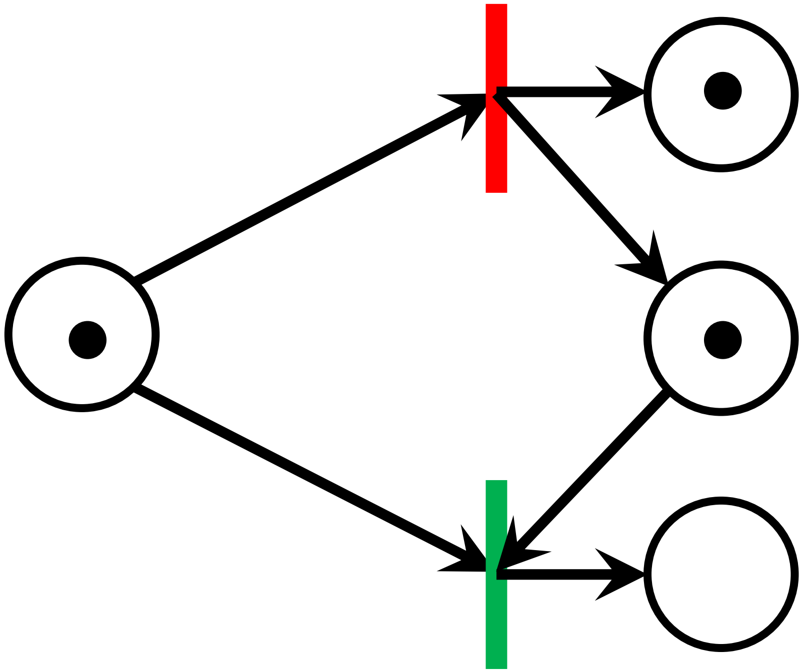
Conflicts (1a/2): Alternative: Prioritisation



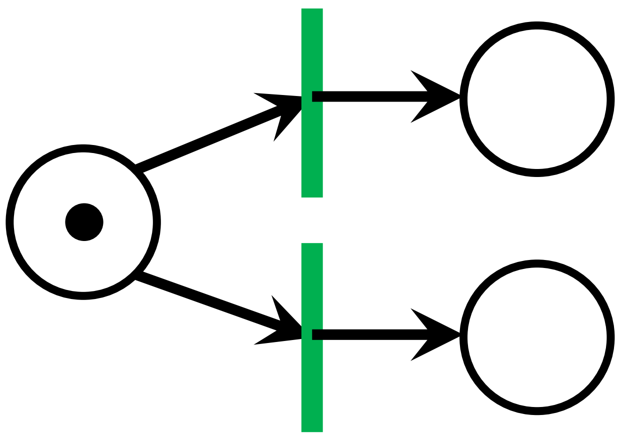


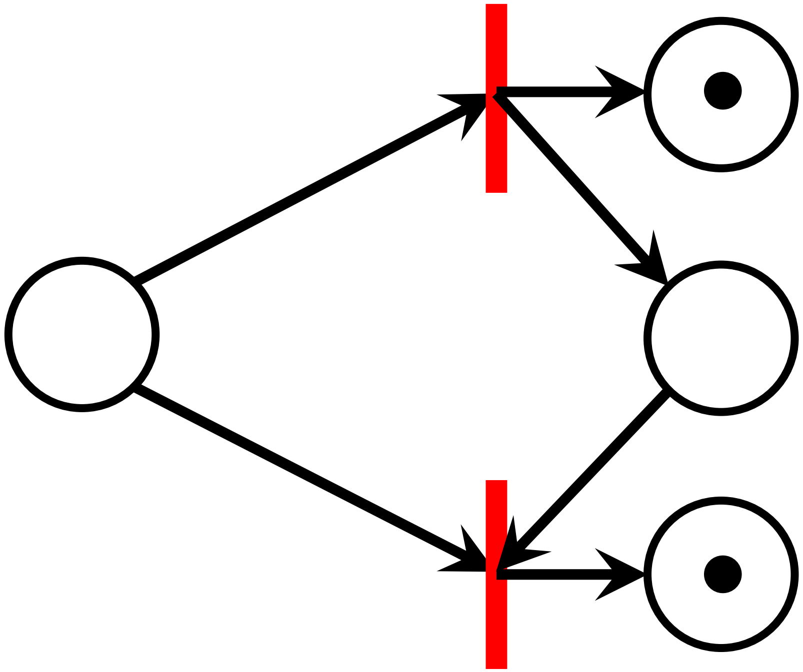
Conflicts (1a/2): Alternative: Prioritisation





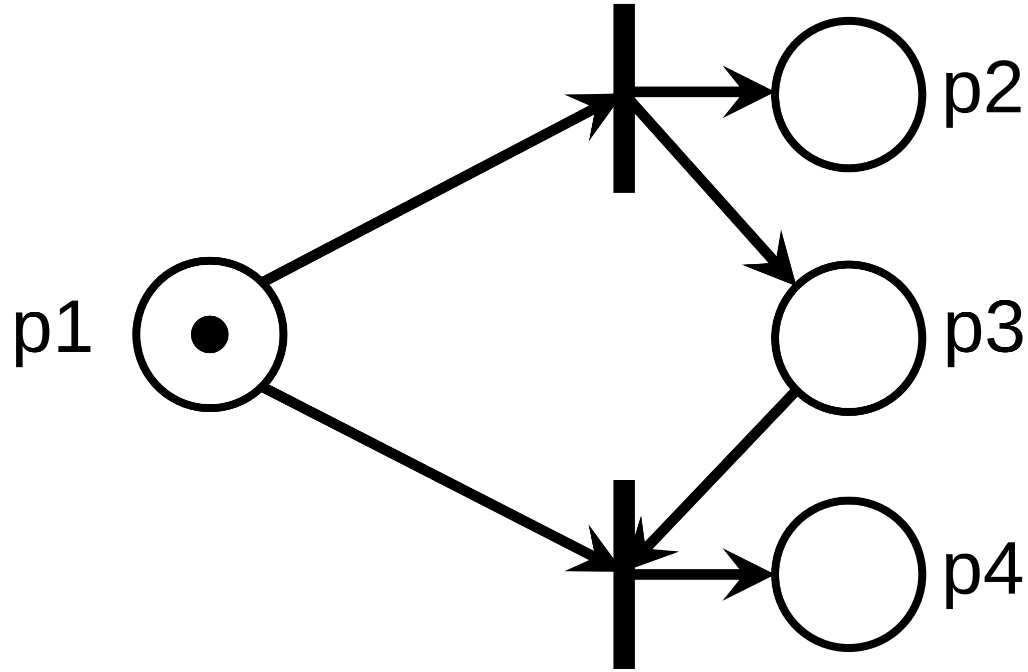
Conflicts (1a/2): Alternative: Prioritisation





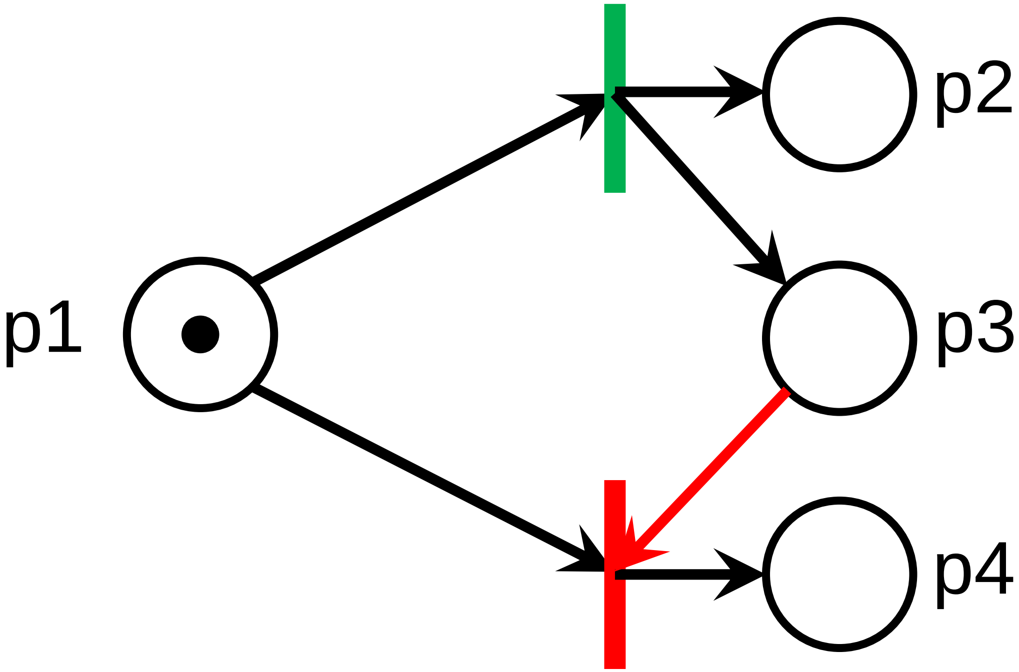
Vectorrepresentation

M= (p1, p2, p3, p4)
(1, 0, 0, 0)



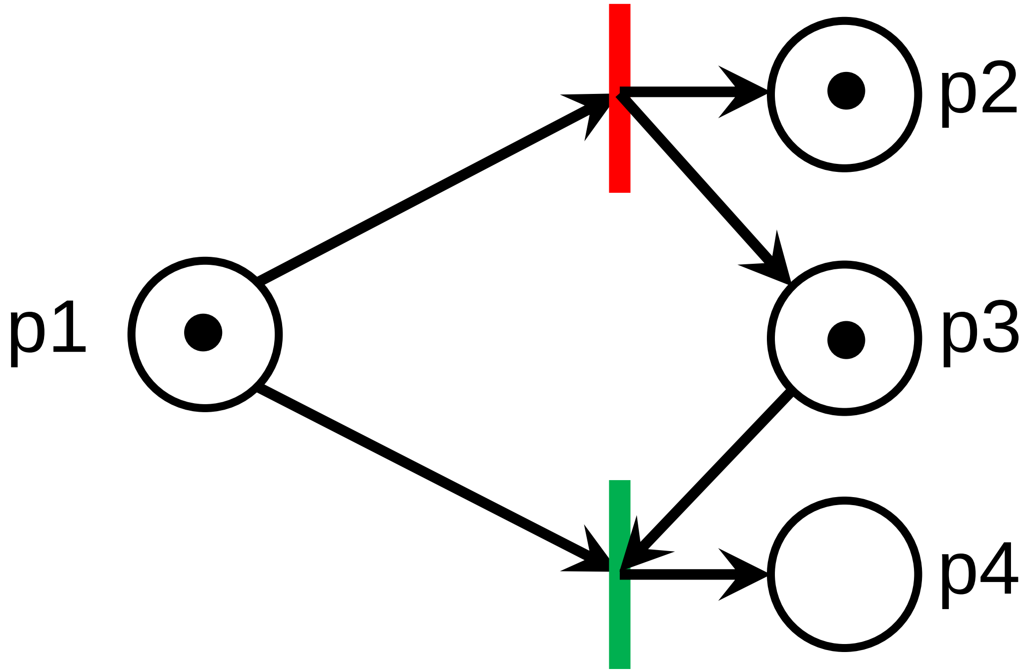
Vectorrepresentation

$M0 = (1, 0, 0, 0)$



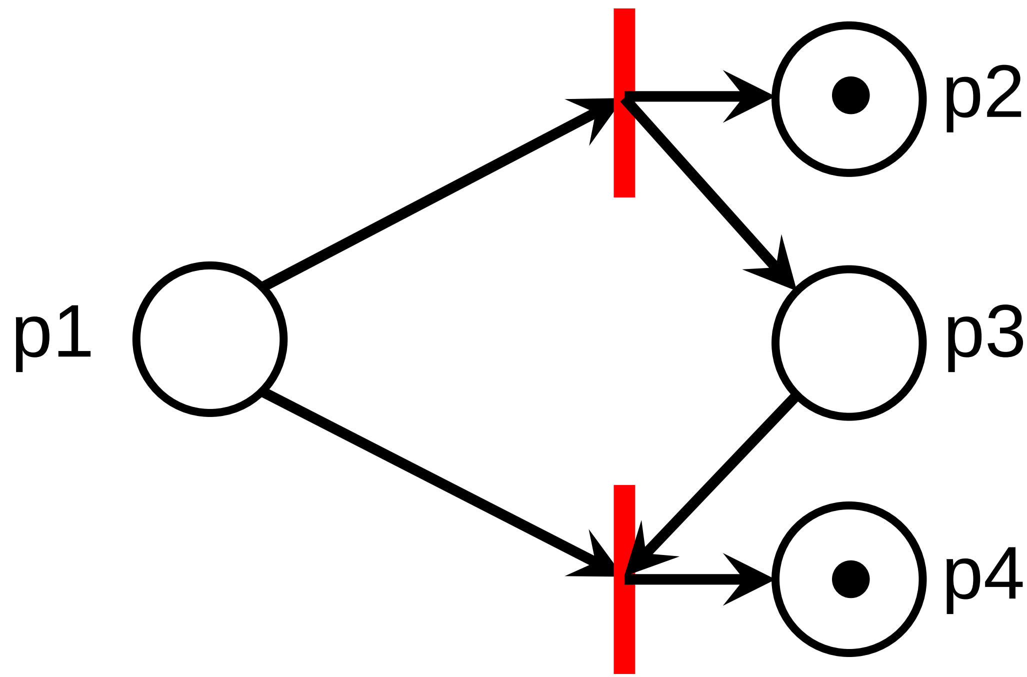
Vectorrepresentation

M1 = (1, 1, 1, 0)

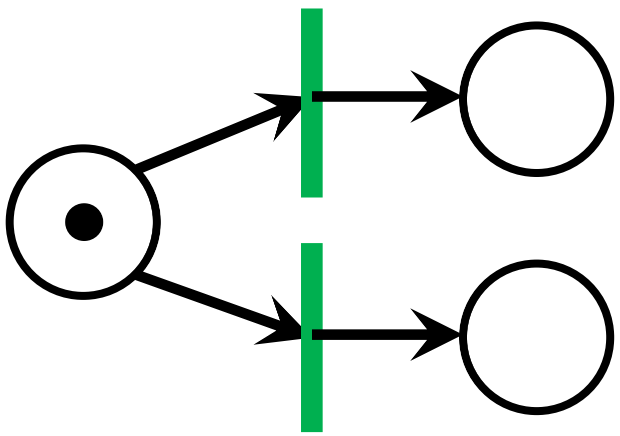


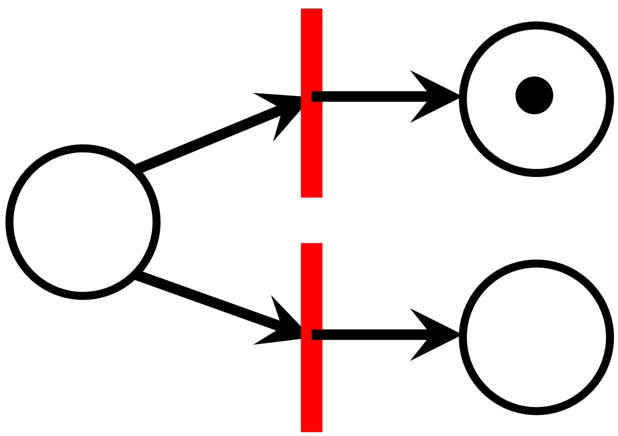
Vectorrepresentation

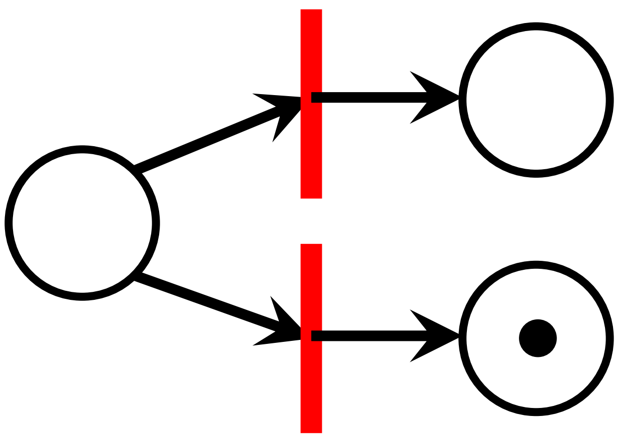
M = (0, 1, 0, 1)



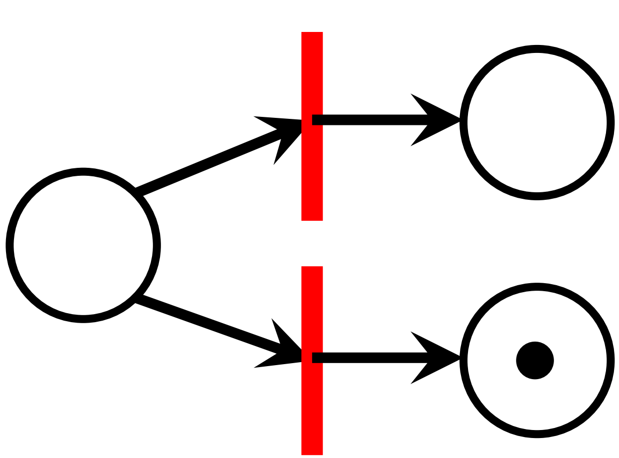
Conflicts (1a/2): Alternative

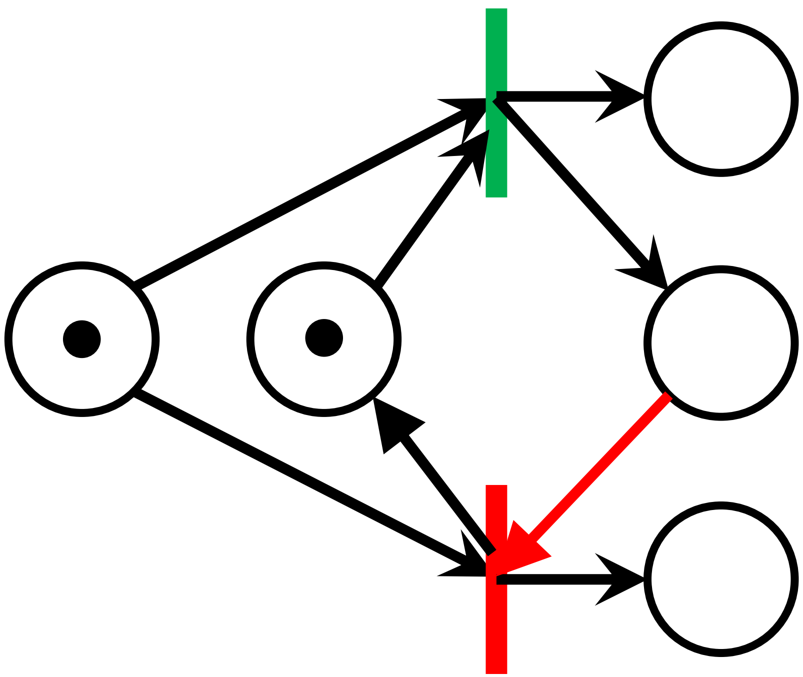




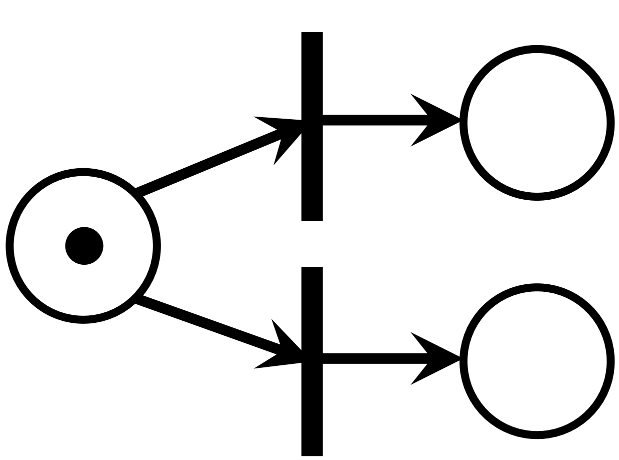


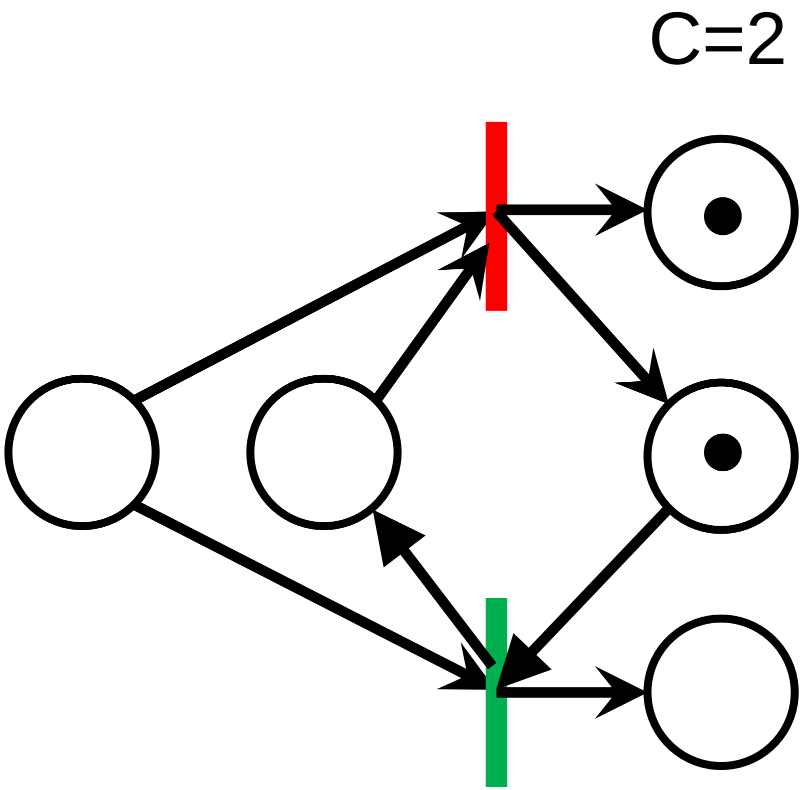
Conflicts (1b/2): Alternative: Toggle



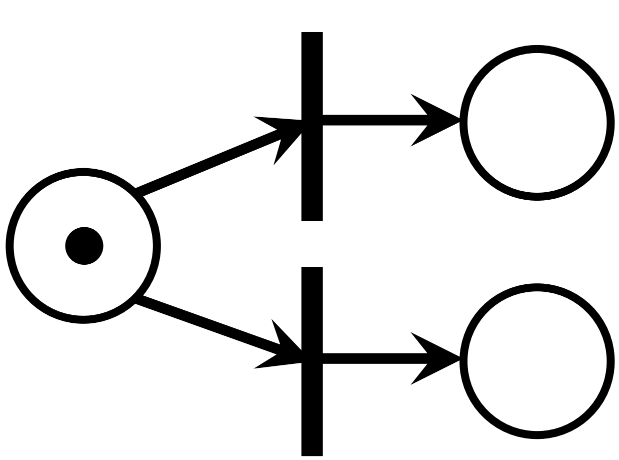


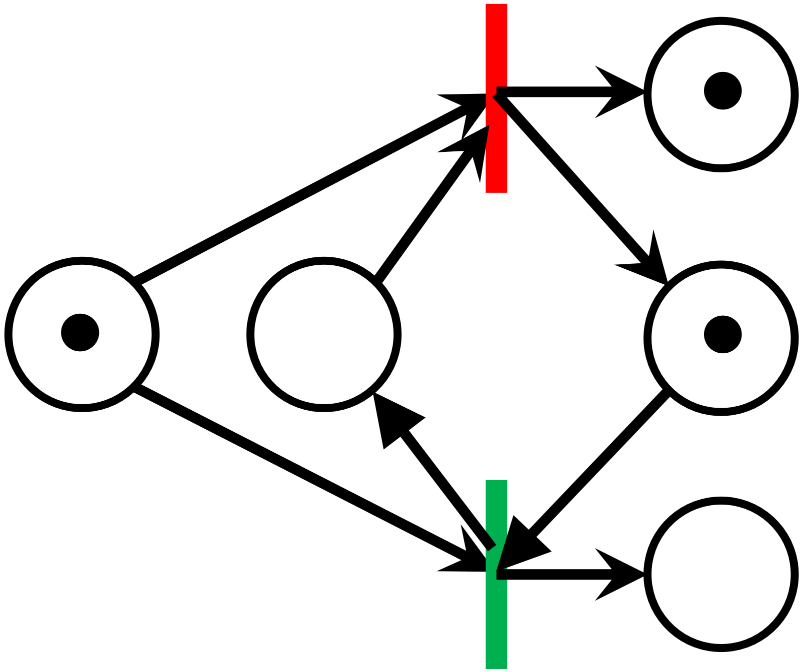
Conflicts (1b/2): Alternative: Toggle



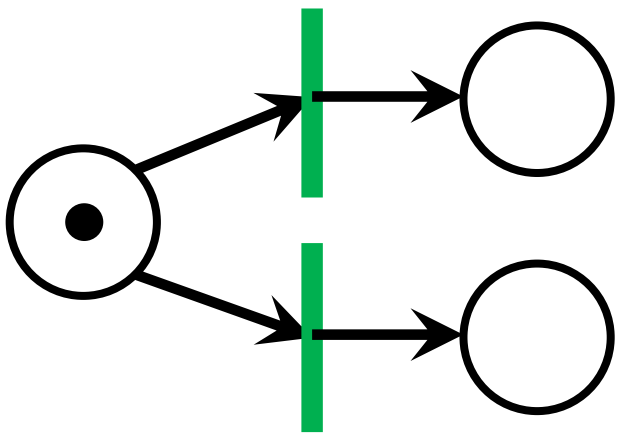


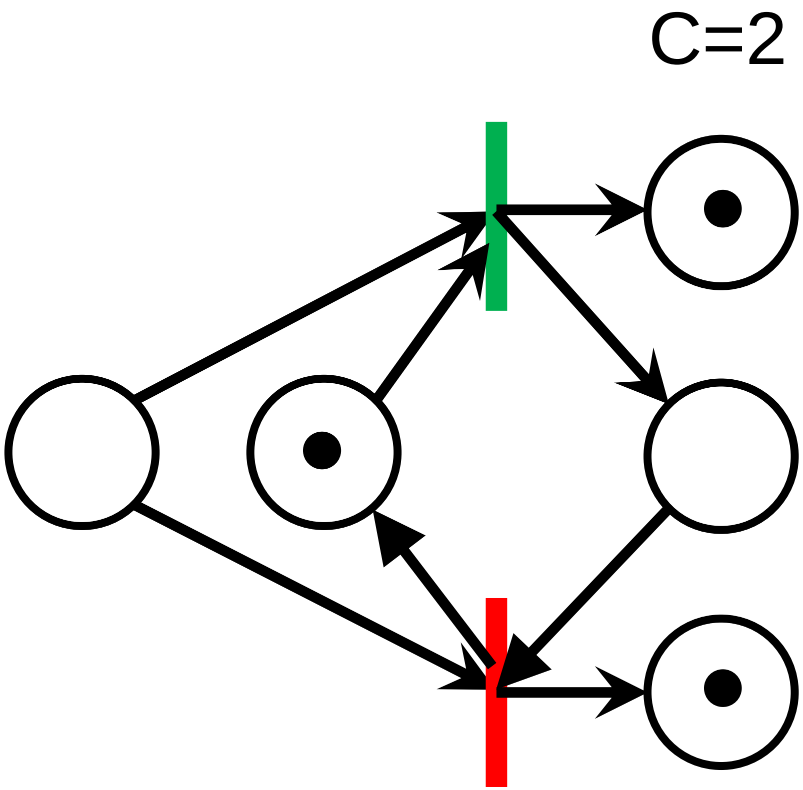
Conflicts (1b/2): Alternative: Toggle



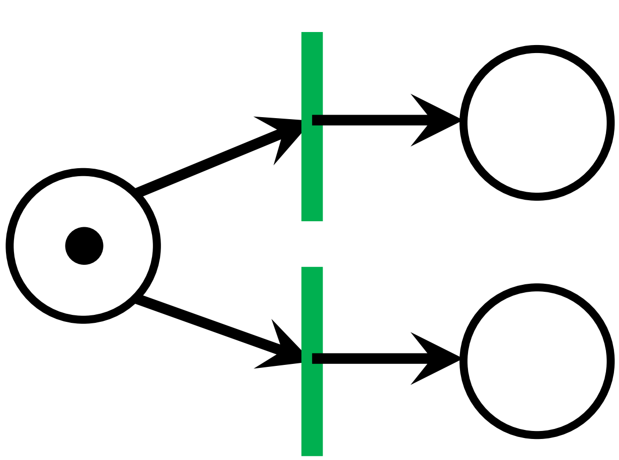


Conflicts (1b/2): Alternative: Toggle

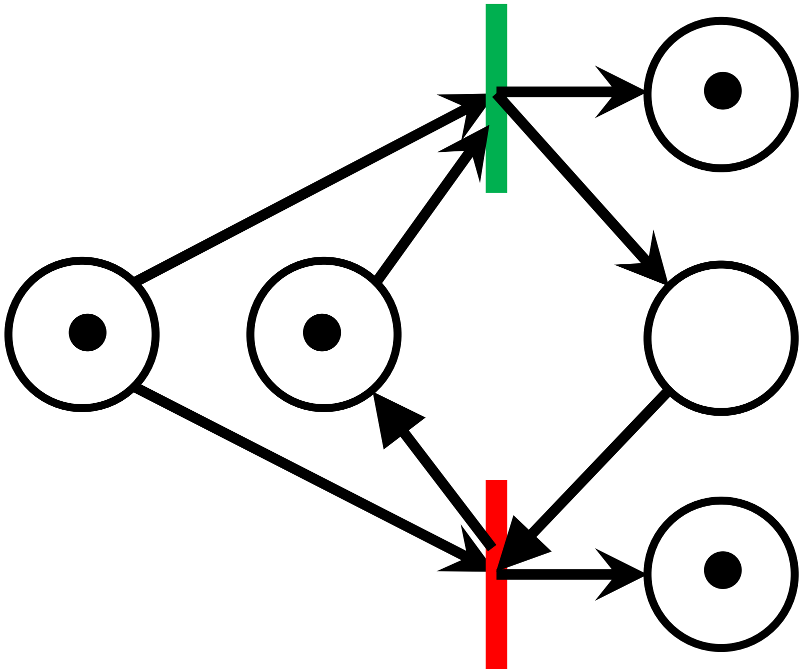




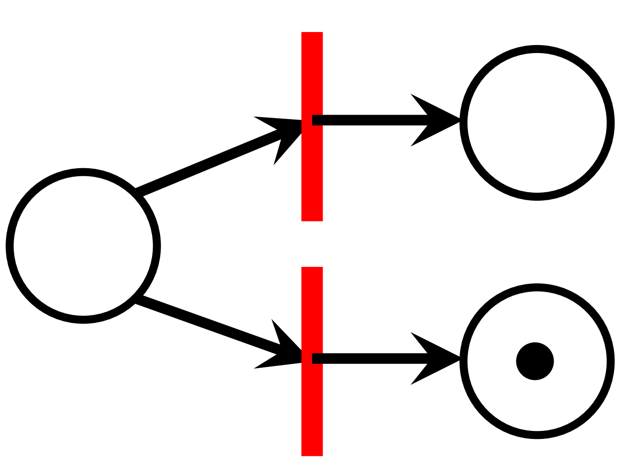
Conflicts (1b/2): Alternative: Toggle

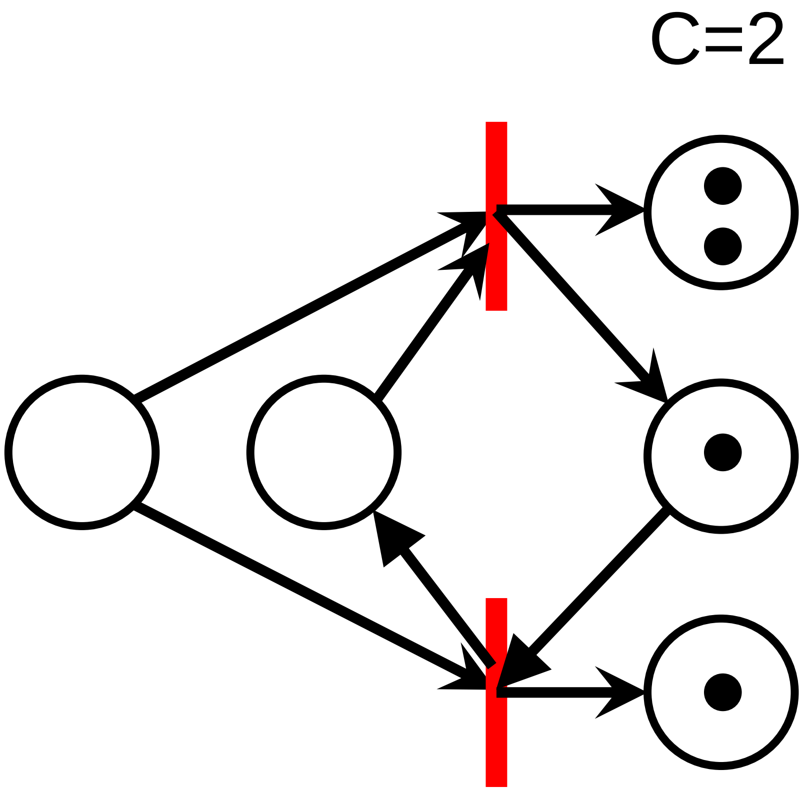


C=2

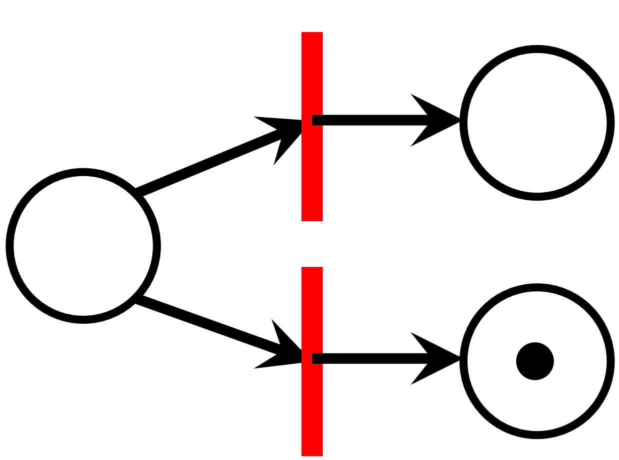


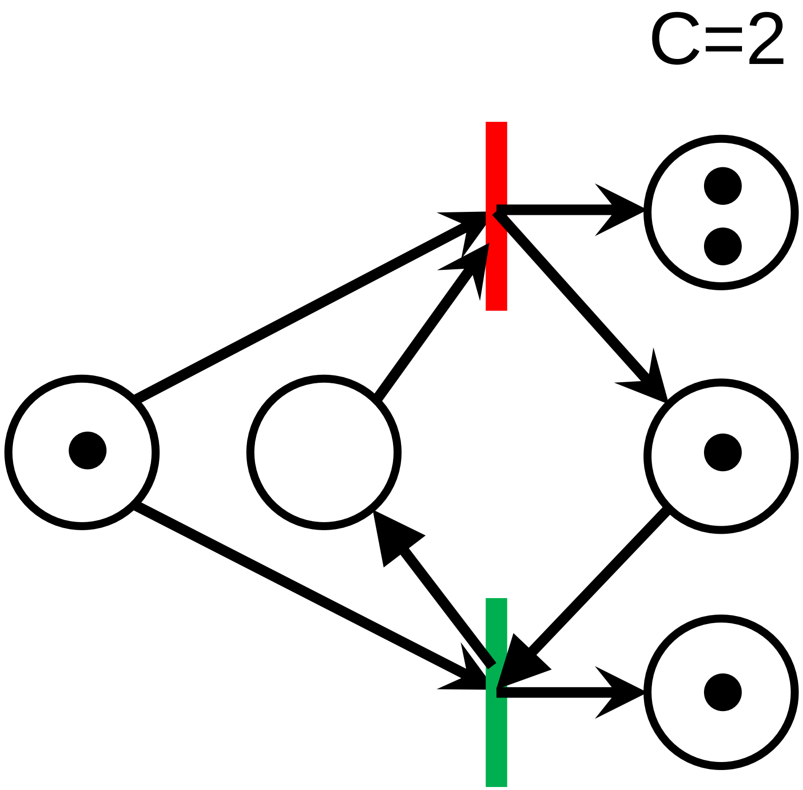
Conflicts (1b/2): Alternative: Toggle



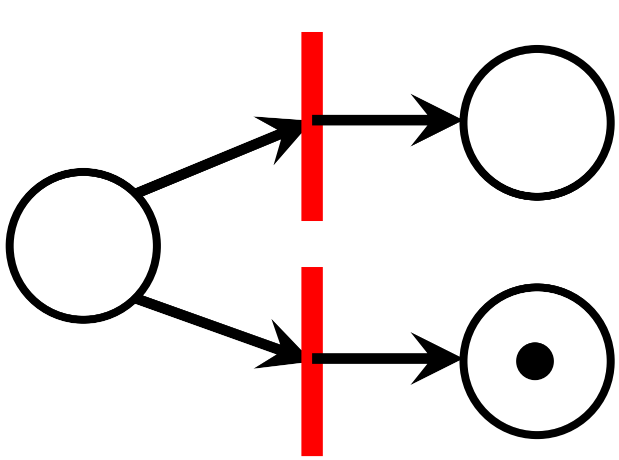


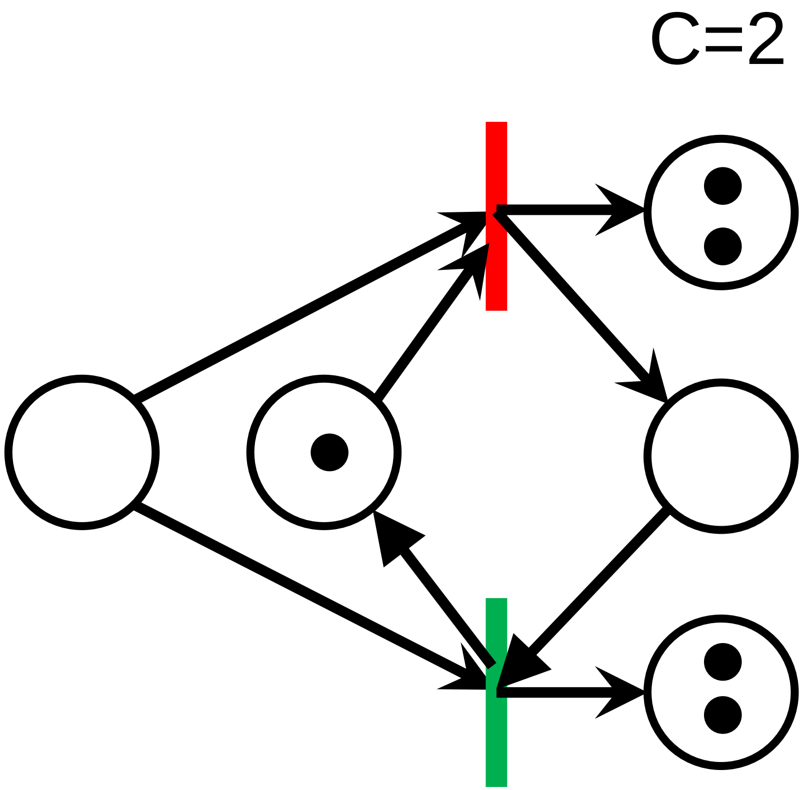
Conflicts (1b/2): Alternative: Toggle



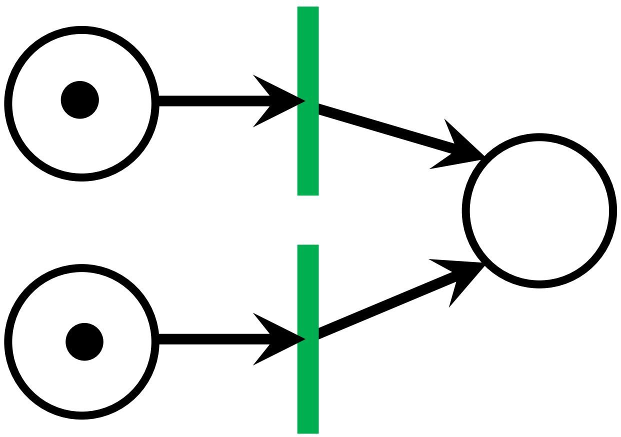


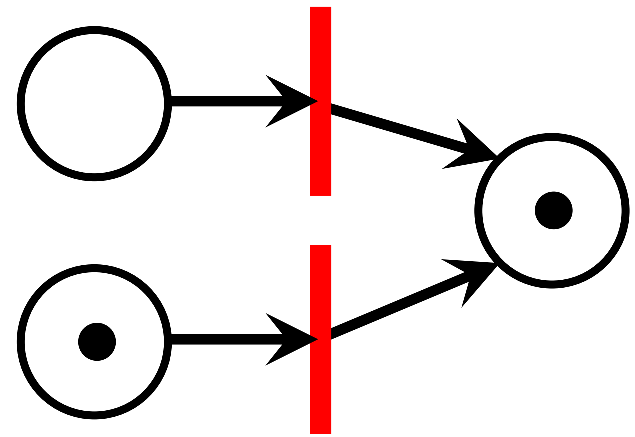
Conflicts (1b/2): Alternative: Toggle



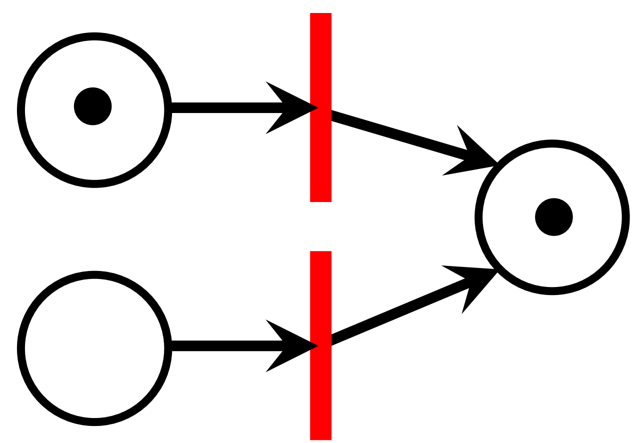


Conflicts (2/2): Synchronisation

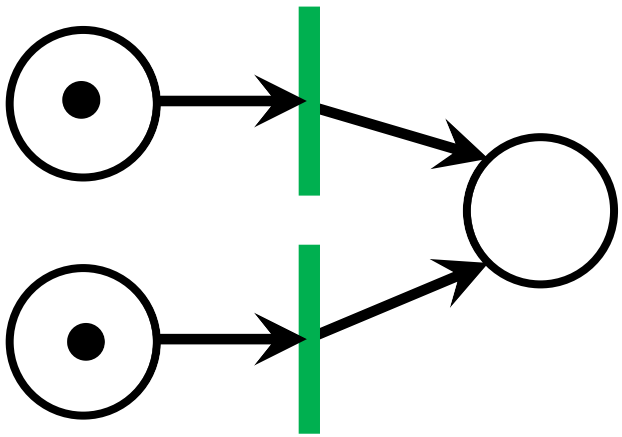


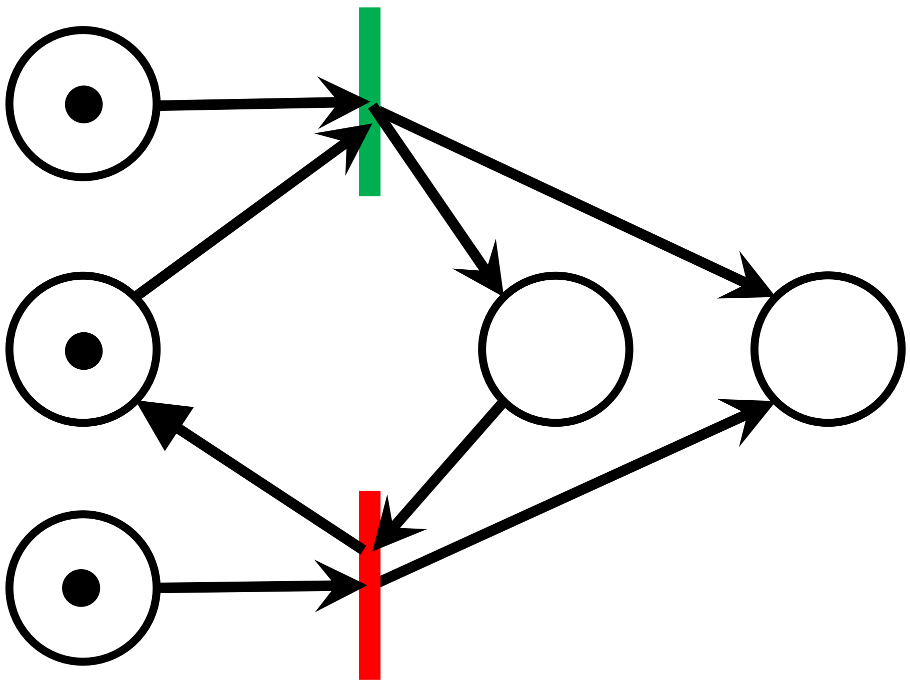


Conflicts (1b/2): Alternative: Toggle

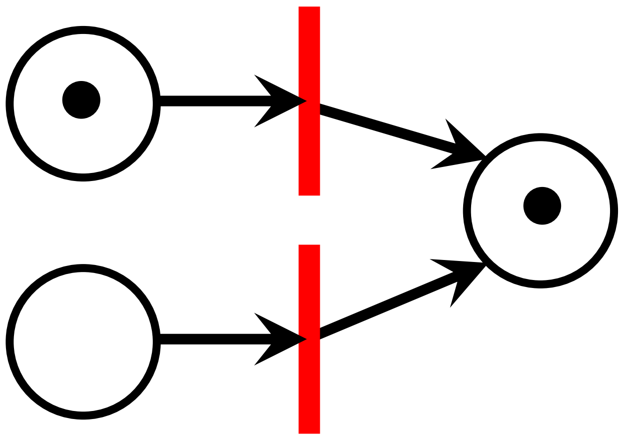


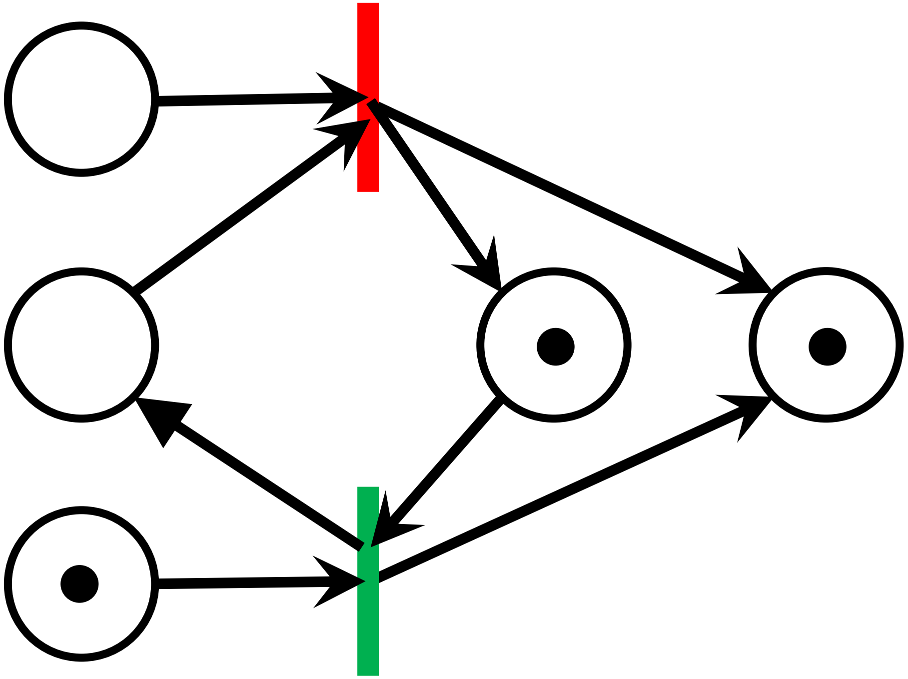
Conflicts (2/2): Synchronisation



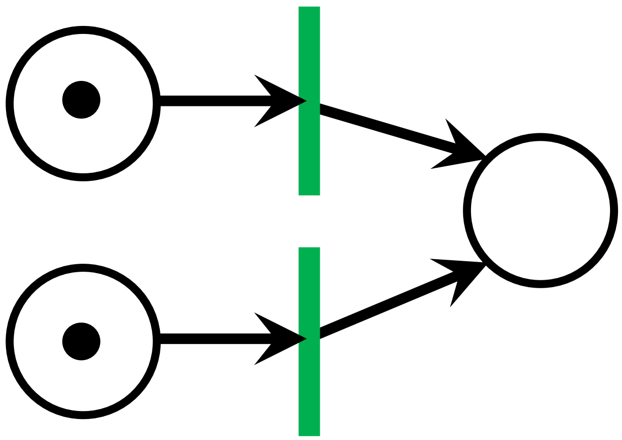


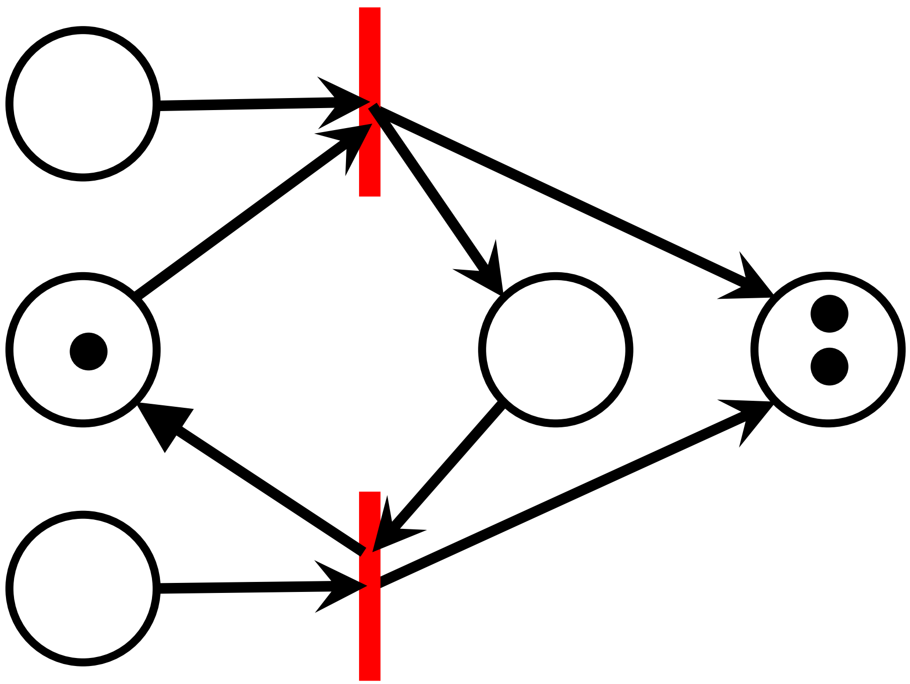
Conflicts (2/2): Synchronisation





Conflicts (2/2): Synchronisation





Persitence

A net is called **persistent**,
if there is no two transition conflict for any reachable transition.

Patterns

Patterns and Examples are in:

Van der Aalst: Classical Petricnets

You will find the link in Moodle.

Petri net (one formal definition)

Petri net

A Petri net is a 6-tuple G(P,T,F,K,W,M0) with:

- P : is a finite set of places
- T : is a finite set of transitions
- F : is a set of flow relations [$F = F_i \cup F_o = (TxP_i) \cup (PxT_o)$] *)
- C : $P \rightarrow N$ capacity of the places
- W : $F \rightarrow N$ weight of the arcs
- M0 : initial marking

A transition t in a petri net may fire(switch), if

1. all input places have enough tokens <u>and</u>
2. there are enough free places at the destination places.

*) some authors use: $F = F_i \cup F_o = (TxP_i) \cup (TxP_o)$

Analysibility of Petri nets

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

A Petri net has a good analysability, e.g.

1. Boundedness (number of tokens is limited)
2. Liveness (free of deadlocks)
3. Reachability

Petri net: 1. Boundedness

Boundedness:

- A place is bounded, if it has only one token either at the initial marking m_0 or at all reachable markings.
- A net is bounded, if all places are bounded.
- A place is called k-bounded , if
  it has only k tokens
  either at the initial marking m_0 or
  at all reachable markings.

The number of tokens is limited:

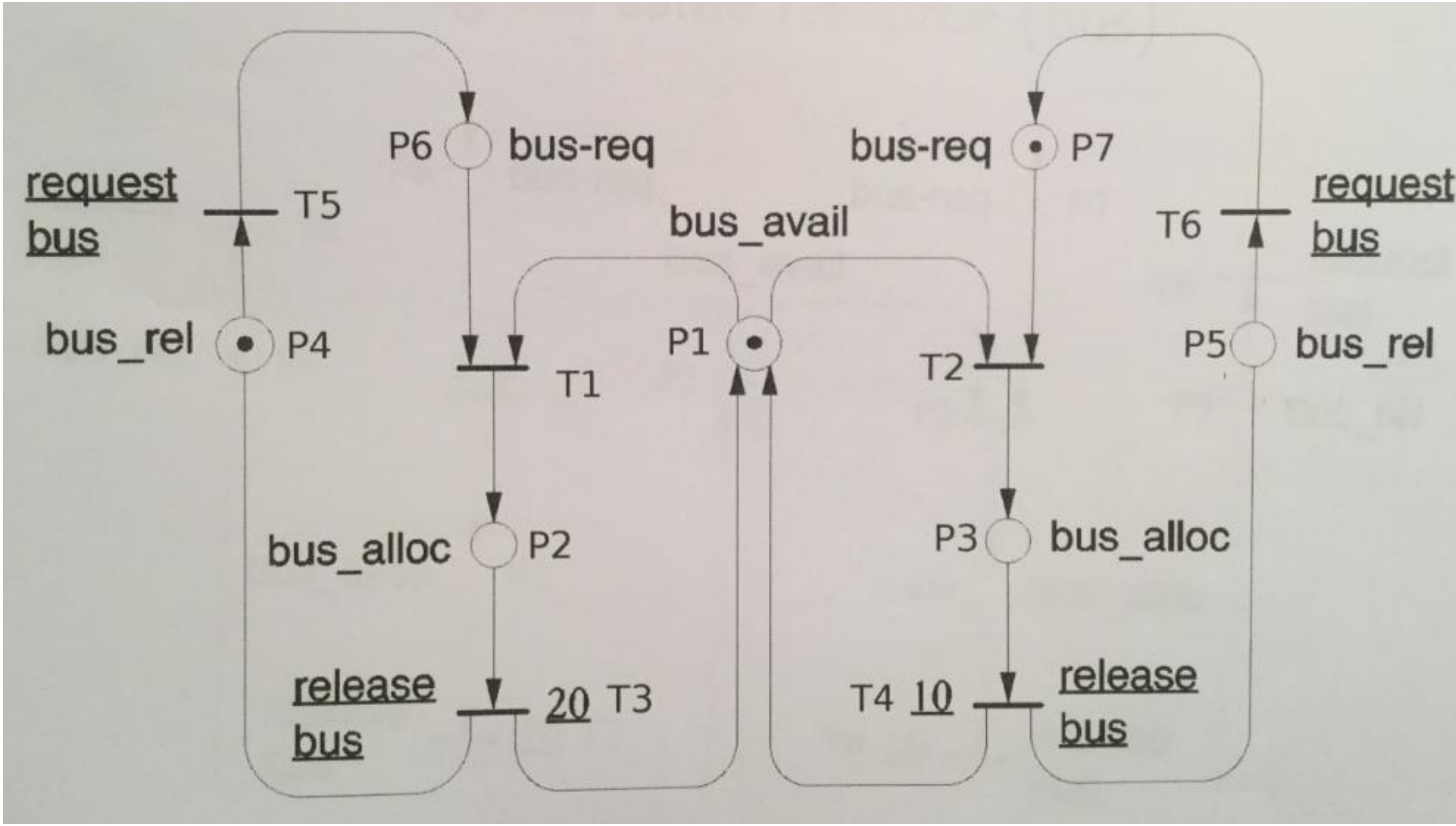
- No feedback or
- Feedback and
  the number of distributions is lower or
  equal to the number of mergers

Liveness:

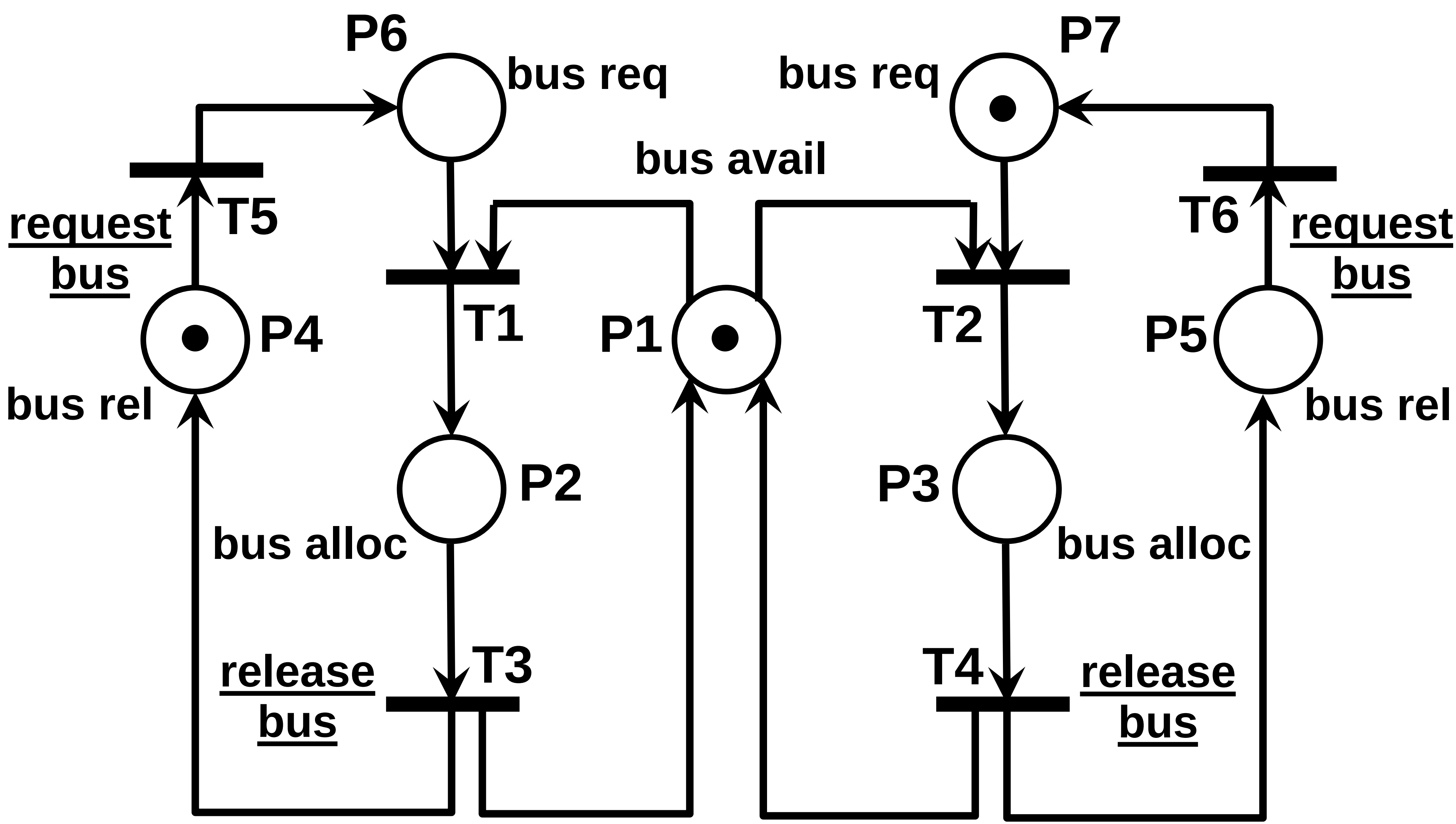
- Liveness (free of deadlocks)

- Reachability
  - Using an analysis of reachability, it is tested, if all node could be reached from each node.
  - Reachability table
  - Reachability graph

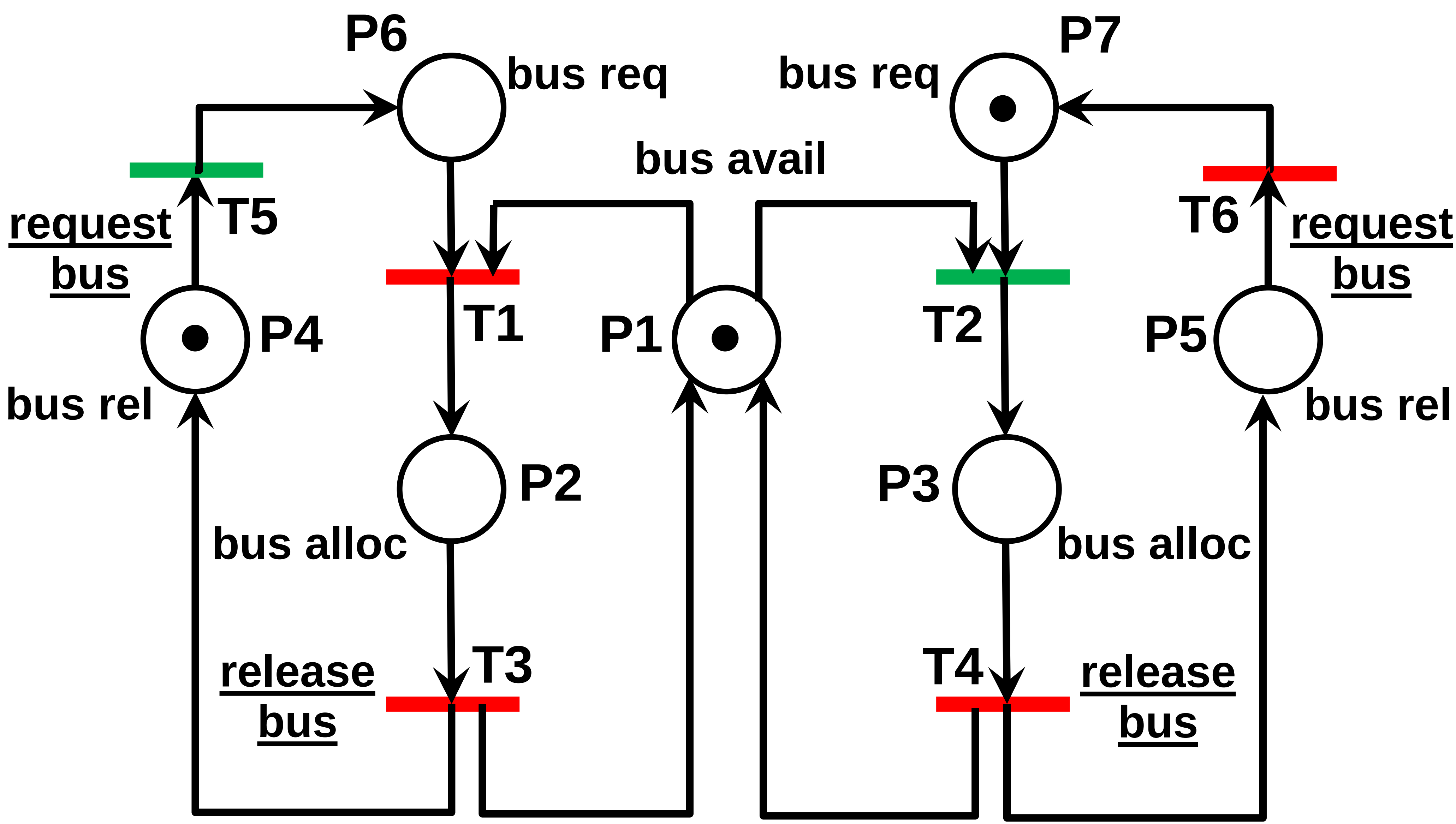
Example: two task using one bus



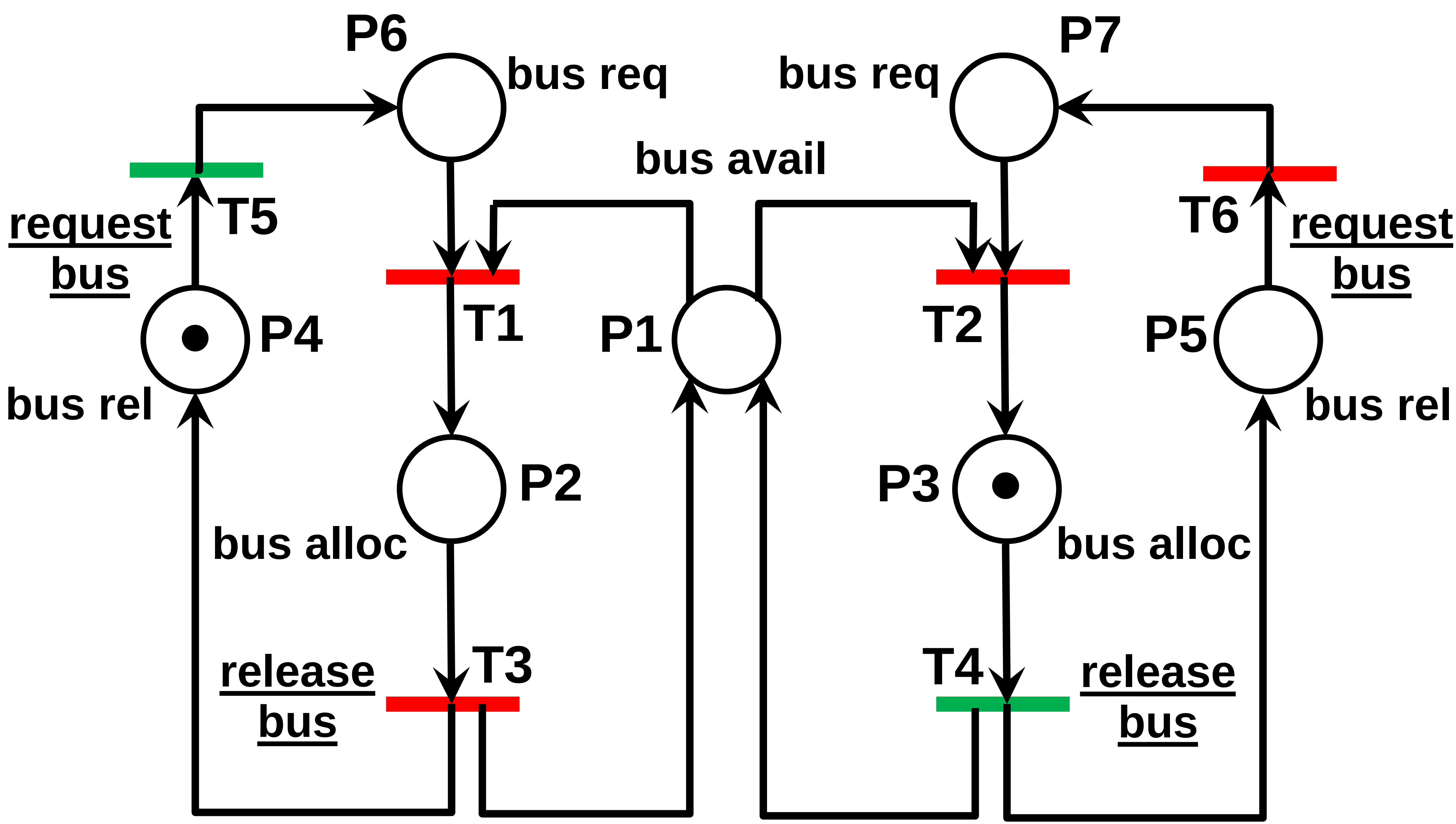
Example: two tasks using one bus



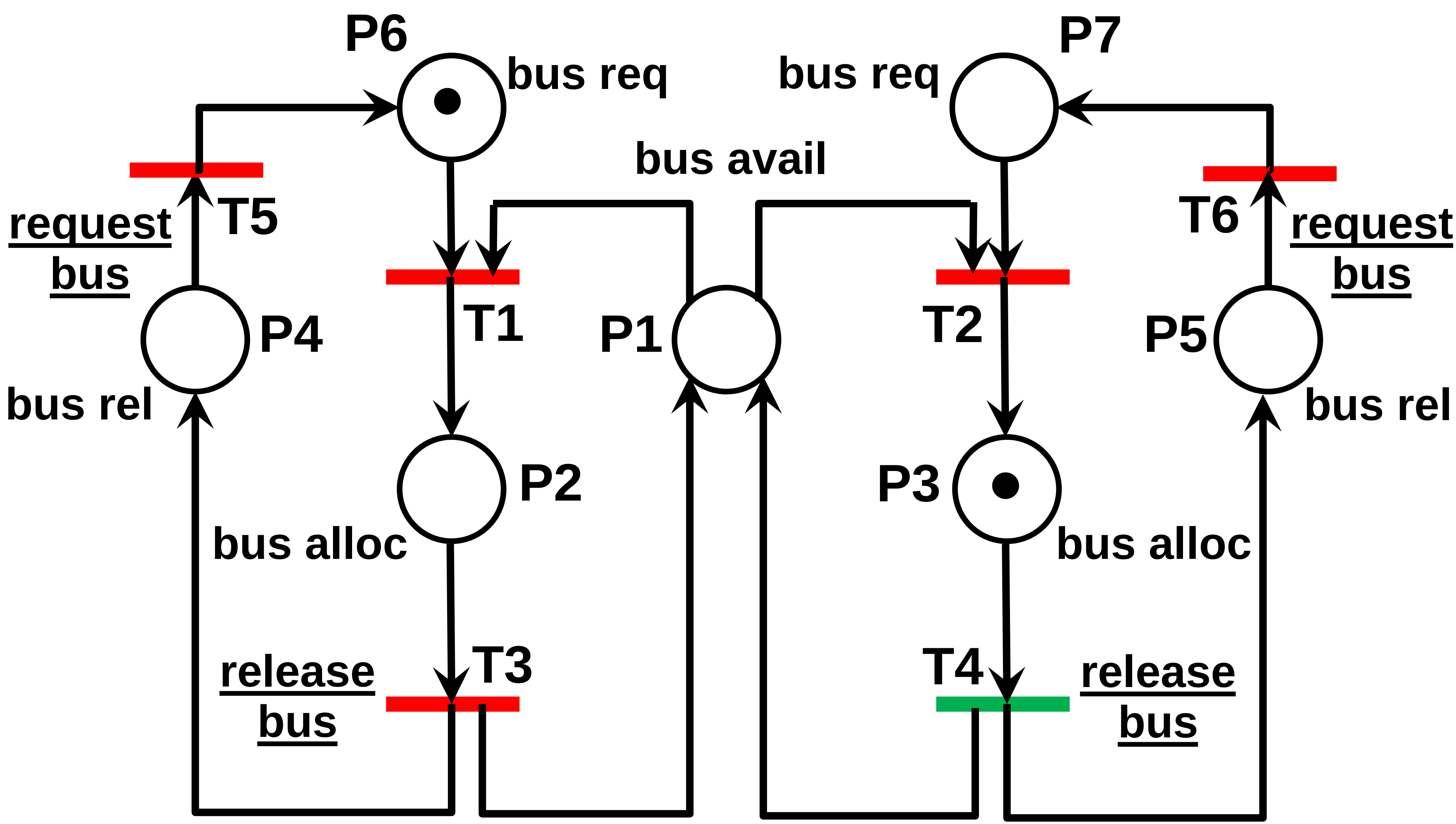
Example: two tasks using one bus



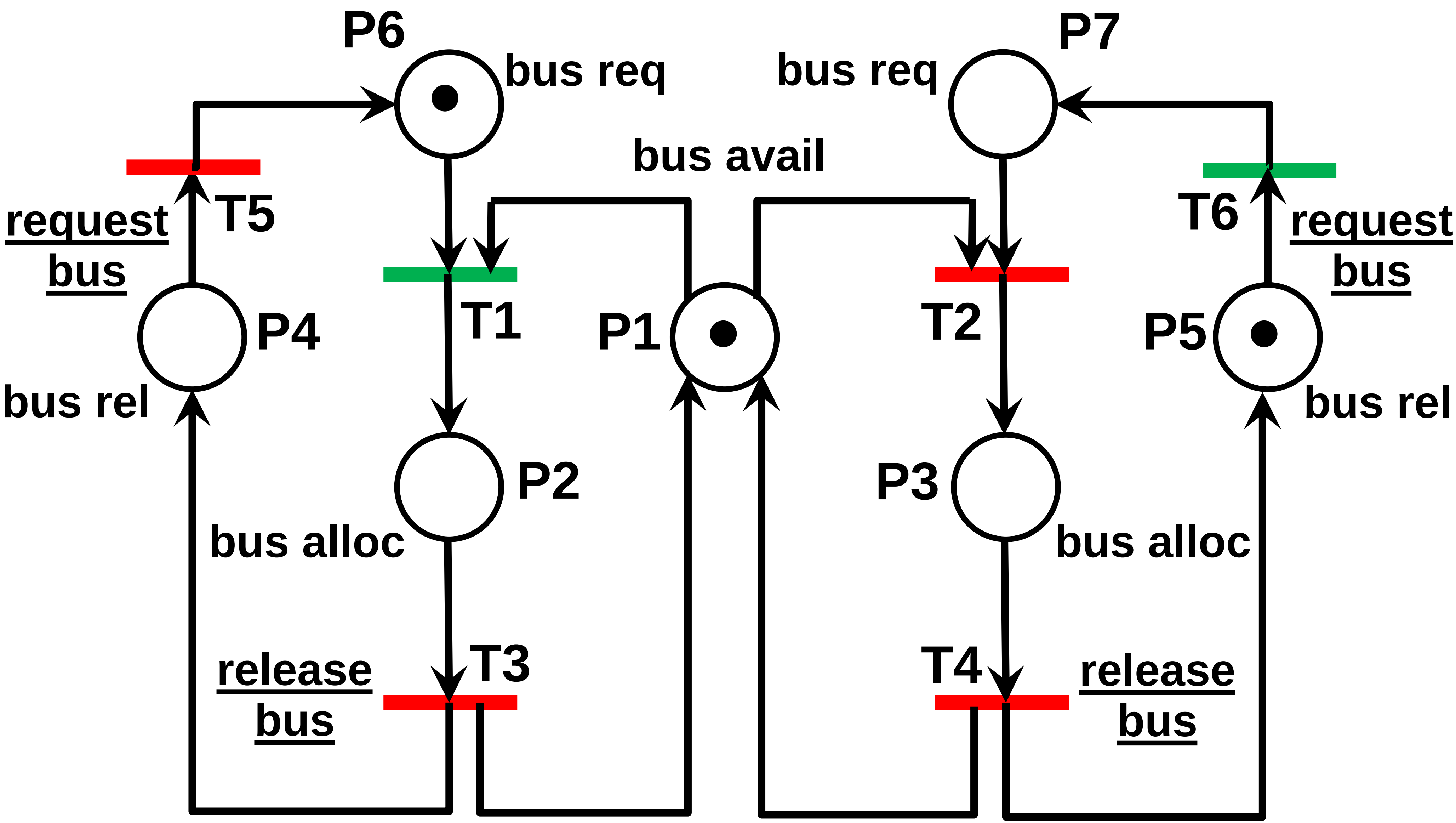
Example: two tasks using one bus



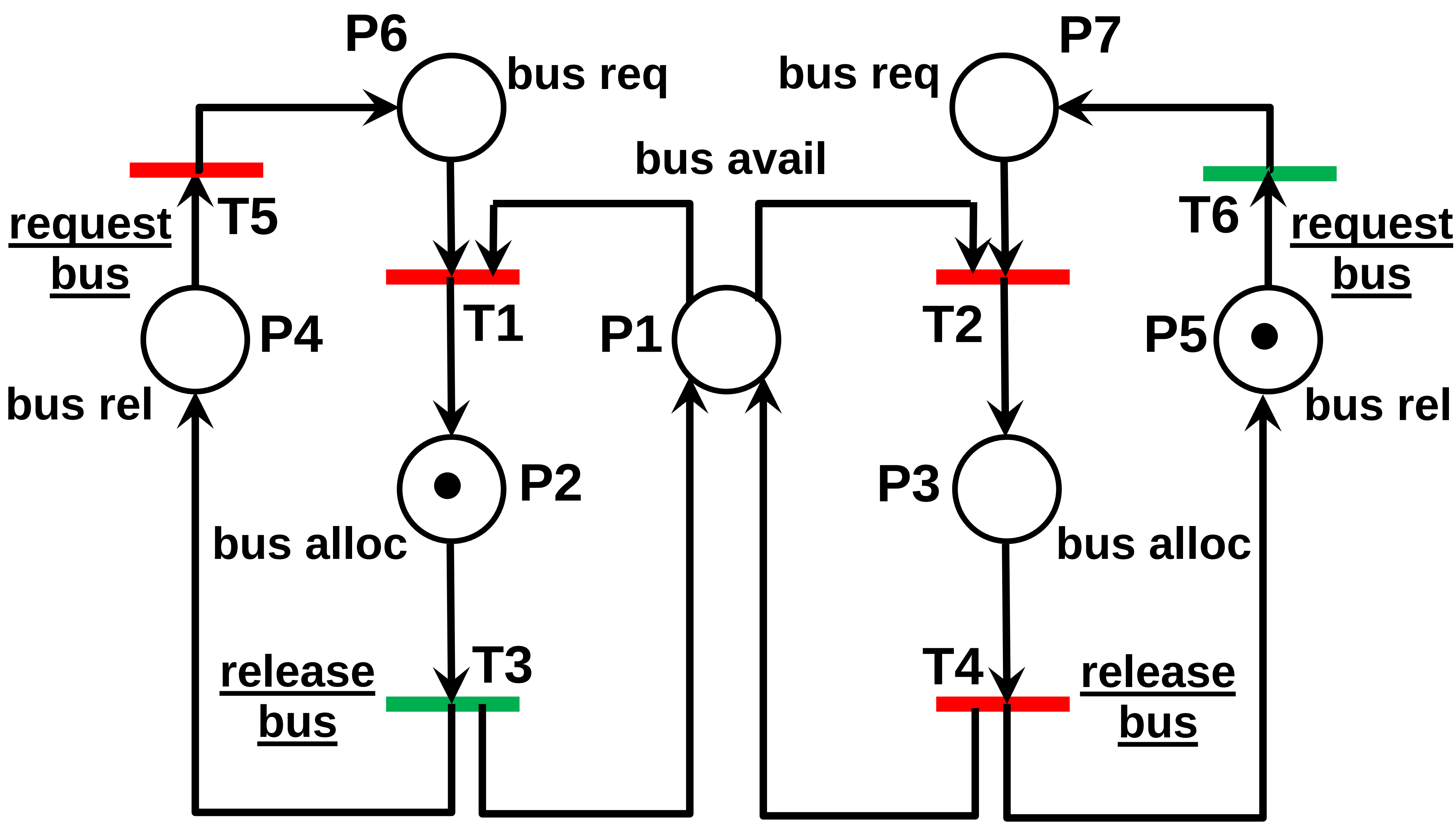
Example: two tasks using one bus



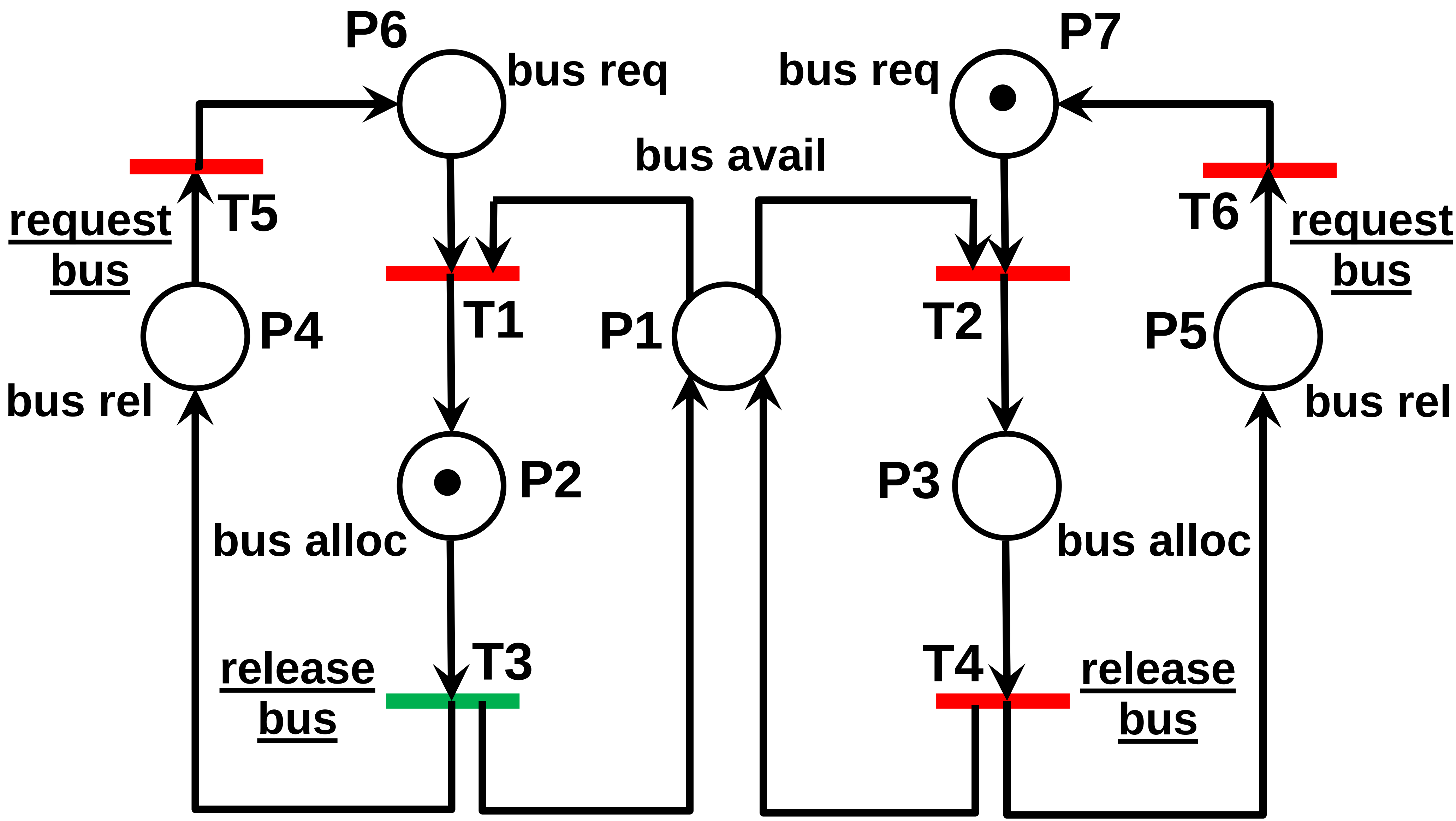
Example: two tasks using one bus



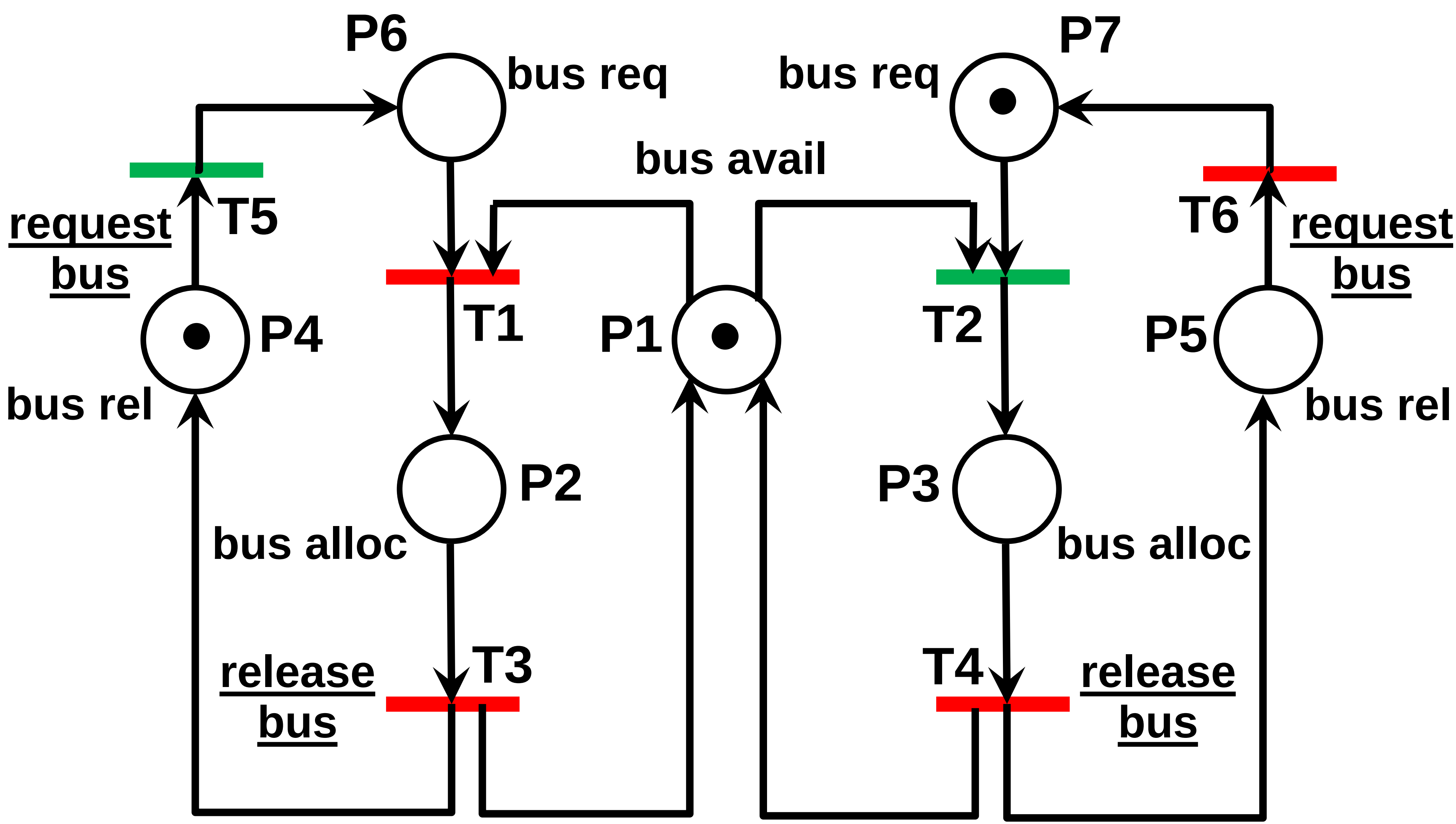
Example: two tasks using one bus



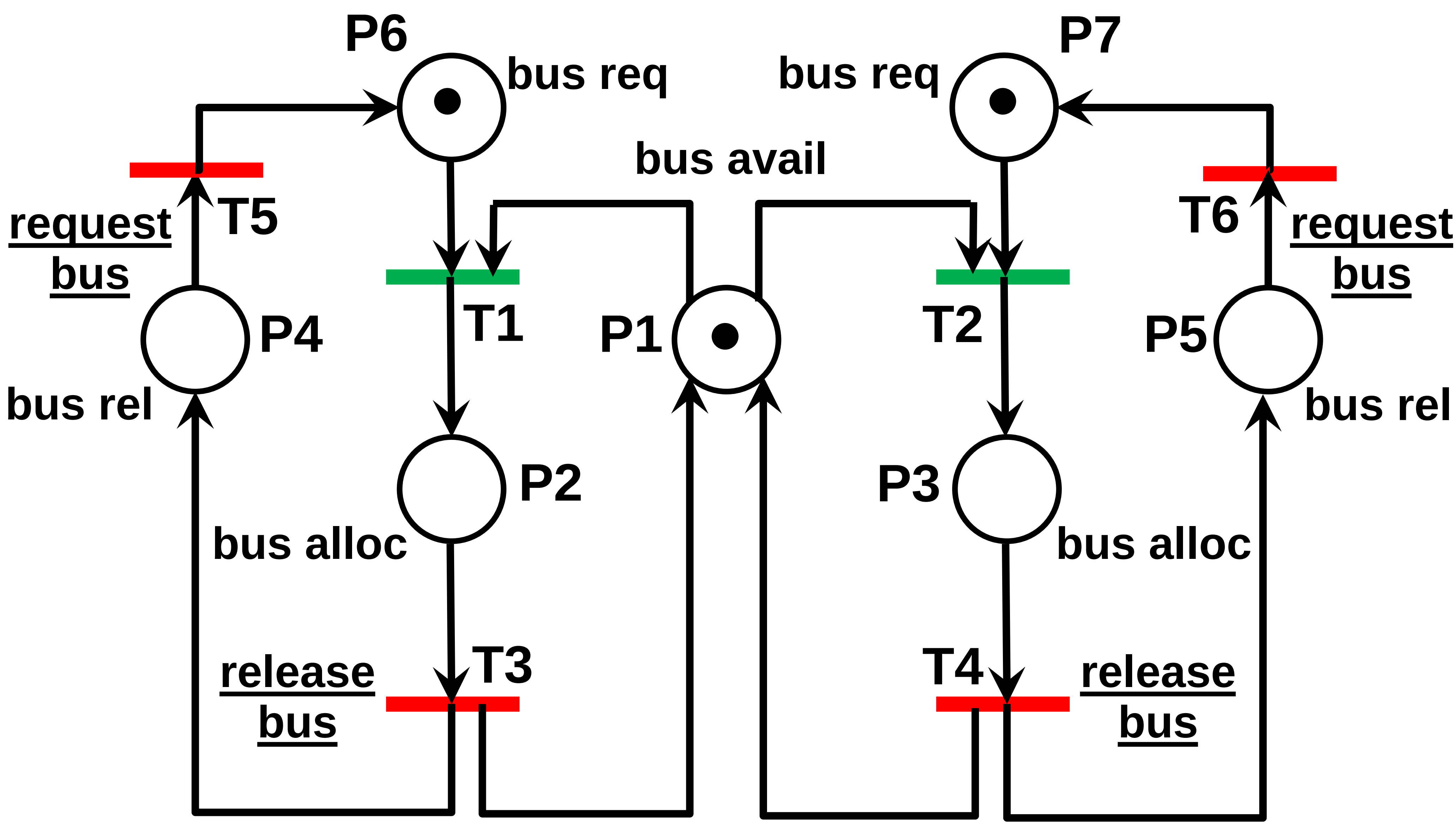
Example: two tasks using one bus



Example: two tasks using one bus



Example: two tasks using one bus



Petri Net Properties

- Properties
  - Behavioural properties
    depend on the initial state of the net
  - Structural properties
    depend not on the initial state of the net but
    on the topology, the structure respectively.

Petri Net: Behavioural Properties

Some of the most important behavioural properties of Petri Nets for modelled system are:

1. Reachability
2. Safeness
3. Liveness

Petri Nets: Structural Properties

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

The most important structural properties of Petri Nets are:

1. Boundedness
2. Persistence

Variants of Petri nets

- Colored
- Timed
- Deterministic/stochastic
- Inhibitor nets

Petri nets

Please be aware that Petri nets a mathematical construct. There are a lot of variation and a lot of use cases. There are complete series of lectures at some universities and there are internationals groups working on this topic only.

However, to model RTS it is essential
to know and understand the fundamentals and
to apply the graphical representation of Petri nets!