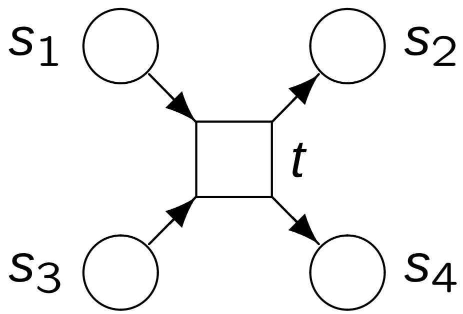
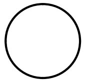
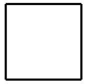
Petri Nets

Petri nets

Petri nets are a basic model of parallel and distributed systems, designed by Carl Adam Petri in 1962 in his PhD Thesis: “Kommunikation mit Automaten”. The basic idea is to describe state changes in a system with transitions.

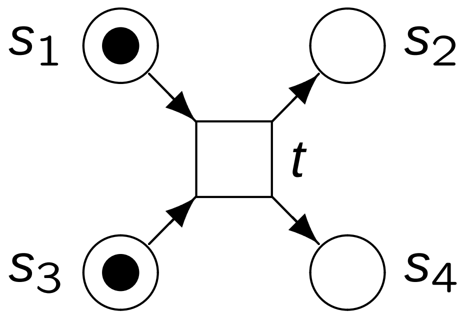


Petri nets contain places  (Stelle) and transitions  (Transition) that may be connected by directed arcs.

Transitions symbolise actions; places symbolise states or conditions that need to be met before an action can be carried out.

Behaviour of Petri nets

Places may contain tokens that may move to other places by executing ("firing") actions.



In the example, transition t may "fire" if there are tokens on places s_1 and s_3. Firing t will remove those tokens and place new tokens on s_2 and s_4.

Place/Transition Nets

Place/Transition Nets

Let us study Petri nets and their firing rule in more detail:

- A place may contain several tokens, which may be interpreted as resources.
- There may be several input and output arcs between a place and a transition. The number of these arcs is represented as the weight of a single arc.
- A transition is enabled if its each input place contains at least as many tokens as the corresponding input arc weight indicates.
- When an enabled transition is fired, its input arc weights are subtracted from the input place markings and its output arc weights are added to the output place markings.

Place/Transition Net

A Place/Transition Net (P/T net) is a tuple $N = \langle P, T, F, W, M_0 \rangle$, where

- P is a finite set of places,
- T is a finite set of transitions,
- the places P and transitions T are disjoint ($P \cap T = \emptyset$),
- $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation,
- $W : F \rightarrow (\mathbb{N} \setminus \{0\})$ is the arc weight mapping, and
- $M_0 : P \rightarrow \mathbb{N}$ is the initial marking representing the initial distribution of tokens.

P/T nets: Remarks

If $\langle p, t \rangle \in F$ for a transition t and a place p, then p is an input place of t,

If $\langle t, p \rangle \in F$ for a transition t and a place p, then p is an output place of t,

Let $a \in P \cup T$. The set ${}^\bullet a = \{a' \mid \langle a', a \rangle \in F\}$ is called the pre-set of a, and the set $a^\bullet = \{a' \mid \langle a, a' \rangle \in F\}$ is its post-set.

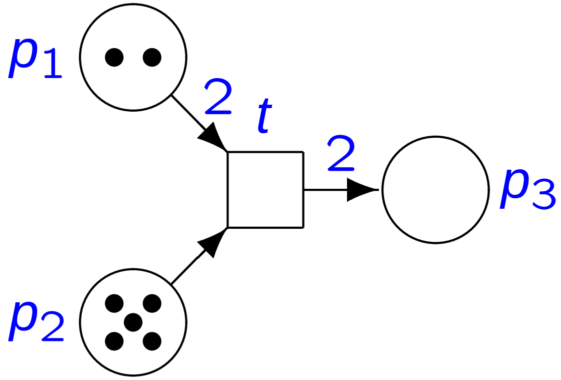
When drawing a Petri net, we usually omit arc weights of 1. Also, we may either denote tokens on a place either by black circles, or by a number.

Alternative definitions

Sometimes the notation *S* (for Stellen) is used instead of *P* (for places) in the definition of Place/Transition nets.

Some definitions also use the notion of a place capacity (the maximum number of tokens allowed in a place, possibly unbounded). Place capacities can be simulated by adding some additional places to the net (we will see how later), and thus for simplicity we will not define them in this course.

Place/Transition Net: Example



The place/transition net $\langle P, T, F, W, M_0 \rangle$ above is defined as follows:

- $P = \{p_1, p_2, p_3\}$,
- $T = \{t\}$,
- $F = \{\langle p_1, t\rangle, \langle p_2, t\rangle, \langle t, p_3\rangle\}$,
- $W = \{\langle p_1, t\rangle \mapsto 2, \langle p_2, t\rangle \mapsto 1, \langle t, p_3\rangle \mapsto 2\}$,
- $M_0 = \{p_1 \mapsto 2, p_2 \mapsto 5, p_3 \mapsto 0\}$.

Notation for markings

Often we will fix an order on the places (e.g., matching the place numbering), and write, e.g., $M_0 = \langle 2, 5, 0 \rangle$ instead.

When no place contains more than one token, markings are in fact sets, in which case we often use set notation and write instead $M_0 = \{p_5, p_7, p_8\}$.

Alternatively, we could denote a marking as a multiset, e.g. $M_0 = \{p_1, p_1, p_2, p_2, p_2, p_2, p_2\}$.

The notation $M(p)$ denotes the number of tokens in place p in marking M.

The firing rule revisited

Let $\langle P, T, F, W, M_0 \rangle$ be a Place/Transition net and $M : P \to \mathbb{N}$ one of its markings.

Firing condition:

Transition $t \in T$ is M-enabled, written $M \xrightarrow{t}$, iff $\forall p \in {}^\bullet t : M(p) \geq W(p, t)$.

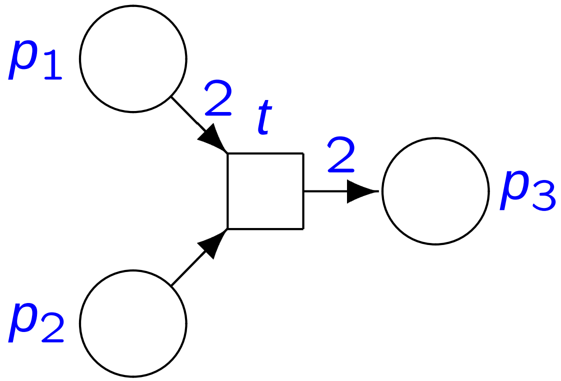
Firing rule:

An M-enabled transition t may fire, producing the successor marking M', written $M \xrightarrow{t} M'$, where

$$\forall p \in P : M'(p) = M(p) - \bar{W}(p, t) + \bar{W}(t, p)$$

where $\bar{W}$ is defined as $\bar{W}(x, y) := W(x, y)$ for $\langle x, y \rangle \in F$ and $\bar{W}(x, y) := 0$ otherwise.

The firing rule of Place/Transition Nets: Example



Marking M	$M \xrightarrow{t}$	M'
$\{p_1 \mapsto 2, p_2 \mapsto 5, p_3 \mapsto 0\}$	enabled	$\{p_1 \mapsto 0, p_2 \mapsto 4, p_3 \mapsto 2\}$
$\{p_1 \mapsto 0, p_2 \mapsto 4, p_3 \mapsto 2\}$	disabled	
$\{p_1 \mapsto 1, p_2 \mapsto 5, p_3 \mapsto 0\}$	disabled	

Note: If $M \xrightarrow{t} M'$, then we call M' the successor marking of M.

Reachable markings

Let M be a marking of a Place/Transition net $N = \langle P, T, F, W, M_0 \rangle$.

The set of markings reachable from M (the reachability set of M, written $reach(M)$) is the smallest set of markings, such that:

1. $M \in reach(M)$, and
2. if $M' \xrightarrow{t} M''$ for some $t \in T$, $M' \in reach(M)$, then $M'' \in reach(M)$.

Let $\mathcal{M}$ be a set of markings. The previous notation is extended to sets of markings in the obvious way:

$$reach(\mathcal{M}) = \bigcup_{M \in \mathcal{M}} reach(M)$$

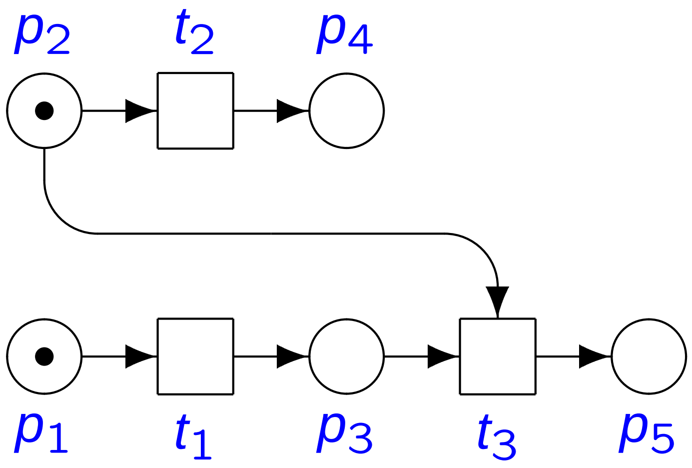
The set of reachable markings $reach(N)$ of a net $N = \langle P, T, F, W, M_0 \rangle$ is defined to be $reach(M_0)$.

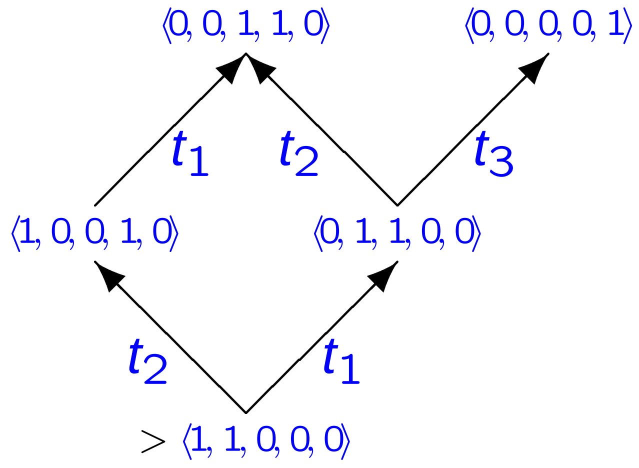
Reachability Graph

The reachability graph of a place/transition net $N = \langle P, T, F, W, M_0 \rangle$ is a rooted, directed graph $G = \langle V, E, v_0 \rangle$, where

- $V = \mathit{reach}(N)$ is the set of vertices, i.e. each reachable marking is a vertex;
- $v_0 = M_0$, i.e. the initial marking is the root node;
- $E = \left\{ \langle M, t, M' \rangle \;\middle|\; M \in V \text{ and } M \xrightarrow{t} M' \right\}$ is the set of edges, i.e. there is an edge from each marking (resp. vertex) M to each of its successor markings, and the edge is labelled with the firing transition.

Reachability Graph: Example





- The weight of each arc is 1.
- The graph shows that t_3 cannot be fired if t_2 is fired before t_1. Thus, intuitively speaking, t_1 and t_2 are not independent, even though their presets and postsets are mutually disjunct.

Computing the reachability graph

REACHABILITY-GRAPH($\langle P, T, F, W, M_0 \rangle$)

$\langle V, E, v_0 \rangle := \langle \{M_0\}, \emptyset, M_0 \rangle$;
$Work$: set $:= \{M_0\}$;
while $Work \neq \emptyset$
do select M from $Work$;
  $Work := Work \setminus \{M\}$;
  for $t \in \mathsf{enabled}(M)$
  do $M' := \mathsf{fire}(M, t)$;
    if $M' \notin V$
      then $V := V \cup \{M'\}$
        $Work := Work \cup \{M'\}$;
    $E := E \cup \{\langle M, t, M' \rangle\}$;
return $\langle V, E, v_0 \rangle$;

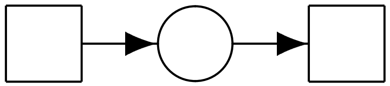
The algorithm makes use of two functions:

- $\mathsf{enabled}(M) := \{t \mid M \xrightarrow{t}\}$
- $\mathsf{fire}(M, t) := M'$ if $M \xrightarrow{t} M'$

The set $Work$ may be implemented as a stack, in which case the graph will be constructed in a depth-first manner, or as a queue for breadth-first. Breadth first search will find the shortest transition path from the initial marking to a given (erroneous) marking. Some applications require depth first search.

The size of the reachability graph

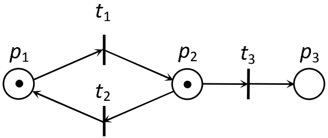
In general, the graph may be infinite, i.e. if there is no bound on the number tokens on some place. Example:



Definition: If each place of a place/transition net can contain at most k tokens in each reachable marking, the net is said to be k-safe.

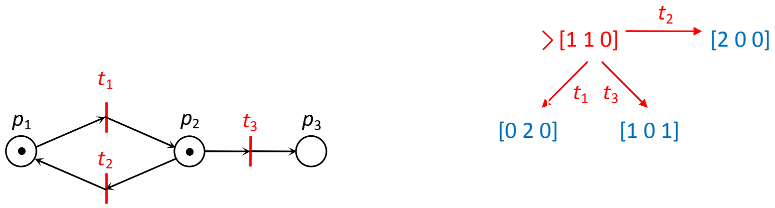
A k-safe net has at most $(k+1)^{|P|}$ markings; for 1-safe nets, the limit is $2^{|P|}$.

Example

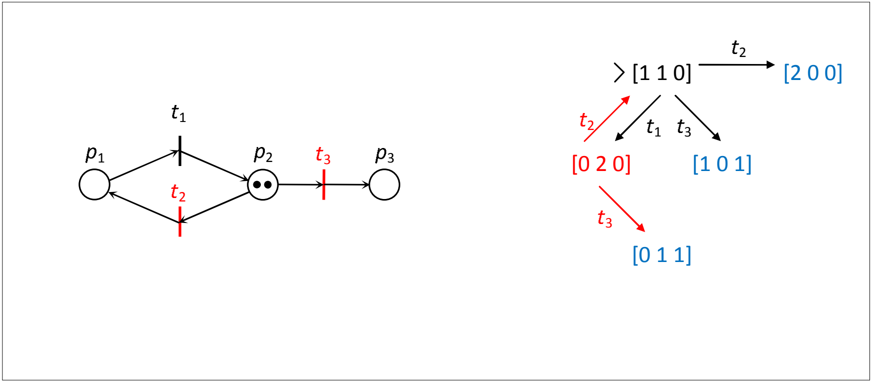


> [1 1 0]

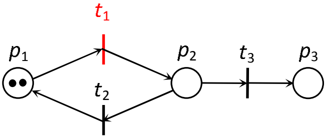
Example

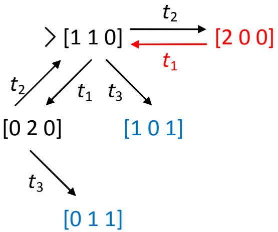


Example

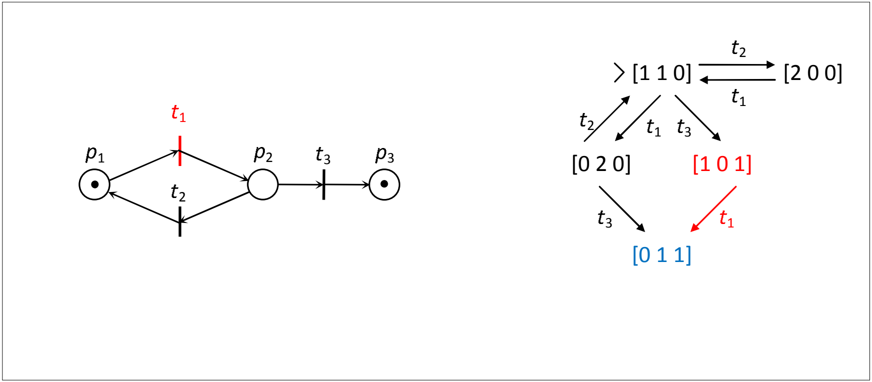


Example

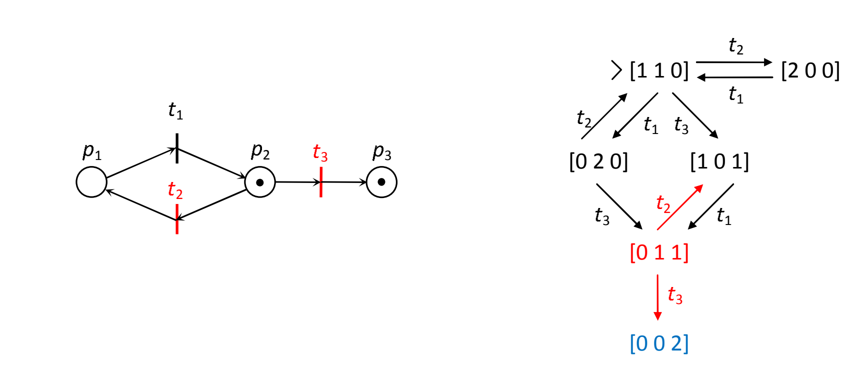




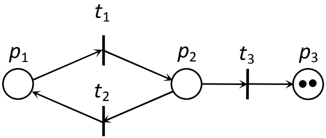
Example

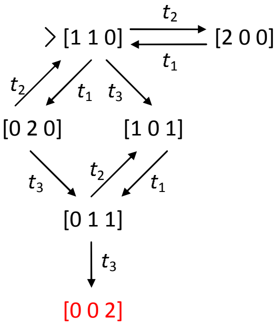


Example



Example





Motivation

A marked net $\langle N, M_0 \rangle$ with $N = (P, T, Pre, Post)$ specifies:

- an initial marking (i.e., state) M_0;
- the rules of evolution.

No explicit enumeration of:

- net language, i.e., the set of sequences of transitions that can fire:

$$L(N, M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\};$$

- reachability set, i.e., the set of reachable markings:

$$R(N, M_0) = \{M \in \mathbb{N}^{|P|} \mid (\exists \sigma \in L(N, M_0))\ M_0[\sigma\rangle M\}.$$

The information on reachable markings and firing sequences is useful to determine if the net has given properties.

Reachability graph

The reachability graph of a marked net $\langle N, M_0 \rangle$ is an automaton $\mathcal{G} = (X, E, \delta, x_0)$ where:

- $X = R(N, M_0)$, i.e., the states of the automaton are the reachable markings;
- $E = T$, i.e., the events in the alphabet are the transitions of the net;
- for any two reachable markings M, M':

  $$\delta(M, t) = M' \quad \Longleftrightarrow \quad M[t\rangle M',$$

  i.e., there exists arc labeled t from M to M' on the automaton iff marking M' is reachable from M firing transition t;
- $x_0 = M_0$, i.e., the initial state of the automaton is the initial marking.

It can be constructed only if the reachability set if finite, i.e., if the net is bounded.

Boundedness

Definition

A place $p \in P$ is k-bounded if for any marking $M \in R(N, M_0)$ it holds $M(p) \leq k$, i.e., in all reachable markings the number of tokens it contains never exceeds k.

Useful to determine maximal capacity or overflow of buffers.

Definition

A marked net $\langle N, M_0 \rangle$ is k-bounded if all its places are k-bounded.

A bounded net has a finite reachability set, while an unbounded net has an infinite reachability set.

Liveness

Definition

A transition $t \in T$ is quasi-live if there exists a firing sequence $\sigma \in T^*$ such that $M_0[\sigma t\rangle$, i.e., transition t can *eventually* fire.

A transition $t \in T$ is live if for any marking $M \in R(N, M_0)$ there exists a firing sequence $\sigma \in T^*$ such that $M[\sigma t\rangle$, i.e., from from any reachable marking t can *eventually* fire.

Useful to characterize an event that can occur at least once (quasi-liveness) or that can always eventually occur (liveness).

Definition

A marked net $\langle N, M_0\rangle$ is quasi-live (resp. live) if all its transitions are quasi-live (resp., live).

Reversibility

Definition

A marked net $\langle N, M_0 \rangle$ is reversible if for any marking $M \in R(N, M_0)$ it holds $M_0 \in R(N, M)$, i.e., from any reachable marking M it is possible to reach the initial marking M_0.

Useful to determine if a system can always be reinitialized.

What does the reachability graph tell us?

Proposition

Given a marked net $\langle N, M_0 \rangle$ let $\mathcal{G}$ be its reachability graph with set of states X constructed using the previous algorithm.

- $R(N, M_0) = X$
- $L(N, M_0) = L(\mathcal{G})$

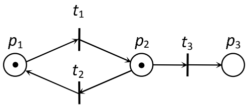
Two main informations from the reachability graph $\mathcal{G}$.

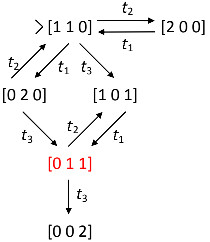
- Marking M is reachable $\iff$ M is a node of $\mathcal{G}$.
- $\sigma \in L(N, M_0) \iff \delta(M_0, \sigma)$ is defined in $\mathcal{G}$

A stronger property also holds

- $M[\sigma\rangle M' \iff$ there exists a path from M to M' labeled by σ

What does the reachability graph tell us?

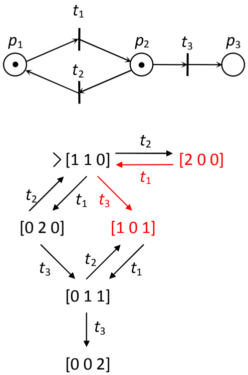




Example:

- $M = [0\ 1\ 1]^T$ is reachable

What does the reachability graph tell us?



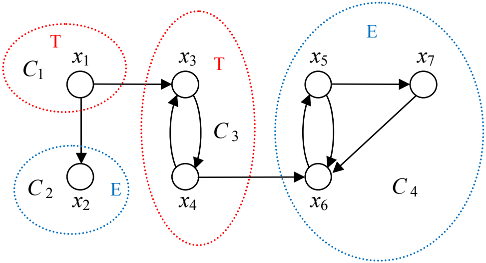
Example:

- $M = [0\ 1\ 1]^T$ is reachable
- $[2\ 0\ 0]^T\ [t_1 t_3\rangle\ [1\ 0\ 1]^T$

Partition of an automaton in components

The states of an automaton can be partitioned into **strongly connect components** (i.e, maximal set of states mutually reachable).

1. Transient components: there are paths going out of the component.
2. Ergodic (or absorbing) components: there no are paths going out of the component.

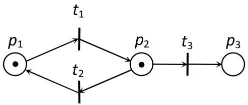


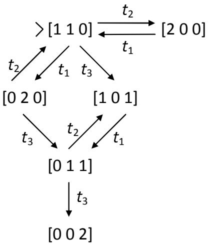
This will be useful to check for reversibility and liveness.

Boundeness

- The reachability graph of marked net $\langle N, M_0 \rangle$ can only be constructed if the net is bounded.
- The bound k_p of place p is $\max M(p)$ for all nodes in $\mathcal{G}$.
- The bound k on the net is $\max k_p$ for all places.

Example



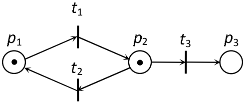


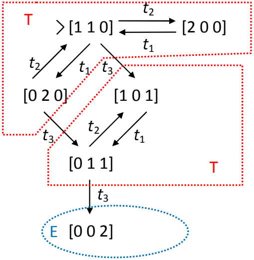
- The bound of all places is $k_p = 2$
- The net is 2-bounded.

Liveness

- A transition t is quasi-live $\iff$ an arc t appears in the graph.
- A transition t is live $\iff$ an arc t appears in all ergodic components.

Example 1



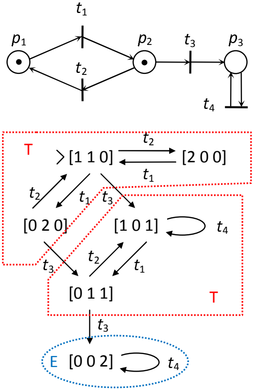


- All transitions are quasi-live
- No transition is live: once we reach the ergodic component no transition can fire.

Liveness

- A transition t is quasi-live $\iff$ an arc t appears in the graph.
- A transition t is live $\iff$ an arc t appears in all ergodic components.

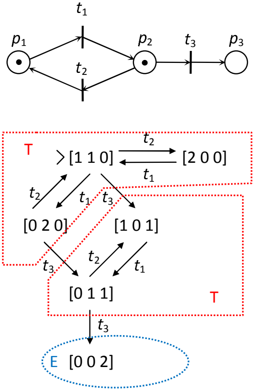
Example 2



- All transitions are quasi-live
- Transition t_4 is the only one live.

Reversibility

- Marked net $\langle N, M_0 \rangle$ is reversible $\iff$ the graph is strongly connected, i.e., it consists of a single connected component.



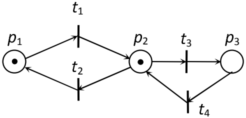
Example 1

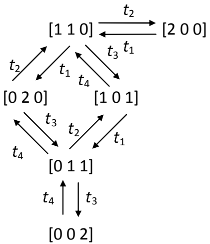
- The graph is not strongly connected: the net is not reversible.
- E.g., from $M = [0\ 1\ 1]^T$ the initial marking $M_0 = [1\ 1\ 0]^T$ is not reachable.

Reversibility

- Marked net $\langle N, M_0 \rangle$ is reversible $\iff$ the graph is strongly connected, i.e., it consists of a single connected component.

Example 2



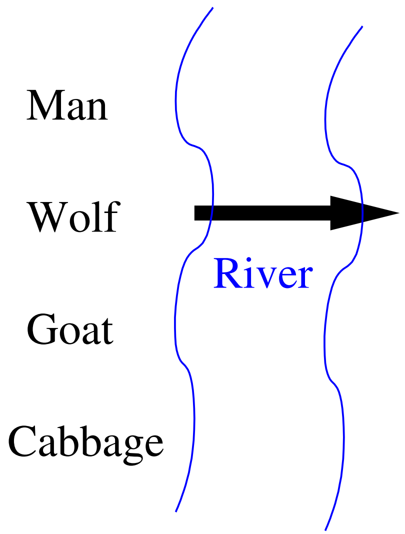


- The graph is strongly connected: the net is reversible.

Example: A logical puzzle

A man is travelling with a wolf, a goat, and a cabbage. The four come to a river that they must cross. There is a boat available for crossing the river, but it can carry only the man and at most one other object. The wolf may eat the goat when the man is not around, and the goat may eat the cabbage when unattended.

Can the man bring everyone across the river without endangering the goat or the cabbage? And if so, how?



Example: Modeling

We are going to model the situation with a Petri net.

The puzzle mentions the following objects:

> Man, wolf, goat, cabbage, boat. Both can be on either side of the river.

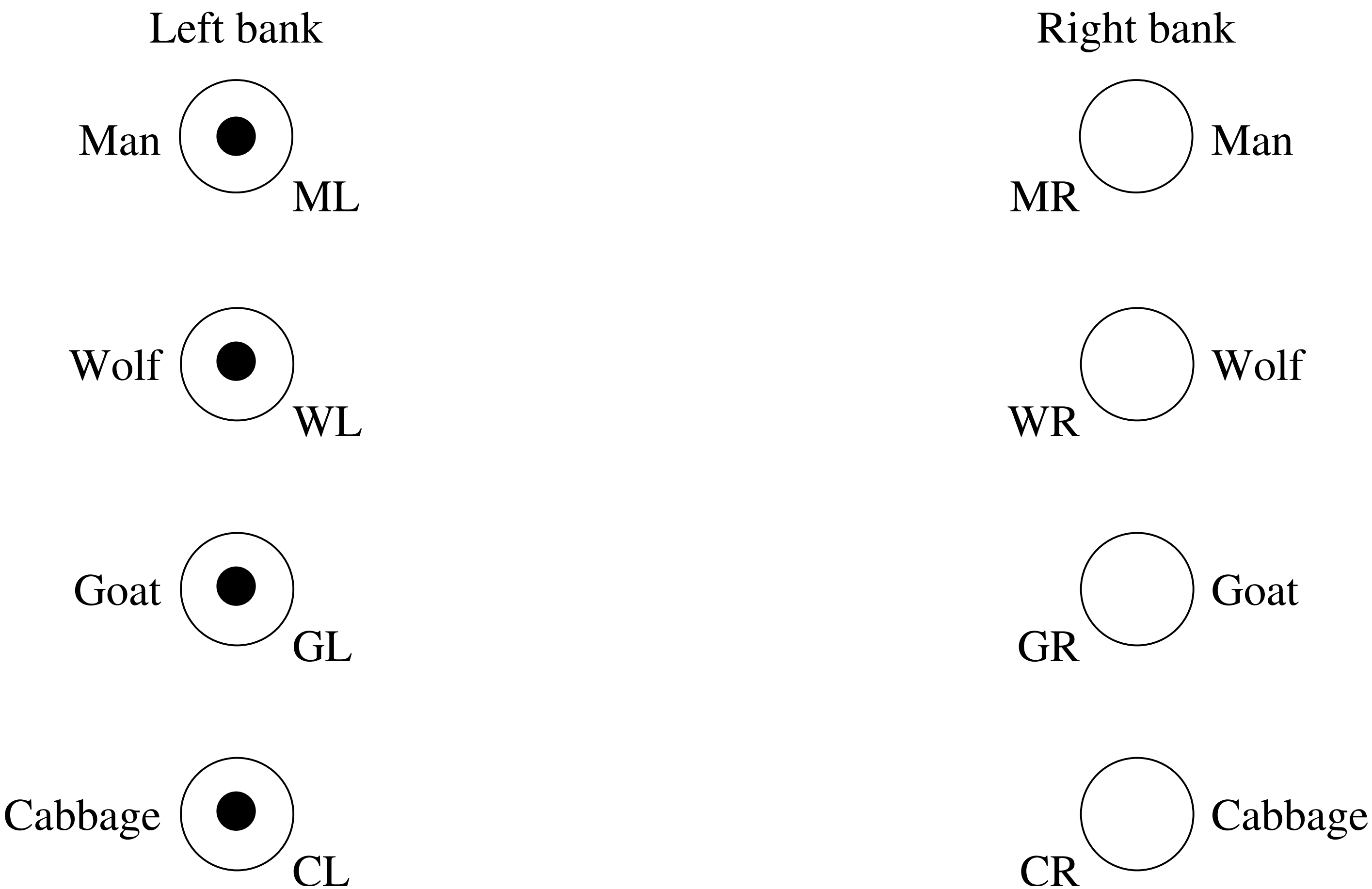
The puzzle mentions the following actions:

> Crossing the river, wolf eats goat, goat eats cabbage.

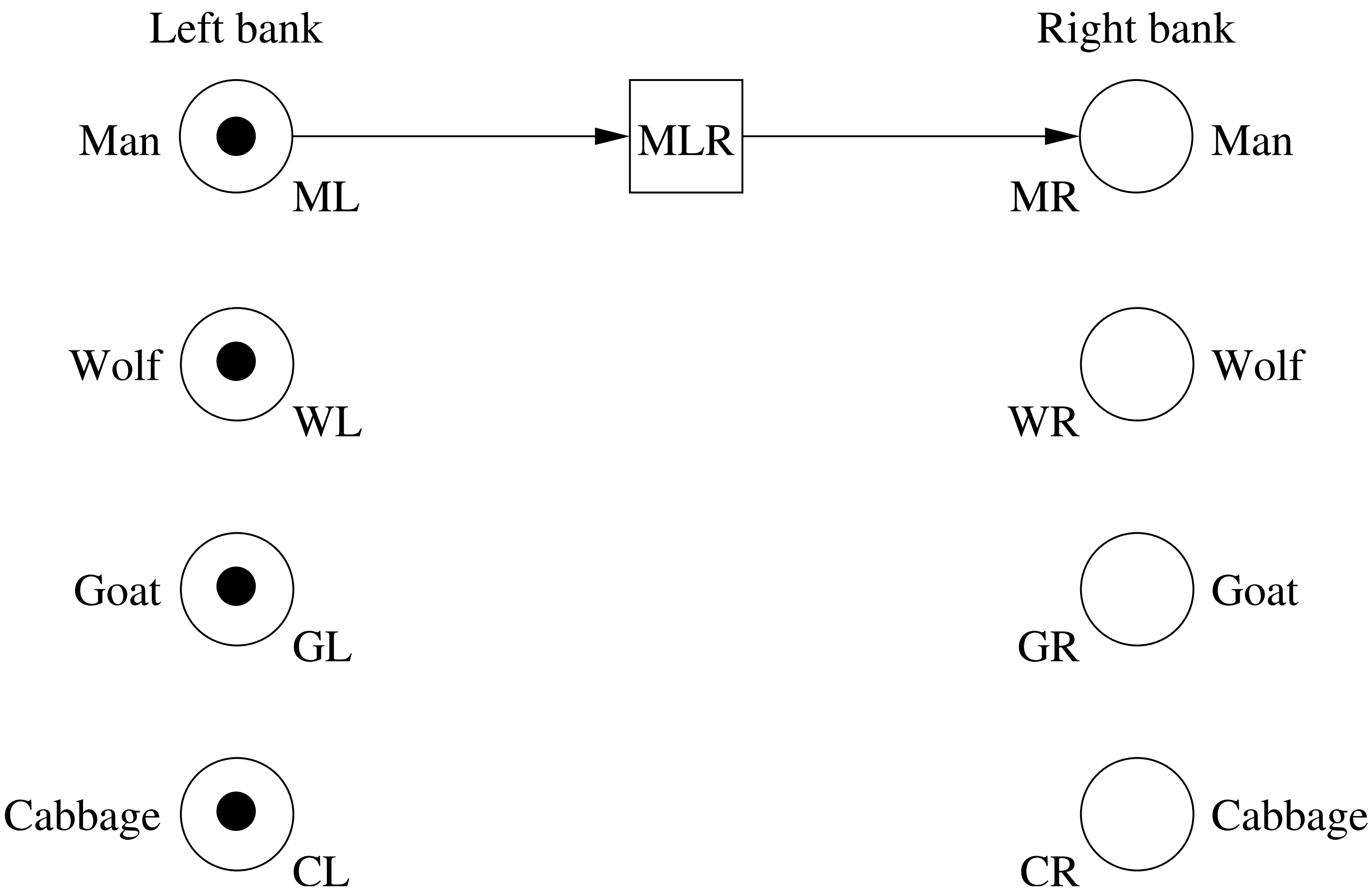
Objects and their states are modeled by places.
Actions are modeled by transitions.

Actually, we can omit the boat, because it is always going to be on the same side as the man.

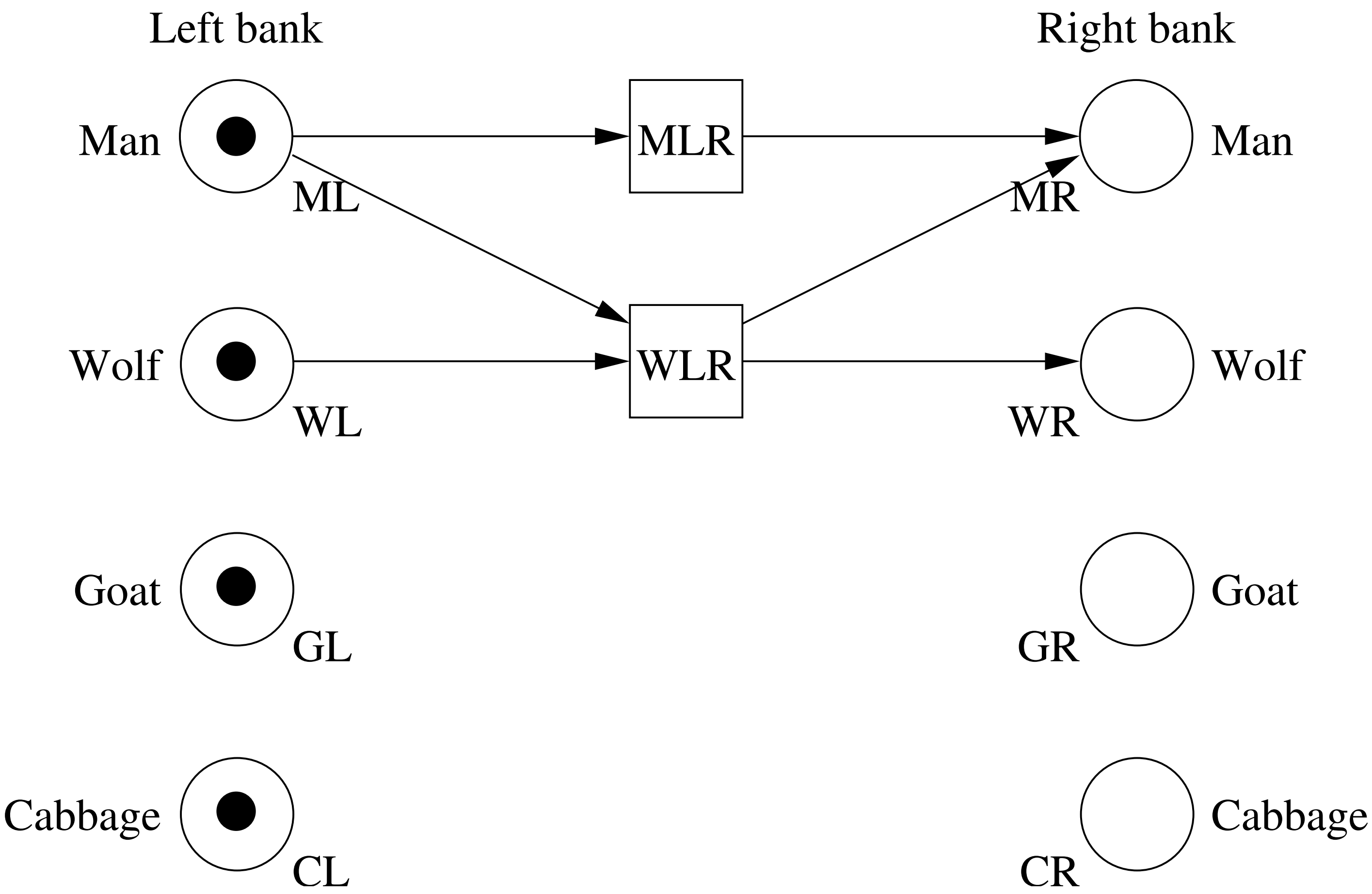
Example: Places



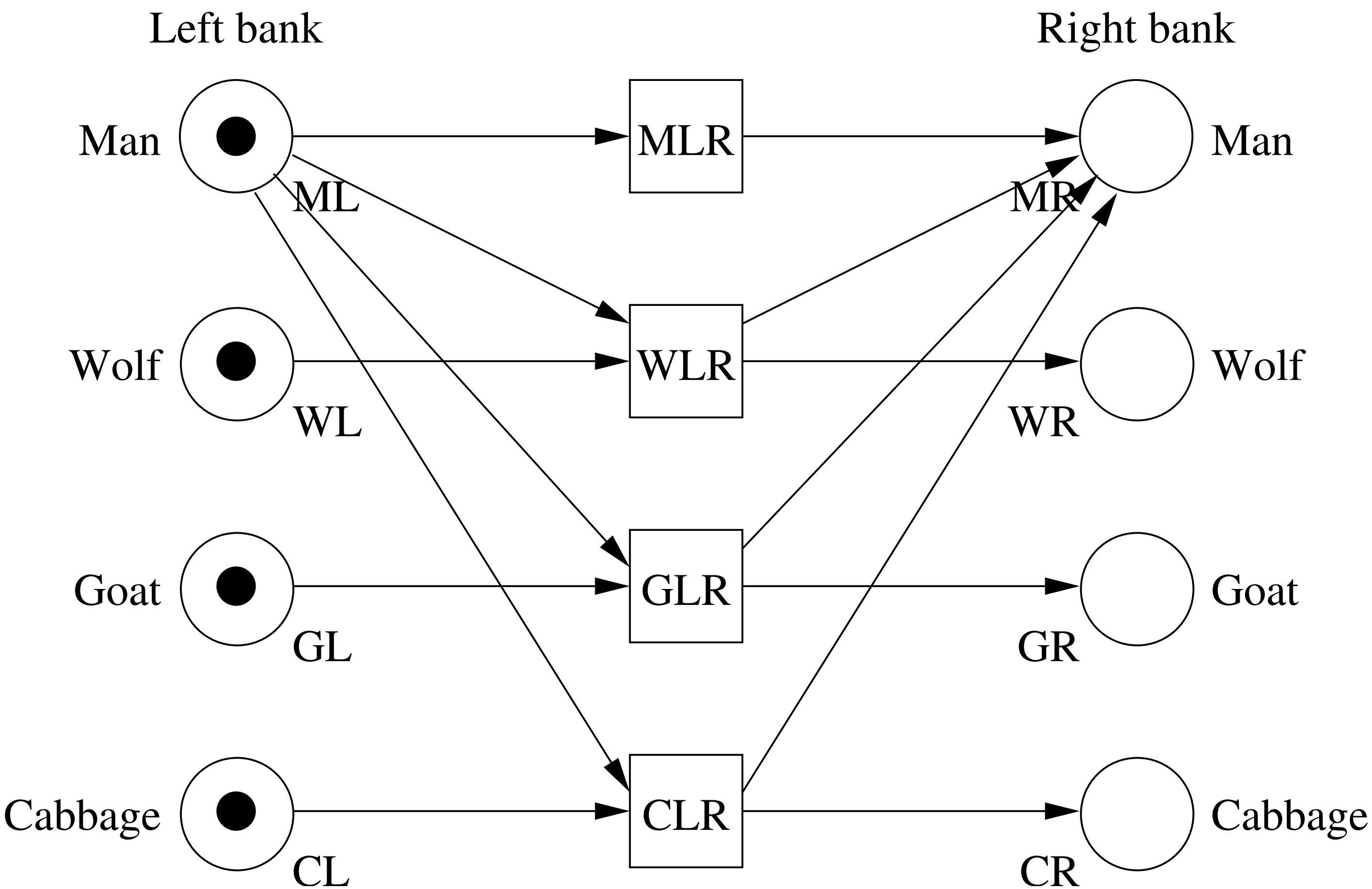
Crossing the river (left to right)



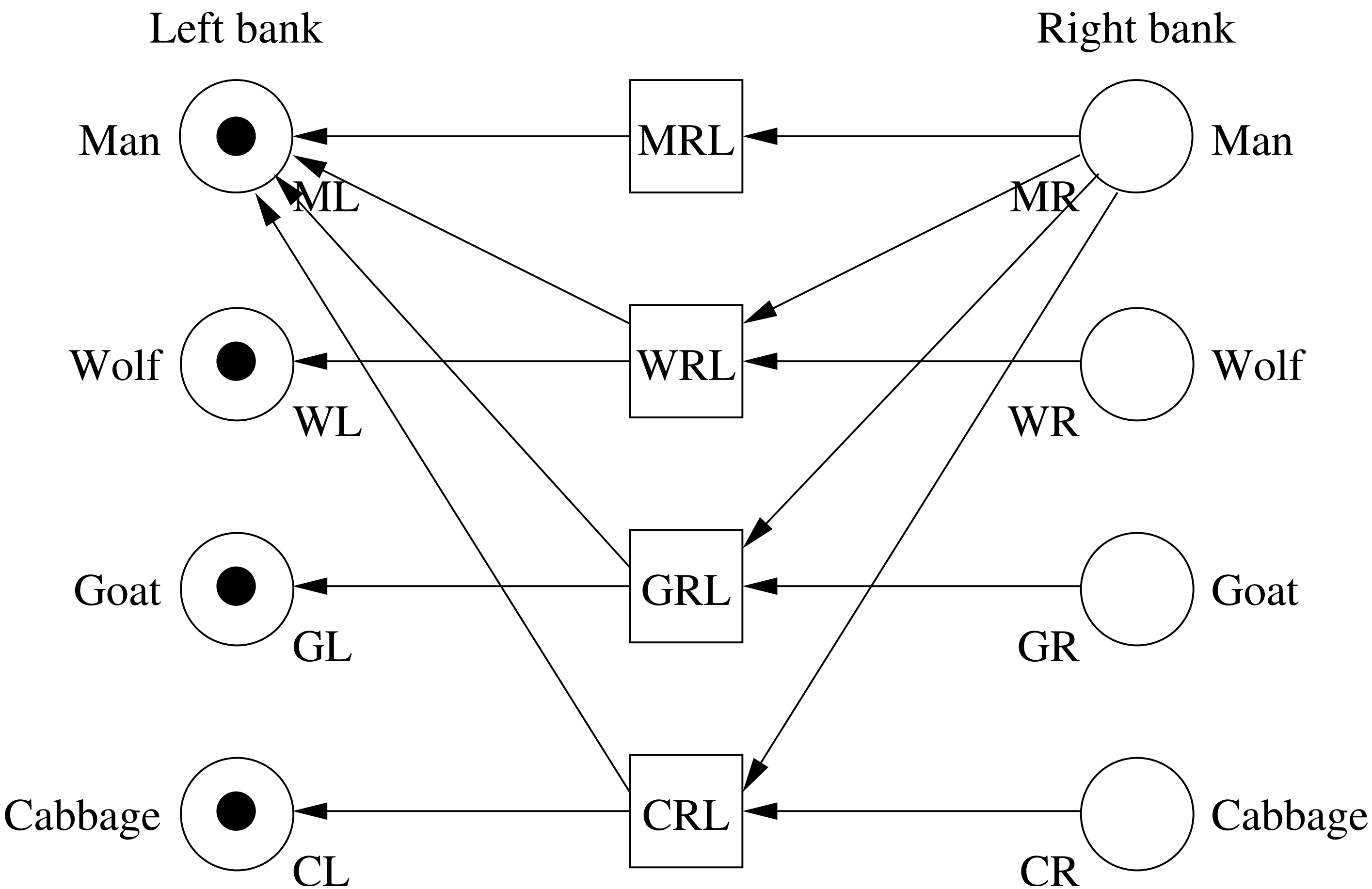
Crossing the river (left to right)



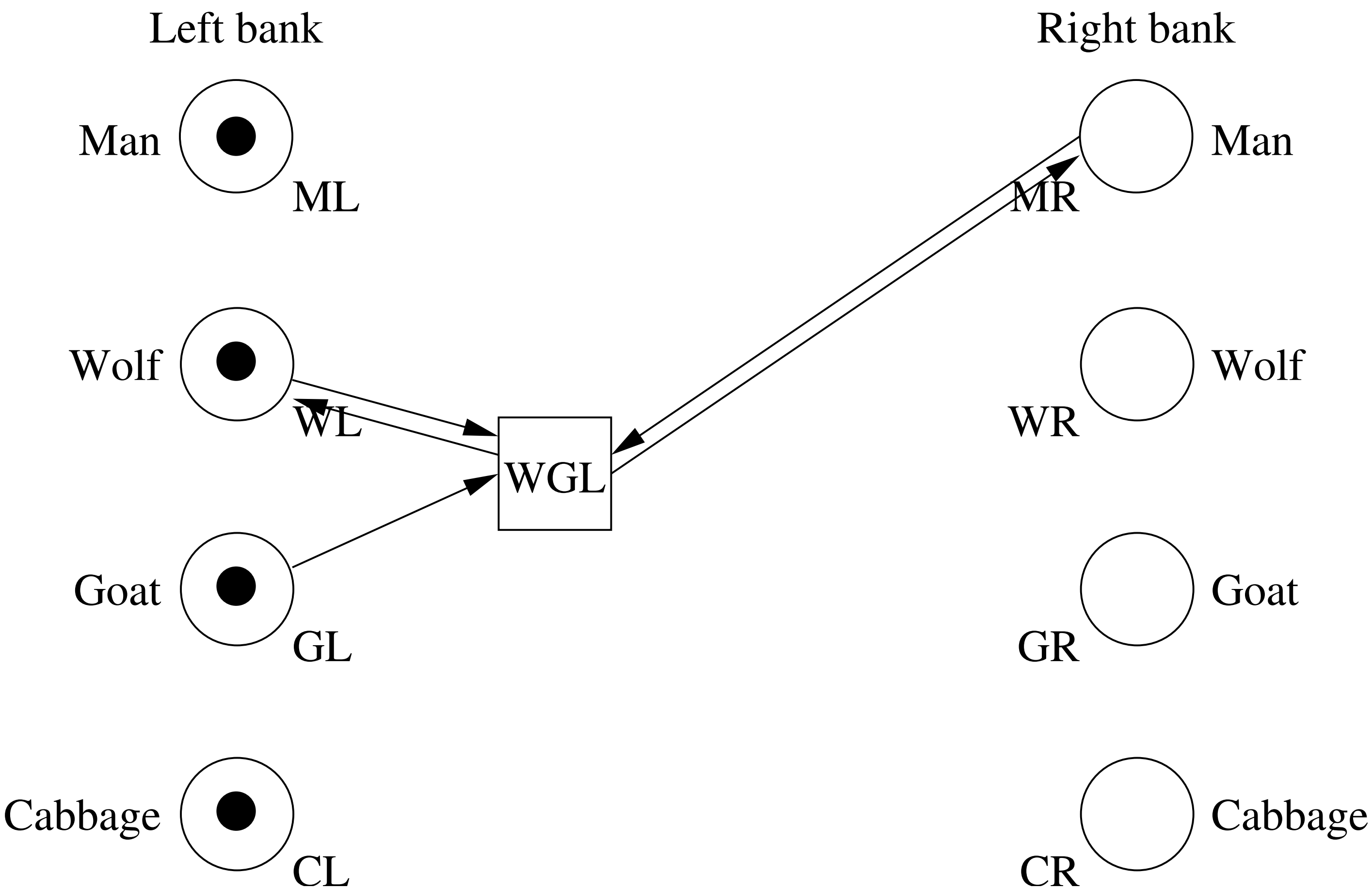
Crossing the river (left to right)



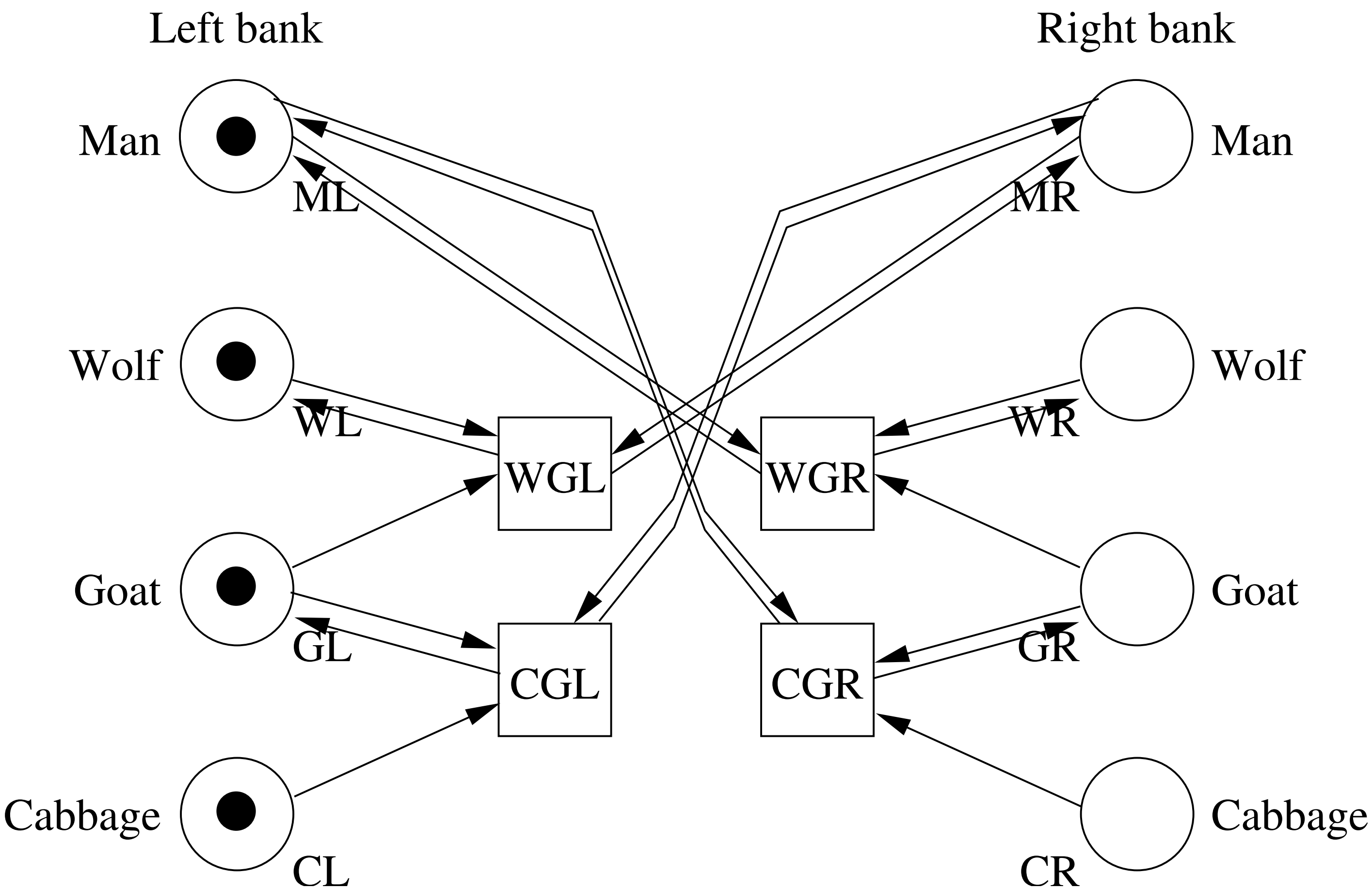
Crossing the river (right to left)



Wolf eats goat



Wolf eats goat, goat eats cabbage



Example: Specification

To solve the problem using the Petri net, we need to translate the questions *"Can the man bring everyone across the river without endangering the goat or the cabbage? And if so, how?"* into properties of the Petri net.

Example: Specification

To solve the problem using the Petri net, we need to translate the questions *"Can the man bring everyone across the river without endangering the goat or the cabbage? And if so, how?"* into properties of the Petri net.

"Can the man bring everyone across the river?"

$\Rightarrow$ Is the marking $\{MR, WR, GR, CR\}$ reachable from $\{ML, WL, GL, CL\}$?

Example: Specification

To solve the problem using the Petri net, we need to translate the questions *“Can the man bring everyone across the river without endangering the goat or the cabbage? And if so, how?”* into properties of the Petri net.

“Can the man bring everyone across the river?”

$\Rightarrow$ Is the marking $\{MR, WR, GR, CR\}$ reachable from $\{ML, WL, GL, CL\}$?

“$\ldots$ without endangering the goat or the cabbage?”

$\Rightarrow$ We need to avoid states in which one of the eating transitions is enabled.

Example: Specification

To solve the problem using the Petri net, we need to translate the questions *“Can the man bring everyone across the river without endangering the goat or the cabbage? And if so, how?”* into properties of the Petri net.

“Can the man bring everyone across the river?”

$\Rightarrow$ Is the marking $\{MR, WR, GR, CR\}$ reachable from $\{ML, WL, GL, CL\}$?

“$\ldots$ without endangering the goat or the cabbage?”

$\Rightarrow$ We need to avoid states in which one of the eating transitions is enabled.

“How?”

$\Rightarrow$ Give a path that leads from one marking to the other. (Optionally: Find a shortest path.)

Result

Constructing the reachability graph yields a graph with (at most) 36 nodes.

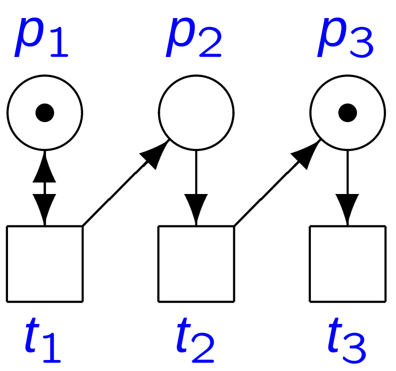
The marking $\{MR, WR, GR, CR\}$ *is reachable* without enabling an "eating" transition!

The transitions fired along a shortest path (there are two) are:

- GLR (man and goat cross the river),
- MRL (man goes back alone),
- WLR (man and wolf cross the river),
- GRL (man and goat go back),
- CLR (man and cabbage cross the river),
- MRL (man goes back alone),
- GLR (man and goat cross the river).

Coverability Graph Method

As we have mentioned before, the reachability graph of P/T-net can be infinite (in which case the algorithm for computing the reachability graph will not terminate). For example, consider the following net.



We will show a method to find out whether the reachability graph of a P/T-net is infinite or not. This can be done by using the coverability graph method.

ω-Markings

First we introduce a new symbol ω to represent "arbitrarily many" tokens.

We extend the arithmetic on natural numbers with ω as follows. For all $n \in \mathbb{N}$:

$n + \omega = \omega + n = \omega$,

$\omega + \omega = \omega$,

$\omega - n = \omega$,

$0 \cdot \omega = 0$, $\omega \cdot \omega = \omega$,

$n \geq 1 \Rightarrow n \cdot \omega = \omega \cdot n = \omega$,

$n \leq \omega$, and $\omega \leq \omega$.

Note: $\omega - \omega$ remains undefined, but we will not need it.

We will extend the notion of markings to ω-markings. In an ω-marking, each place p will either have $n \in \mathbb{N}$ tokens, or ω tokens (infinitely many).

Firing Rule and ω-markings

The firing condition and firing rule (reproduced below) neatly extend to ω-markings with the extended arithmetic rules:

Firing condition:

Transition $t \in T$ is M-enabled, written $M \xrightarrow{t}$, iff $\forall p \in {}^{\bullet}t : M(p) \geq W(p,t)$.

Firing rule:

An M-enabled transition t may fire, producing the successor marking M', where

$$\forall p \in P : M'(p) = M(p) - \bar{W}(p,t) + \bar{W}(t,p).$$

Basically, if a transition has a place with ω tokens in its preset, that place is considered to have sufficiently many tokens for the transition to fire, regardless of the arc weight.

If a place contains an ω-marking, then firing any transition connected with an arc to that place will not change its marking.

Definition of Covering

An ω-marking M' covers an ω-marking M, denoted $M \leq M'$, iff

$$\forall p \in P: M(p) \leq M'(p).$$

An ω-marking M' strictly covers an ω-marking M, denoted $M < M'$, iff

$$M \leq M' \quad \text{and} \quad M' \neq M.$$

Coverability and Transition Sequences (1/2)

Observation: Let M and M' be two markings such that $M \leq M'$.
Then for all transitions t, the following holds:

$$\text{If } M \xrightarrow{t} \text{ then } M' \xrightarrow{t}.$$

In other words, if M' has at least as many tokens as M has (on each place), then M' enables at least the same transitions as M does.

This observation can be extended to *sequences* of transitions:
Define $M \xrightarrow{t_1 t_2 \ldots t_n} M'$ to denote:

$$\exists M_1, M_2, \ldots, M_n : M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \cdots \xrightarrow{t_n} M_n = M'.$$

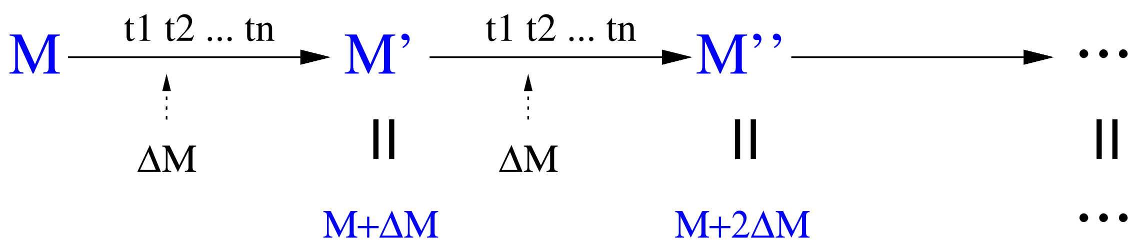
Now, if $M \xrightarrow{t_1 t_2 \ldots t_n}$ and $M \leq M'$, then $M' \xrightarrow{t_1 t_2 \ldots t_n}$.

Coverability and Transition Sequences (2/2)

Assume that $M' \in reach(M)$ (with $M < M'$). Then clearly there is some sequence of transitions $t_1 t_2 \ldots t_n$ such that $M \xrightarrow{t_1 t_2 \ldots t_n} M'$. Thus, there is a marking M'' with $M' \xrightarrow{t_1 t_2 \ldots t_n} M''$.

Let $\Delta M := M' - M$ (place-wise difference). Because $M < M'$, the values of ΔM are non-negative and at least one value is non-zero.

Clearly, $M'' = M' + \Delta M = M + 2\Delta M$.



By firing the transition sequence $t_1 t_2 \ldots t_n$ repeatedly we can “pump” an arbitrary number of tokens to all the places having a non-zero marking in $\triangle M$.

The basic idea for constructing the coverability graph is now to replace the marking M' with a marking where all the places with non-zero tokens in $\triangle M$ are replaced by ω.

Coverability Graph Algorithm (1/2)

COVERABILITY-GRAPH$(\langle P, T, F, W, M_0 \rangle)$

$\langle V, E, v_0 \rangle := \langle \{M_0\}, \emptyset, M_0 \rangle$;
$Work$: set $:= \{M_0\}$;
while $Work \neq \emptyset$
do select M from $Work$;
    $Work := Work \setminus \{M\}$;
    for $t \in \mathrm{enabled}(M)$
    do $M' := \mathrm{fire}(M, t)$;
        $M' := \mathrm{AddOmegas}(M, t, M', V, E)$;
        if $M' \notin V$
          then $V := V \cup \{M'\}$
                $Work := Work \cup \{M'\}$;
        $E := E \cup \{\langle M, t, M' \rangle\}$;
return $\langle V, E, v_0 \rangle$;

The coverability graph algorithm is almost exactly the same as the reachability graph algorithm, with the addition of the call to subroutine $\mathrm{AddOmegas}(M, t, M', V, E)$, where all the details w.r.t. coverability graphs are contained. As for the implementation of $Work$, the same comments as for the reachability graph apply.

Coverability Graph Algorithm (2/2)

The following notations are used in the AddOmegas subroutine:

- $M'' \rightarrow_E M$ iff $\langle M'', t, M \rangle \in E$ for some $t \in T$.

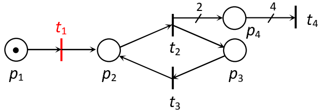
- $M'' \rightarrow_{E^*} M$ iff
  $\exists n \geq 0 \colon \exists M_0, M_1, \ldots, M_n \colon M'' = M_0 \rightarrow_E M_1 \rightarrow_E M_2 \rightarrow_E \cdots \rightarrow_E M_n = M$.

AddOmegas(M, t, M', V, E)

1 **for** $M'' \in V$
2 **do if** $M'' < M'$ and $M'' \rightarrow_{E^*} M$
3   **then** $M' := M' + ((M' - M'') \cdot \omega)$;
4 **return** M';

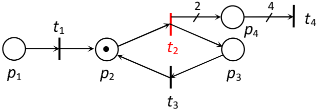
Line 3 causes all places whose marking in M' is strictly larger than in the "parent" M'' to contain ω, while markings of other places remain unchanged.

Example



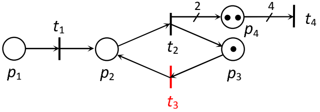
〉[1 0 0 0] $\xrightarrow{t_1}$ [0 1 0 0]

Example



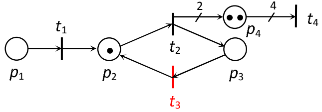
$[1\ 0\ 0\ 0] \xrightarrow{t_1} [0\ 1\ 0\ 0] \xrightarrow{t_2} [0\ 0\ 1\ 2]$

Example



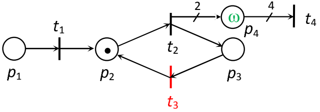
[1 0 0 0] $\xrightarrow{t_1}$ [0 1 0 0] $\xrightarrow{t_2}$ [0 0 1 2] $\xrightarrow{t_3}$ [0 1 0 2]

Example



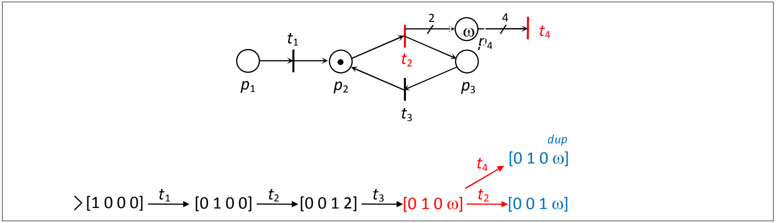
> [1 0 0 0] $\xrightarrow{t_1}$ [0 1 0 0] $\xrightarrow{t_2}$ [0 0 1 2] $\xrightarrow{t_3}$ [0 1 0 2]

Example

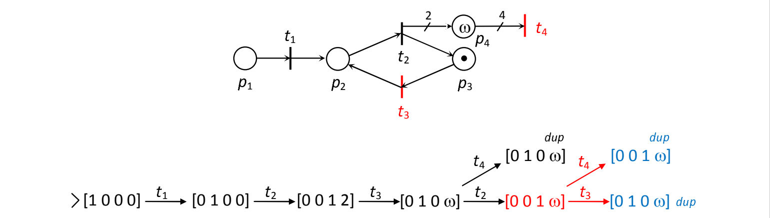


$\rangle$ [1 0 0 0] $\xrightarrow{t_1}$ [0 1 0 0] $\xrightarrow{t_2}$ [0 0 1 2] $\xrightarrow{t_3}$ [0 1 0 ω]

Example



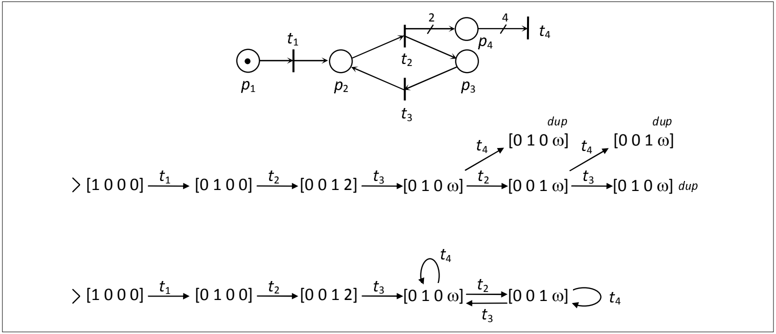
Example



t1
t2
t3
t4
2
4
ω
p1
p2
p3
p4
dup
dup
[1 0 0 0]
[0 1 0 0]
[0 0 1 2]
[0 1 0 ω]
[0 1 0 ω]
[0 0 1 ω]
[0 0 1 ω]
[0 1 0 ω]
dup

Example

From the coverability tree we obtain the coverability graph.



Nodes of the coverability graph

In general a node of the coverability graph is ω-marking $M_\omega \in (\mathbb{N} \cup \{\omega\})^m$.

Definition (Set of markings represented by an ω-marking)

Given an ω-marking M_ω we denote

$$\mathcal{M}(M_\omega) = \{M \in \mathbb{N}^m \mid M(p) = M_\omega(p) \text{ if } M_\omega(p) \neq \omega\}.$$

Ex1: $M_\omega = [3\ 0\ \omega]^T \quad \longrightarrow \quad \mathcal{M}(M_\omega) = \{[3\ 0\ x]^T \mid x \in \mathbb{N}\}.$

Ex2: $M_\omega = [3\ 0\ 1]^T \quad \longrightarrow \quad \mathcal{M}(M_\omega) = \{[3\ 0\ 1]^T\}.$

What does the coverability graph tell us?

Proposition

Given a marked net $\langle N, M_0 \rangle$ let $\mathcal{G}$ be its coverability graph with set of nodes X constructed using the previous algorithm.

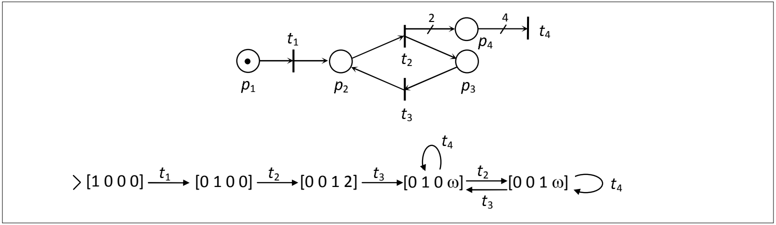
- $R(N, M_0) \subseteq \bigcup_{M_\omega \in X} \mathcal{M}(M_\omega)$.
- $L(N, M_0) \subseteq L(\mathcal{G})$.

Two main informations from the reachability graph

- M is reachable $\implies$ there exists in $\mathcal{G}$ a node M_ω with $M \in \mathcal{M}(M_\omega)$.
- $\sigma \in L(N, M_0) \implies \delta(M_0, \sigma)$ is defined in $\mathcal{G}$.

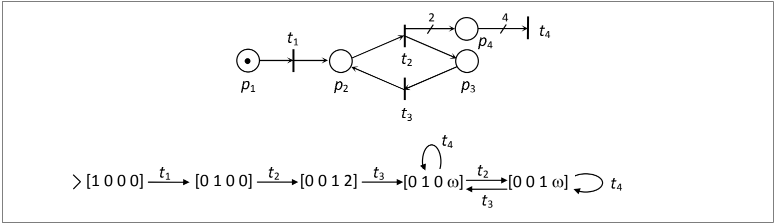
Note that the coverability graph provides a necessary but not sufficient condition for marking reachability and existence of a firable sequence.

Example



- $M = [0\ 1\ 1\ 0]^T$ is not reachable: no M_ω in $\mathcal{G}$ such that $M \in \mathcal{M}(M_\omega)$.
- $M = [0\ 1\ 0\ 20]^T$ is reachable: note that $M \in \mathcal{M}([0\ 1\ 0\ \omega]^T)$.
- $M = [0\ 1\ 0\ 21]^T$ is not reachable even if $M \in \mathcal{M}([0\ 1\ 0\ \omega]^T)$ (always even number of tokens in p_4).

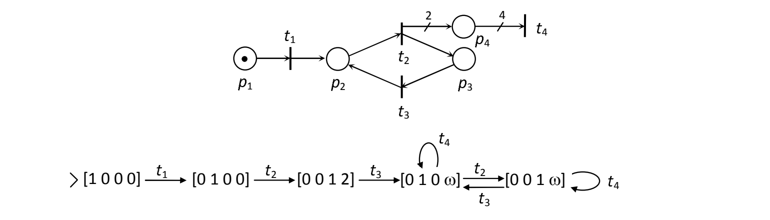
Example



- $t_1 t_2 t_4$ is not a firing sequence: $\delta(M_0, t_1 t_2 t_4)$ is not defined in $\mathcal{G}$.
- $t_1 t_2 t_3$ is a firing sequence: note that $\delta(M_0, t_1 t_2 t_3)$ is defined in $\mathcal{G}$.
- $t_1 t_2 t_3 t_4$ is not a firing sequence even if $\delta(M_0, t_1 t_2 t_3 t_4)$ is defined in $\mathcal{G}$. Transition t_4 needs 4 tokens to fire, hence t_2 must fire at least twice.

Boundeness

- Place p if unbounded $\iff$ there exists in $\mathcal{G}$ a node M_ω with $M_\omega(p) = \omega$.
- Place p is k_p bounded $\iff$ $k_p = \max\{M_\omega(p)\}$ for all M_ω in $\mathcal{G}$.



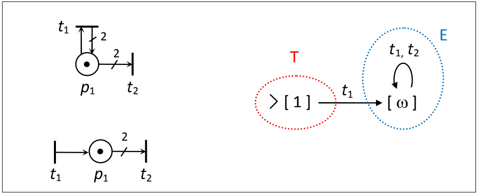
Example

- Places p_1, p_2, p_3 are 1-bounded.
- Place p_3 is unbounded.

The coverability graph provides a necessary and sufficient condition for boundedness.

Liveness

- A transition t is quasi-live $\iff$ an arc t appears in the graph.
- A transition t is live $\Longrightarrow$ an arc t appears in all ergodic components.



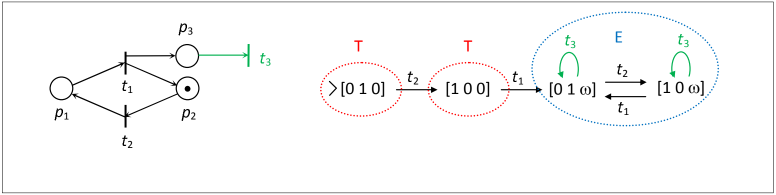
Example

- Two different nets with the same coverability graph: all transitions are quasi-live in both nets.
- The necessary condition for liveness is satisfied but in the first net no transition is live, while the second net is live.

The coverability graph provides a necessary and sufficient condition for quasi-liveness but only a necessary condition for liveness.

Reversibility

- $\langle N, M_0 \rangle$ is reversible $\Longrightarrow$ a marking M_ω such that $M_0 \in \mathcal{M}(M_\omega)$ appears in all ergodic components of the graph.



Example

- Two nets (with/without t_3): the necessary condition for reversibility is satisfied for both but the net with t_3 is reversible, the net without t_3 is not.

The coverability graph provides only a necessary condition for reversibility.

Are these properties decidable?

If a net is bounded, marking reachability and all other properties are decidable by analysis of the reachability graph.

If a net is unbounded, the coverability graph does not provide a test for marking reachability, liveness and reversibility.

Are these properties decidable for unbounded nets with some other procedure?

The answer is yes: it follows from the decidability of marking reachability that was proved by Kosaraju (1982).

However the procedure (and the proof) is rather complicated. If interested read: C. Reutenauer, *The Mathematics of Petri Nets*, Prentice Hall, 1990.

Termination of the Coverability Graph Algorithm (1/2)

Dickson's lemma: Every infinite sequence $u_1 u_2 \ldots$ of n-tuples of natural numbers contains an infinite subsequence $u_{i_1} \leq u_{i_2} \leq u_{i_3} \leq \ldots$.

Proof: By induction on n.

Base: $n = 1$. Let u_{i_1} be the smallest of $u_1 u_2 \ldots$, let u_{i_2} be the smallest of $u_{i_1+1} u_{i_1+2} \ldots$ etc.

Step: $n > 1$. Consider the projections $v_1 v_2 \ldots$ and $w_1 w_2 \ldots$ of $u_1 u_2 \ldots$ onto the first $n-1$ components and the last component, respectively. By induction hypothesis, there is an infinite subsequence $v_{j_1} \leq v_{j_2} \leq v_{j_3} \leq \ldots$.
Consider the infinite sequence $w_{j_1} \leq w_{j_2} \leq \ldots$. By induction hypothesis, this sequence has an infinite subsequence $w_{i_1} \leq w_{i_2} \leq \ldots$. So we have $u_{i_1} \leq u_{i_2} \leq u_{i_3} \leq \ldots$.

Termination of the Coverability Graph Algorithm (2/2)

Theorem: The Coverability Graph Algorithm terminates.

Proof: Assume that the algorithm does not terminate. We derive a contradiction.

If the algorithm does not terminate, then the Coverability Graph is infinite. Since every node of the graph has at most $|T|$ successors, the graph contains an infinite path $\Pi = M_1 M_2 \ldots$.

If an ω-marking M_i of Π satisfies $M_i(p) = \omega$ for some place p, then $M_{i+1}(p) = M_{i+2}(p) = \ldots = \omega$.

So Π contains an ω marking M_j such that all markings $M_{j+1}, M_{j+2}, \ldots$ have ω's at exactly the same places as M_j. Let Π' be the suffix of Π starting at M_j.

Consider the projection $\Pi'' = m_j m_{j+1} \ldots$ of Π' onto the non-ω places. Let n be the number of non-ω places. Π'' is an infinite sequence of distinct n-tuples of natural numbers.

By Dickson's lemma, this sequence contains markings M_k, M_l such that $k < l$ and $M_k \leq M_l$. This is a contradiction, because, since $M_k \neq M_l$, when executing $\mathsf{AddOmegas}(M_{l-1}, t, M_l, V, E)$ the algorithm adds at least one ω to M_l

Remarks on the Coverability Graph Algorithm

If the reachability graph is finite, the algorithm $\mathsf{AddOmegas}(M, t, M', V, E)$ will always return M' as its output (i.e., the third parameter).

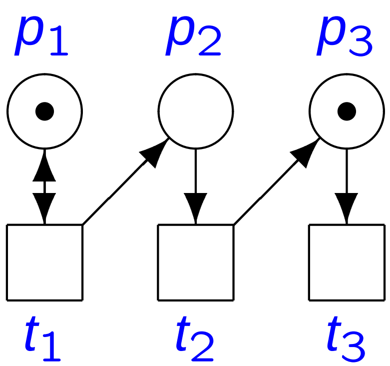
In this case the coverability graph algorithm will return the reachability graph (but it will run more slowly).

Implementations of the algorithm are bound to be slow because of the **for** loop in $\mathsf{AddOmegas}$, which has to traverse the potentially large size of the graph.

The result of the algorithm is not unique, e.g. it depends on the implementation of $Work$ and on the exact order of fired transitions on line 5 of the main routine.

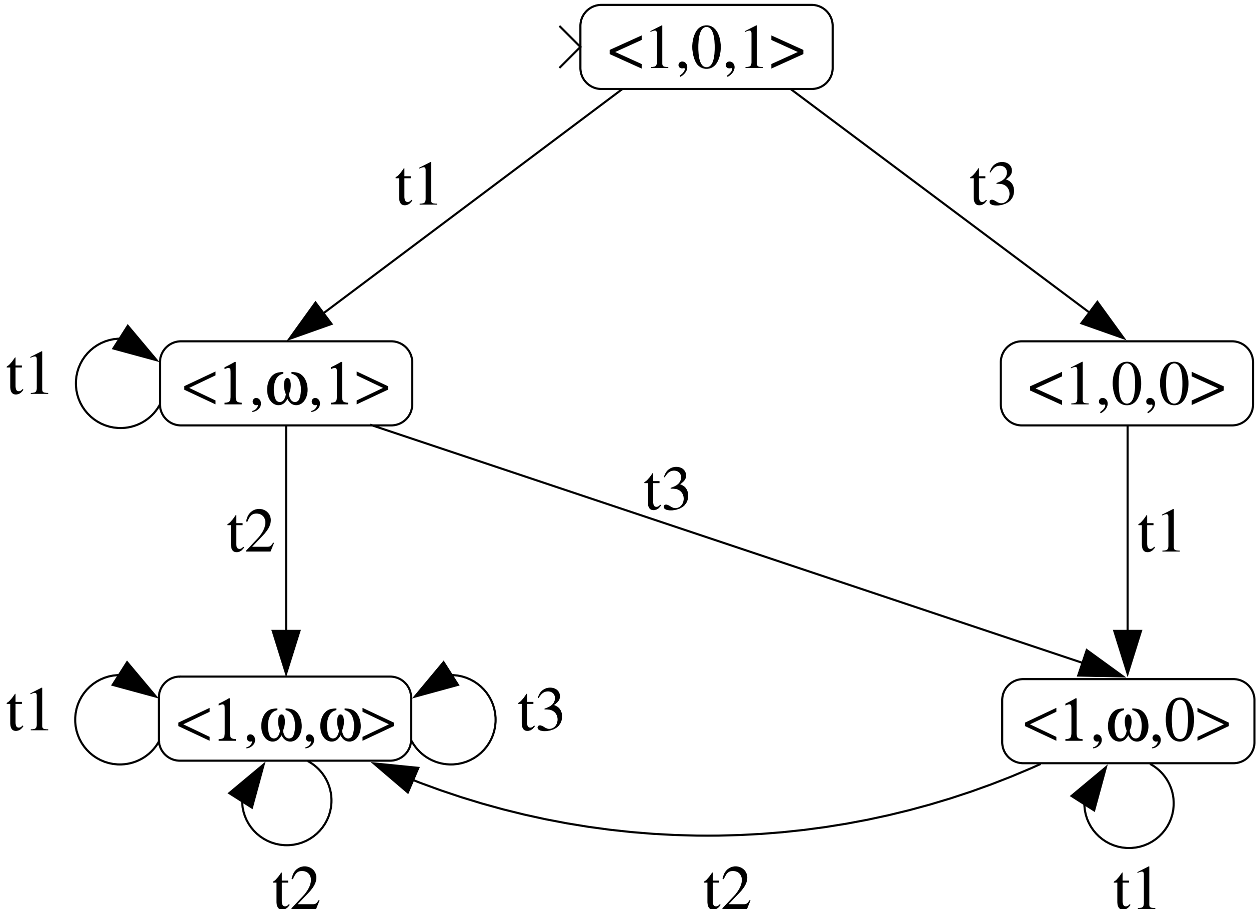
Example 1: Coverability Graph

Recall the P/T-net example given in the previous lecture:



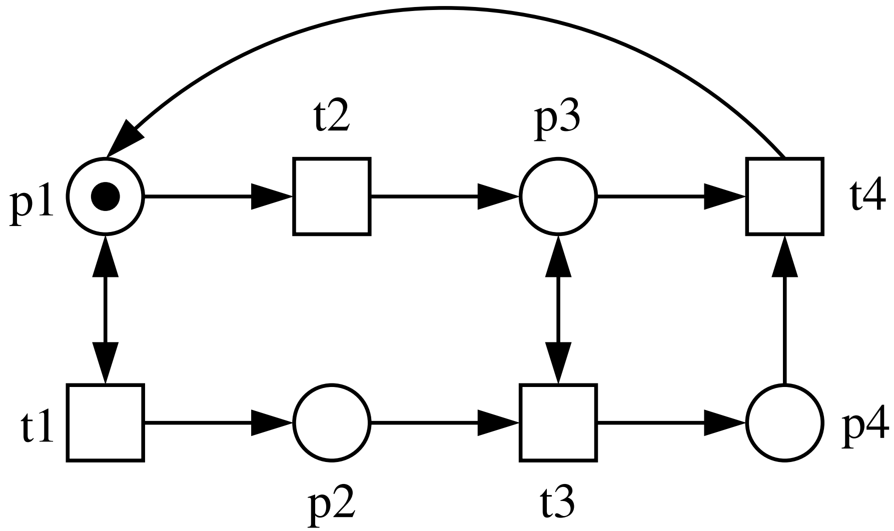
We will now compute the coverability graph for it.

Example 1: Coverability Graph

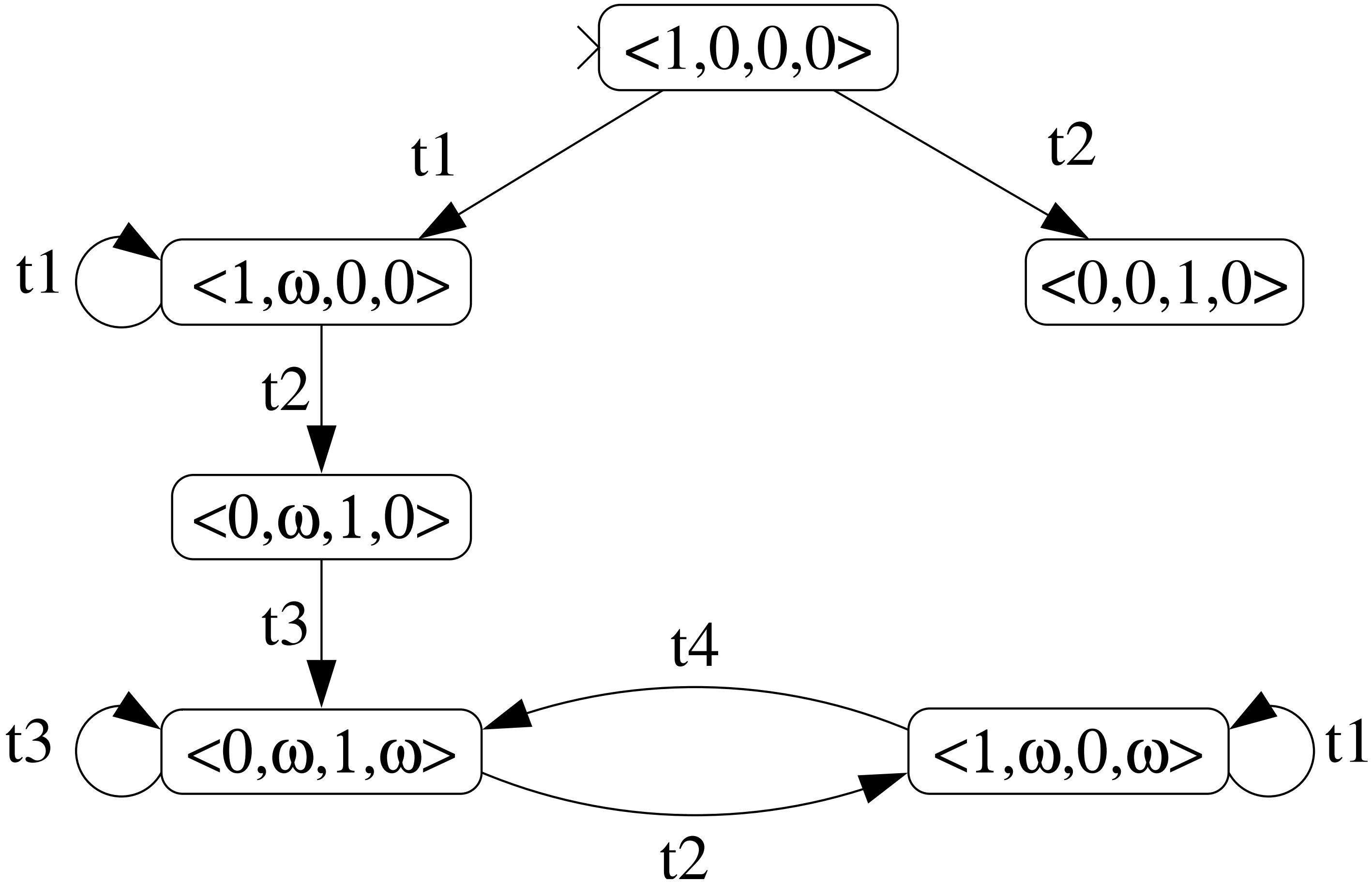


Example 2

Consider the following P/T-net. We will now compute a coverability graph for it.



Example 2: Coverability graph



Reachability and coverability graphs: Comparison (1)

Let $N = \langle P, T, F, W, M_0 \rangle$ be a net.

The reachability graph has the following fundamental property:

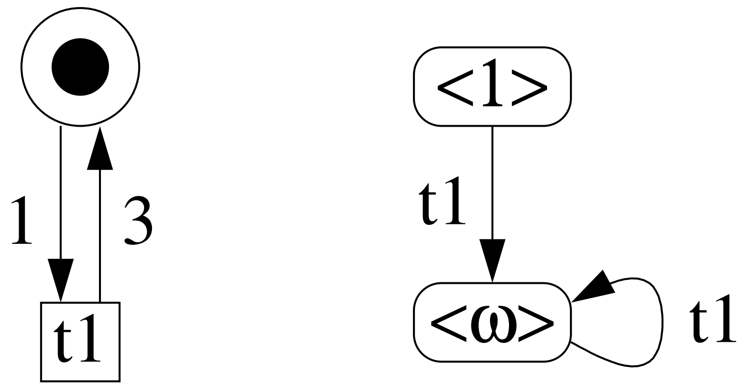
> A marking M of N is reachable *if and only if* M is a vertex of the reachability graph of N.

The coverability graph has the following fundamental property:

> If a marking M of N is reachable, then M is covered by some vertex of the coverability graph of N.

Notice that the first property is an equivalence, the second one an implication!

More specifically, the reverse implication *does not* hold: A marking that is covered by some vertex of the coverability graph is not necessarily reachable, as shown by the following example:



In the net, only markings with an odd number of tokens are reachable, but markings with an even number of tokens are also covered.

Reachability and coverability graphs: Comparison (2)

The reachability graph captures exact information about the reachable markings
(but its computation may not terminate).

The coverability graph computes an overapproximation
(but remains exact as long as the number of markings is finite).

Summary: Which properties can we check so far?

Reachability: Given some marking M and a net N, is M reachable in N?
More generally: Given a set of markings $\mathcal{M}$, is some marking of $\mathcal{M}$ reachable?

Application: This is often used to check whether some 'bad' state can occur (classical example: violation of mutual exclusion property) if $\mathcal{M}$ is taken to be the set of 'error' states. Sometimes (as in the man/wolf/etc example), this analysis can check for the existence of a solution to some problem.

Using the reachability graph: Exact answer is obtained.

Using the coverability graph: Approximate answer. When looking for 'bad' states, this analysis is safe in the sense that bad states will not be missed, but the graph may indicate 'spurious' errors.

Summary (cont'd)

Finding paths: Given a reachable marking M, find a firing sequence that leads from M_0 to M.

Application: Used to supplement reachability queries. If M represents an error state, the firing sequence can be useful for debugging. When solving puzzles, the path represents actions leading to the solution.

Using the reachability graph: Find a path from M_0 to M in the graph, obtain sequence from edge labels.

Using the coverability graph: Not so suitable – edges may represent 'shortcuts' (unspecified repetitions of some loop).

Summary (cont'd)

Enabledness: Given some transition t, is there a reachable marking in which t is enabled?
(Sometimes, t is called dead if the answer is no. Actually, this is a special case of reachability.)

Application: Check whether some 'bad' action is possible. Also, is some desirable action is never enabled, a 'no' answer is an indication of some problem with the model.
In some Petri-net tools, checking for enabledness is easier to specify than checking for reachability. In that case, reachability queries can be framed as enabledness queries by adding 'artificial' transitions that can fire iff a given marking is reachable.

Using the reachability graph: Check whether there is an edge labeled with t.

Using the coverability graph: ?

Summary (cont'd)

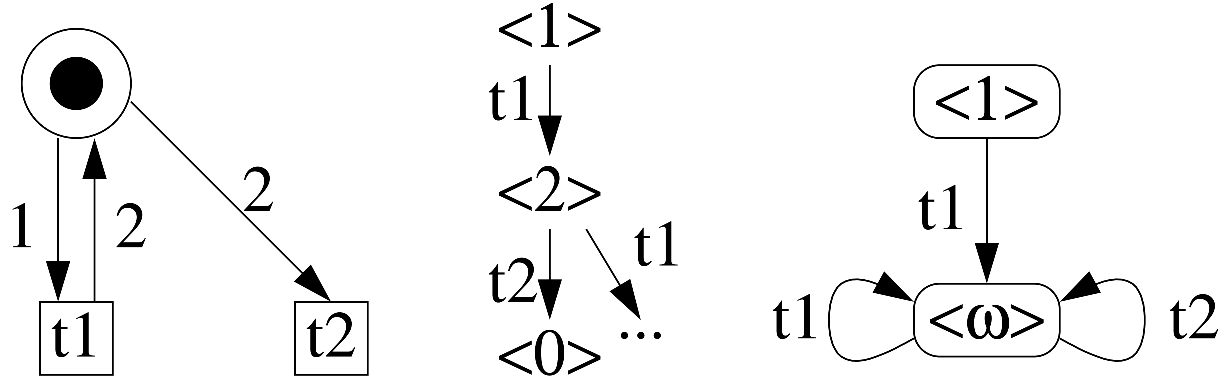
Deadlocks: Given a net N, is N deadlock-free?

> A marking M of a Place/Transition net $N = \langle P, T, F, W, M_0 \rangle$ is called a deadlock if no transition $t \in T$ is enabled in M. A net N is deadlock-free if no reachable marking is a deadlock

Application: Deadlocks tend to indicate errors (classical example: philosophers may starve).

Using the reachability graph: Check whether there is a vertex without an outgoing edge.

Using the coverability graph: Unsuitable – the graph may miss deadlocks!



Summary (cont'd)

Boundedness: Given a net N, is there a constant k such that N is k-safe? Otherwise, which places can assume an unbounded number of tokens?

Application: If tokens represent available resources, unbounded numbers of tokens may indicate some problem (e.g. a resource leak). Also, this property should be checked *before* computing the reachability graph!

Using the reachability graph: Unsuitable, computation may not terminate.

Using the coverability graph:

> A place p can assume an unbounded number of tokens iff the coverability graph contains a vertex M where $M(p) = \omega$.
> Iff no vertex with an ω exists, then the net is k-safe, where k is the largest natural number in a marking of the graph.

What is missing? (Outlook)

Sometimes, properties mentioned in the summary can be checked even *without constructing the reachability graph* (which can be pretty large, after all).

Methods for doing this are collectively called structural analyses

So far, we have not learnt how to express (and check) properties like these:

Marking M can be reached infinitely often.

Whenever transition t occurs, transition t' occurs later.

No marking with some property x occurs before some marking with property y has occurred.

Properties like these can be expressed using temporal logic.

Structural analysis of P/T nets

Structural analysis of P/T nets

Structural Analysis: Motivation

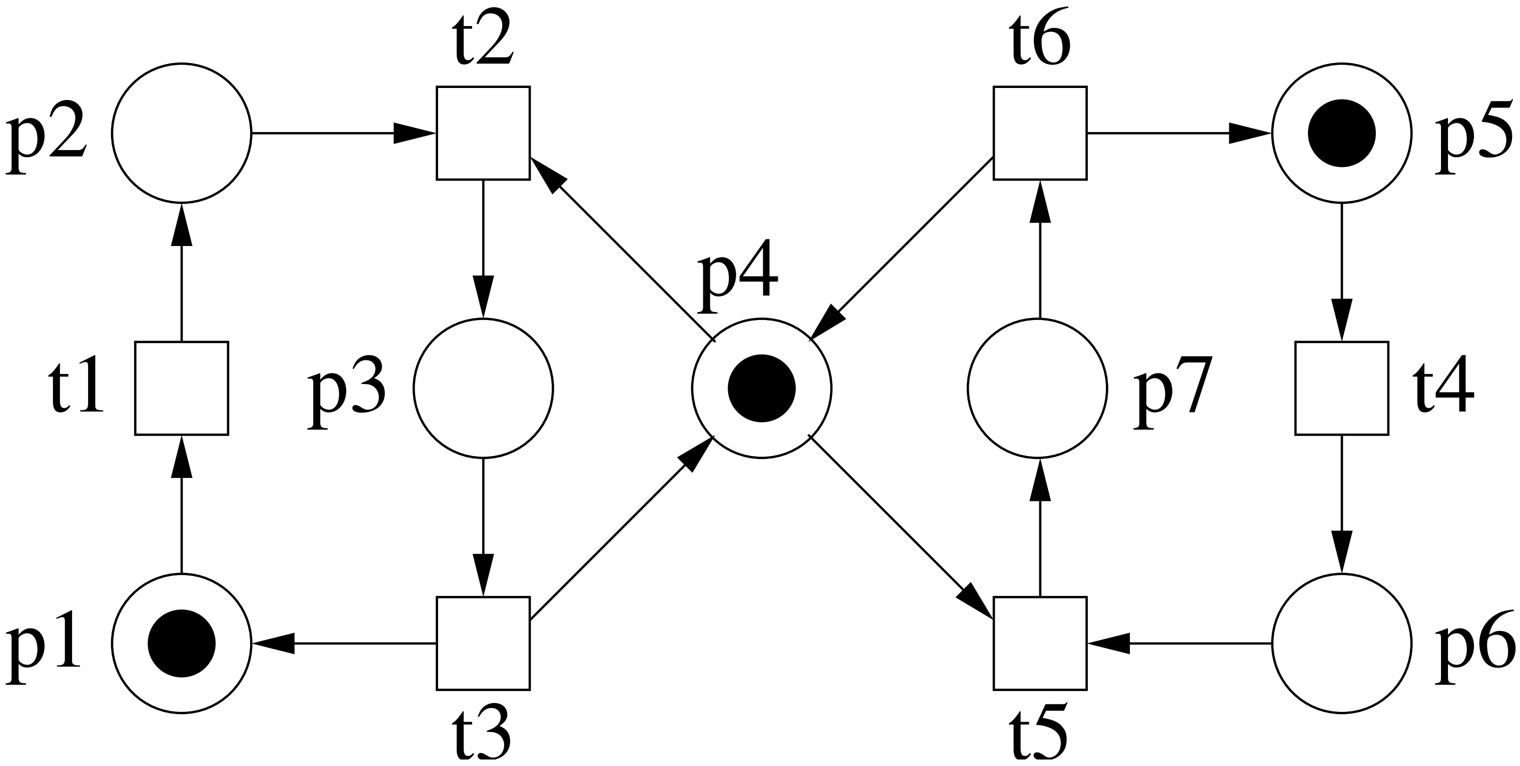
We have seen how properties of Petri nets can be proved by constructing the reachability graph and analysing it.

However, the reachability graph may become huge: exponential in the number of places (if it is finite at all).

Structural analysis makes it possible to prove some properties *without* constructing the reachability graph. The main techniques are:

- Place invariants
- Traps

Example 1



Incidence Matrix: Definition

Let $N = \langle P, T, F, W, M_0 \rangle$ be a P/T net. The corresponding incidence matrix $C_N : P \times T \to \mathbb{Z}$ is the matrix whose rows correspond to places and whose columns correspond to transitions. Column $t \in T$ denotes how the firing of t affects the marking of the net: $C(t,p) = W(t,p) - W(p,t)$.

The incidence matrix of the example from the previous slide:

$$
\begin{array}{c}
\begin{pmatrix}
t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \\
-1 & 0 & 1 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & -1 & 1 \\
0 & 0 & 0 & -1 & 0 & 1 \\
0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & -1
\end{pmatrix}
\begin{array}{c}
\\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7
\end{array}
\end{array}
$$

Markings as vectors

Let us now write marking as column vectors. E.g., the initial marking is $M_0 = (1\ 0\ 0\ 1\ 1\ 0\ 0)^T$.

Likewise, we can write firing counts as column vectors with one entry for each transition. E.g., if t_1, t_2, and t_4 are to fire once each, we can express this with $u = (1\ 1\ 0\ 1\ 0\ 0)^T$.

Then, the result of firing these transitions can be computed as $M_0 + C \cdot u$.

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Caveat

Notice: Bi-directional arcs (an arc from a place to a transition and back) cancel each other out in the matrix!

Thus, when a marking arises as the result of a matrix equation (like on the previous slide), this does not guarantee that the marking is reachable!

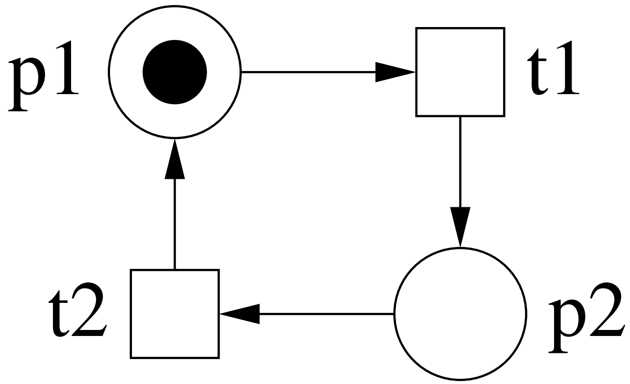
I.e., the markings obtained by the incidence markings are an over-approximation of the actual reachable markings (compare coverability graphs. . .).

However, we *can* sometimes use the matrix equations to show that a marking M is unreachable, i.e. if $M_0 + Cu = M$ has no natural solution for u.

Note: When we are talking about natural (integral) solutions of equations, we mean those whose components are natural (integral) numbers.

Example 2

Consider the following net and the marking $M = (1\ 1)^T$.



$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

has no solution, and therefore M is not reachable.

Invariants

The solutions of the equation $Cu = \mathbf{0}$ are called transition invariants (or: T-invariants). The natural solutions indicate (possible) loops.

For instance, in Example 2, $u = (1\ 1)^T$ is a T-invariant.

The solutions of the equation $C^T x = \mathbf{0}$ are called place invariants (or: P-invariants). A proper P-invariant is a solution of $C^T x = \mathbf{0}$ if $x \neq \mathbf{0}$.

For instance, in Example 1, $x_1 = (1\ 1\ 1\ 0\ 0\ 0\ 0)^T$, $x_2 = (0\ 0\ 1\ 1\ 0\ 0\ 1)^T$, and $x_3 = (0\ 0\ 0\ 0\ 1\ 1\ 1)^T$ are all (proper) P-invariants.

A P-invariant indicates that the number of tokens in all reachable markings satisfies some linear invariant (see next slide).

Properties of P-invariants

Let M be marking reachable with a transition sequence whose firing count is expressed by u, i.e. $M = M_0 + Cu$. Let x be a P-invariant. Then, the following holds:

$$M^T x = (M_0 + Cu)^T x = M_0^T x + (Cu)^T x = M_0^T x + u^T C^T x = M_0^T x$$

For instance, invariant x_2 means that all reachable markings M satisfy (reverting back to the function notation for markings):

$$M(p_3) + M(p_4) + M(p_7) = M_0(p_3) + M_0(p_4) + M_0(p_7) = 1 \qquad (1)$$

As a consequence, a P-invariant in which all entries are either 0 or 1 indicates a set of places in which the number of tokens remains unchanged in all reachable markings.

Note that multiplying an invariant by a constant or component-wise addition of two invariants will again yield a P-invariant. That is, the set of all invariants is a *vector space*.

We can use P-invariants to prove mutual exclusion properties:

> According to equation 1, in every reachable marking of Example 1 exactly one of the places p_3, p_4, and p_7 is marked. In particular, p_3 and p_7 cannot be marked concurrently!

Another example: Mutual exclusion with token passing (demo)

More remarks on P-invariants

P-invariants can also be useful as a *pre-processing step* for reachability analysis.

Suppose that when computing the reachability graph, the marking of a place is normally represented with n bits of storage. E.g. the places p_3, p_4, and p_7 together would require $3n$ bits.

However, as we have discovered invariant x_2, we know that exactly one of the three places is marked in each reachable marking.

Thus, we just need to store in each marking *which* of the three is marked, which required just 2 bits.

Algorithms for P-invariants

A basis of the set of all invariants can be computed using linear algebra.

There is an algorithm called "Farkas Algorithm" (by *J. Farkas*, 1902) to compute a set of so called minimal P-invariants (see the enxt slides). These are positive place invariants from which any other positive invariant can be computed by a linear combination.

Unfortunately there are P/T-nets with an exponential number of minimal P-invariants (in the number of places of the net). Thus the Farkas algorithm needs (at least) exponential time in the worst case.

The INA tool of the group of *Peter Starke* (Humboldt University of Berlin) contains a large number of algorithms for structural analysis of P/T-nets, including invariant generation.

Farkas Algorithm

Input: the incidence matrix C with n rows (places), and m columns (transitions).

$(C \mid E_n)$ denotes the augmentation of C by a $n \times n$ identity matrix (last n columns).

$D_0 := (C \mid E_n)$;

for $i := 1$ **to** m **do**

  for d_1, d_2 rows in D_{i-1} such that $d_1(i)$ and $d_2(i)$ have opposite signs **do**

    $d := |d_2(i)| \cdot d_1 + |d_1(i)| \cdot d_2$; (* $d(i) = 0$ *)

    $d' := d/\text{gcd}(d(1), d(2), \ldots, d(m+n))$;

    augment D_{i-1} with d' as last row;

  endfor;

  delete all rows of the (augmented) matrix D_{i-1} whose i-th component is different from 0, the result is D_i;

endfor;

delete the first m columns of D_m

An example

Incidence matrix

$$C = \begin{pmatrix} -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

$$D_0 = (C \mid E_5) = \left(\begin{array}{cccc|ccccc} -1 & 1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

Addition of the rows 1 and 2, 1 and 4, 2 and 5, 4 and 5:

$$D_1 = \left(\begin{array}{cccc|ccccc} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -2 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & -1 & 2 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right)$$

Addition of rows 3 und 4:

$$D_2 = \left(\begin{array}{cccc|ccccc} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}\right)$$

$$D_3 = D_4 = \left(\begin{array}{cccc|ccccc} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}\right)$$

Minimal P-invariants are $(1, 1, 0, 0, 0)$ and $(0, 0, 0, 1, 1)$.

Biological interpretations

A P-invariant can be regarded as a token conservation component.

Since in the biological interpretation the token represent molecules (or levels of concentration) this means that a P-invariant represents conservation of mass.

A T-invariant identifies a set of transition firings which can return the net to the same marking.

In the biological interpretation a feasible T-invariant identifies a set of reactions which may return a process to a given state and understanding this may provide insight into the behaviour.

Moreover, if the system has a steady state behaviour (e.g. a metabolic network) then the T-invariant gives relative occurrence rates for the reactions involved.

An example with many P-invariants

Incidence matrix for a net with $2n$ places:

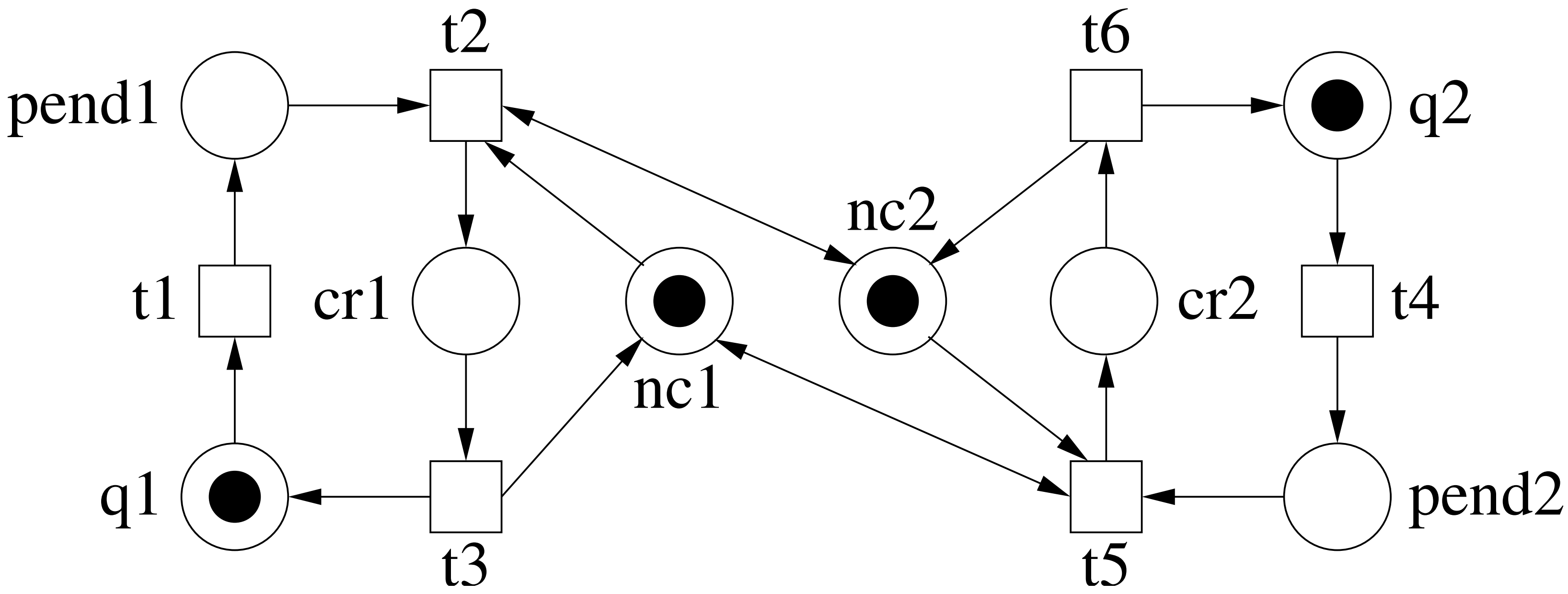
$$C = \begin{pmatrix} -1 & 0 & 0 & & 0 & 1 \\ -1 & 0 & 0 & & 0 & 1 \\ 1 & -1 & 0 & \cdots & 0 & 0 \\ 1 & -1 & 0 & & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \\ 0 & 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{pmatrix}$$

$(y_1, 1-y_1, y_2, 1-y_2, \ldots, y_n, 1-y_n)$ is an invariant for every $y_1, y_2, \ldots, y_n \in \{0,1\}$, and so there are 2^n minimal P-invariants.

This example shows that the number of minimal P-invariants can be exponential in the size of the net. So Farkas algorithm may need exponential time.

Example 3

Consider the following attempt at a mutual exlusion algorithm for cr_1 and cr_2:



The idea is to achieve mutual exclusion by entering the critical section only if the other process is not already there.

Thus, we want to prove that in all reachable markings M:

$$M(cr_1) + M(cr_2) \leq 1$$

The P-invariants we can derive in the net yield:

$$\begin{aligned}
M(q_1) + M(pend_1) + M(cr_1) &= 1 \quad (2)\\
M(q_2) + M(pend_2) + M(cr_2) &= 1 \quad (3)\\
M(cr_1) + M(nc_1) &= 1 \quad (4)\\
M(cr_2) + M(nc_2) &= 1 \quad (5)
\end{aligned}$$

But try as we might, we cannot show the desired property just with these four equations!

Traps

Definition: Let $\langle P, T, F, W, M_0 \rangle$ be a P/T net.
A trap is a set of places $S \subseteq P$ such that $S^\bullet \subseteq {}^\bullet S$.

In other words, each transition which removes tokens from a trap must also put at least one token back to the trap.

A trap S is called marked in marking M iff for at least one place $s \in S$ it holds that $M(s) \geq 1$.

Note: If a trap S is marked in M_0, then it is also marked in all reachable markings.

In Example 3, $S_1 = \{nc_1, nc_2\}$ is a trap.

The only transitions that remove tokens from this set are t_2 and t_5. However, both also add new tokens to S_1.

S_1 is marked initially and therefore in all reachable markings M. Thus:

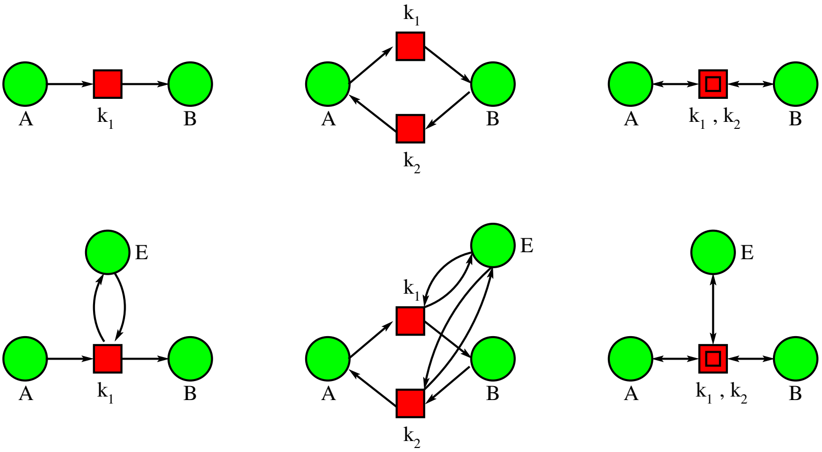
$$M(nc_1) + M(nc_2) \geq 1 \tag{6}$$

Traps can be useful in combination with place invariants to recapture information lost in the incidence matrix due to the cancellation of self-loop arcs.

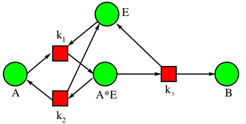
Here: Adding (4) and (5) and subtracting (6) yields $M(cr_1) + M(cr_2) \leq 1$, which proves the mutual exclusion property.

Petri nets: Simple Reactions

1. $A \longrightarrow B$
2. $A \rightleftharpoons B$
3. $A \xrightarrow{E} B$

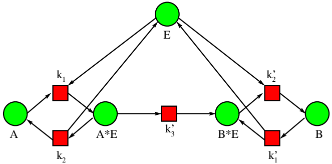


Petri nets: Enzyme Reactions



$$A + E \xrightleftharpoons[k_2]{k_1} A{\cdot}E$$

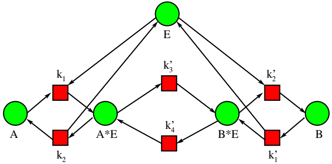
$$A{\cdot}E \xrightarrow{k_3} B + E$$



$$A + E \xrightleftharpoons[k_2]{k_1} A{\cdot}E$$

$$A{\cdot}E \xrightarrow{k_3'} B{\cdot}E$$

$$B{\cdot}E \xrightleftharpoons[k_2']{k_1'} B + E$$



$$A + E \xrightleftharpoons[k_2]{k_1} A{\cdot}E$$

$$A{\cdot}E \xrightleftharpoons[k_4']{k_3'} B{\cdot}E$$

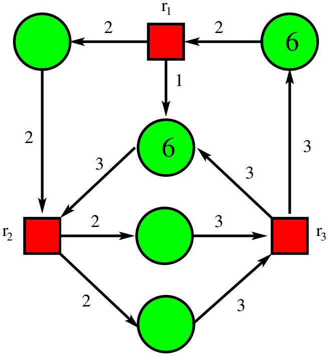
$$B{\cdot}E \xrightleftharpoons[k_2']{k_1'} B + E$$

Petri nets: Incidence Matrix

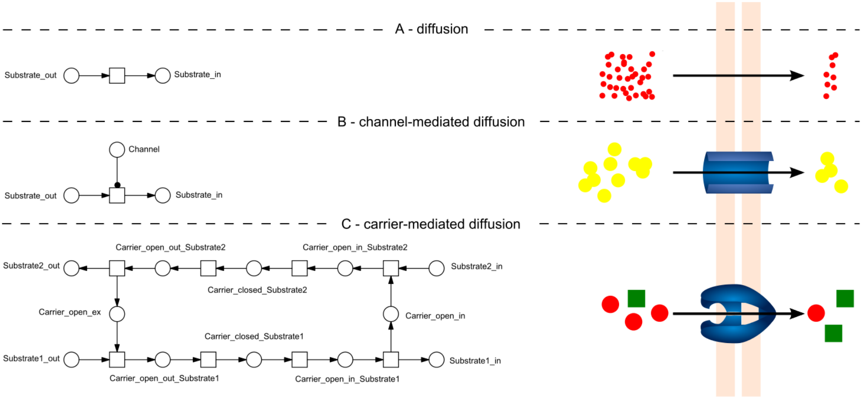
The incidence matrix coinsides for metabolic networks with the stoichiometric matrix.

$$2\,\mathrm{C} \xrightarrow{r1} A + 2B$$
$$3\,\mathrm{A} + 2\,\mathrm{B} \xrightarrow{r2} 2D + 2E$$
$$3\,\mathrm{D} + 3\,\mathrm{E} \xrightarrow{r3} 3A + 3\,\mathrm{C}$$

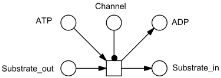
$$\begin{array}{c} \\ \mathrm{A} \\ \mathrm{B} \\ \mathrm{C} \\ \mathrm{D} \\ \mathrm{E} \end{array} \begin{array}{c} \begin{array}{ccc} \mathrm{r}_1 & \mathrm{r}_2 & \mathrm{r}_3 \end{array} \\ \begin{pmatrix} 1 & -3 & 3 \\ 2 & -2 & 0 \\ -2 & 0 & 3 \\ 0 & 2 & -3 \\ 0 & 2 & -3 \end{pmatrix} \end{array}$$

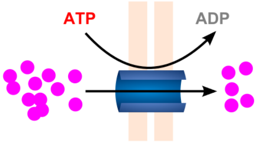


Passive Transport

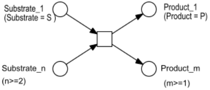


Active Transport



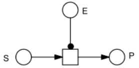


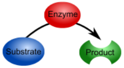
Chemical Reaction



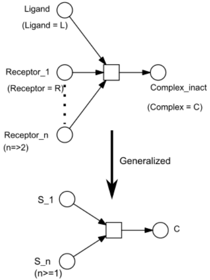
$S_1 + S_2 + \ldots + S_n$ (n>=2)
$\rightarrow P_1 + P_2 + \ldots + P_m$ (m>=1)

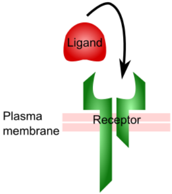
Enzymatic Reaction



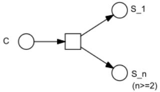


Association

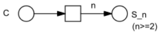




Dissociation

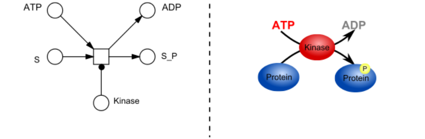


$C \rightarrow S_1 + S_2 + \ldots + S_n$ (n>=2)

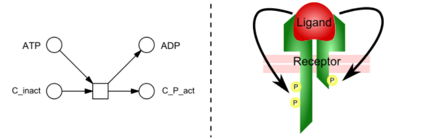


When: $S_1 = S_2 = \ldots = S_n$ (n>=2)

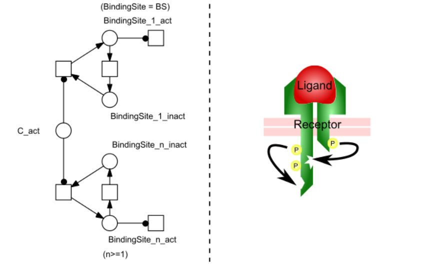
Phosphorylation



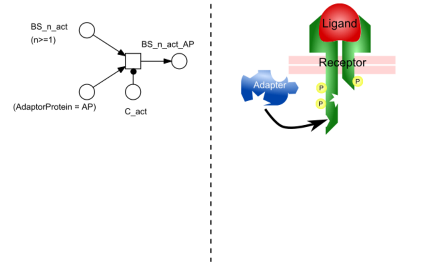
Autophosphorylation



Activation of Functional Sites

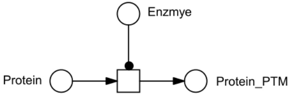


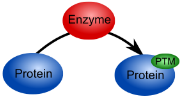
Gathering Functionality by Adaptor Proteins



PTM: addition of functional groups

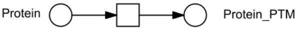
A - enzymatic

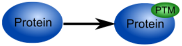




e.g.,
addition of hydrophobic groups for membrane localization (myristoylation, palmitoylation, isoprenylatio etc.);
addition of cofactors for enhanced enzymatic activity (Lipoylation, flavin, heme C, phosphopantetheinylation, retinylidene Schiff base etc.);
addition of smaller chemical groups (acylation, amide bond formation, carboxylation, glycosylation, hydroxylation, oxidation, phosphate ester formation etc.)

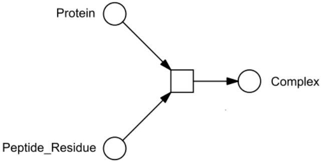
A - non-enzymatic

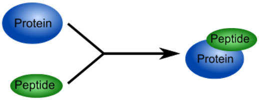




e.g., glycation; biotinylation; pegylation

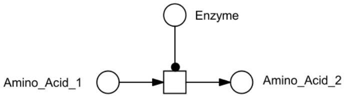
PTM: addition of other proteins or peptides

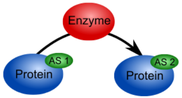




e.g., Covalent protein/peptide association; ISGylation; SUMOylation; Ubiquitination; Neddylation;

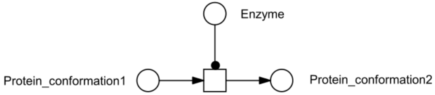
PTM: changing the chemical nature of amino acids

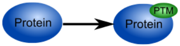




e.g.,
citrullination, or deimination, the conversion of arginine to citrulline;
deamidation, the conversion of glutamine to glutamic acid or asparagine to aspartic acid;
eliminylation, the conversion to an alkene by beta-elimination of phosphothreonine and phosphoserine, or dehydration of threonine and serine, as well as by decarboxylation of cysteine;
carbamylation, the conversion of lysine to homocitrulline;

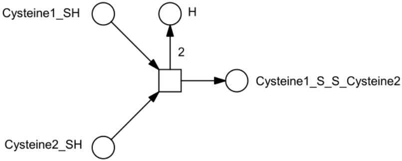
PTM: involving structural changes



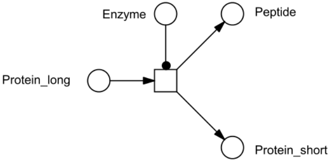


e.g.,
disulfide bridges, the covalent linkage of two cysteine amino acids;
proteolytic cleavage, cleavage of a protein at a peptide bond;
racemization of proline by prolyl isomerase

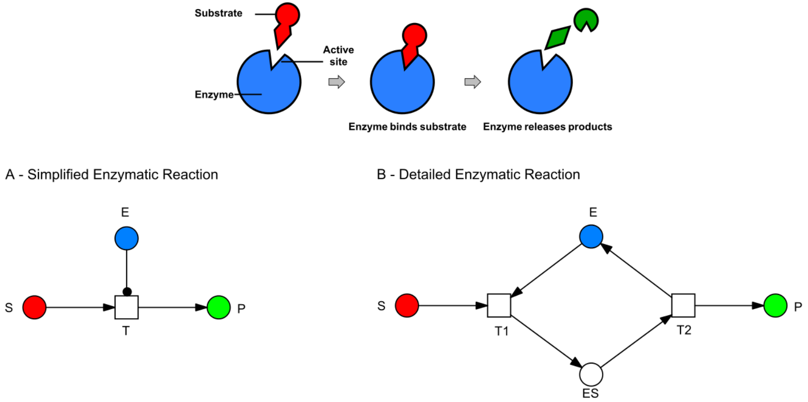
disulfid bridge formation



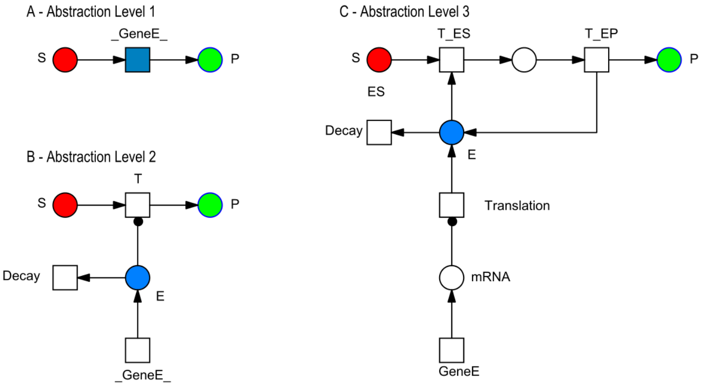
peptide cleavage



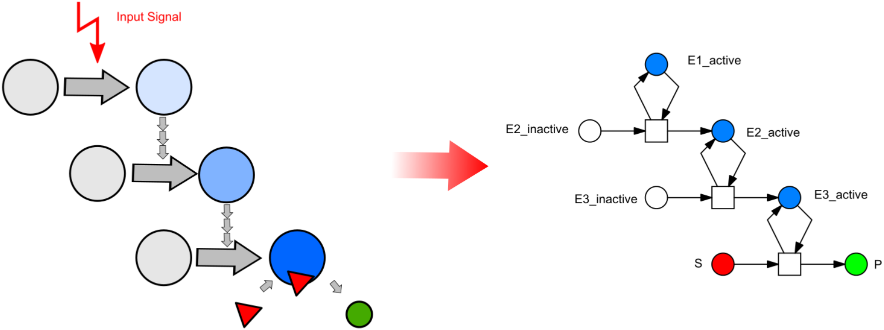
Example 3.1 (Enzymatic Reaction) *Here are two possibilities showing how to represent an enzymatic reaction using Petri nets. In **A**, the enzymatic reaction is simplified to one reaction. In **B**, we consider in addition the formation of an enzyme-substrate-complex. The enzymatic reaction is split into two steps.*



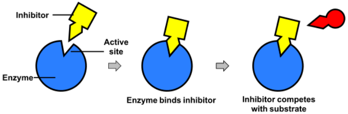
Example 3.2 (Enzymatic Reaction Coupled with Gene Expression) *The simple enzymatic reaction in* ***A*** *can be extended by adding more and more details about the gene expression, see* ***B*** *and* ***C****.*

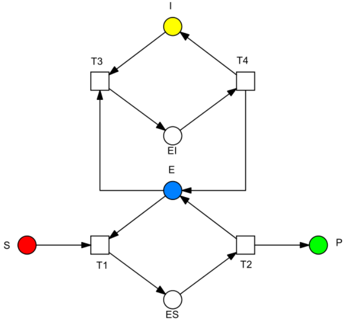


Example 3.4 (Signal Amplification) *In signal amplification multiple enzymes activate each other step by step. Signal amplification can be found in different signal pathway, e.g., in the mitogen-activated protein kinase (MAPK) cascade, where each enzyme can activate several enzymes in the next step of the signal pathway.*

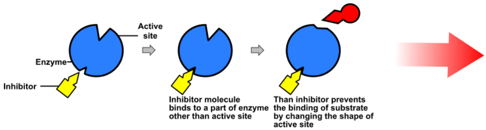


Example 3.5 (Competitive Enzyme Inhibition) *The substrate and the inhibitor can both bind to the active site of the enzyme. The inhibitor and substrate can not bind at the same time to the enzyme, they exclude each other.*





Example 3.6 (Allosteric Enzyme Inhibition) *The inhibitor binds to a distinct site at the enzyme. Thus, the inhibitor does not compete with the substrate and can inhibit the enzyme independently whether the substrate is bound or not.*



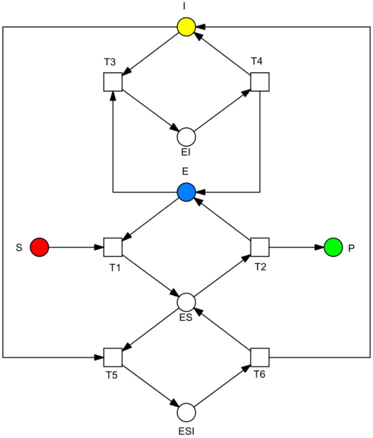


Table 4.2: *General Behavioural Properties of a Petri net and their biological Meaning*

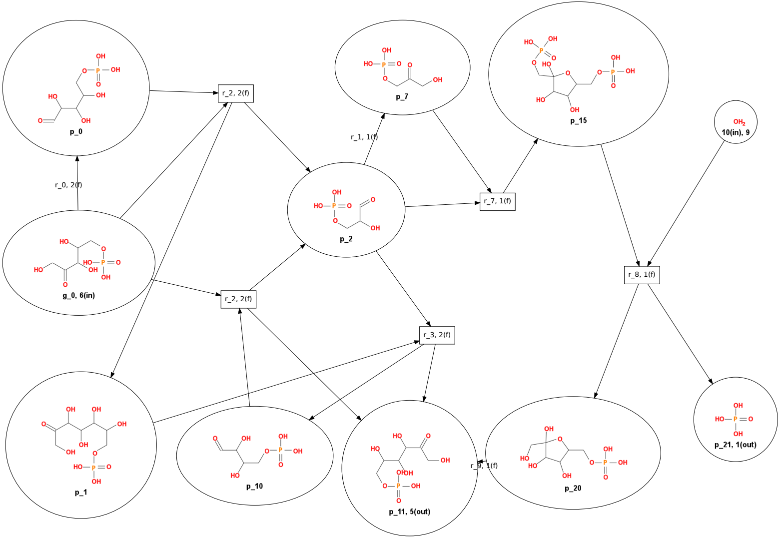
Property		**Informal Definition**	Biological Meaning
SB	Structurally bounded	A Petri is structurally bounded if it is bounded in any initial marking.	It is not possible that any component accumulates in the system independent of the initial conditions.
1-B	1-bounded	A Petri net is 1-bounded if all its places are 1-bounded.	Number of molecules or the concentration of every component is limited to one only.
k-B	k-bounded	A Petri net is k-bounded if all its places are k-bounded.	Number of molecules or the concentration level of each component is limited to a constant number k.
LIV	Liveness	Every transition of a Petri net contributes to the network behaviour forever.	All involved reaction will repeatedly occur and contribute to the time- (and spatial-) dependent development.

REV	Reversibility	The initial marking can be reached again from each reachable marking.	The initial state of a system can be reproduced by any possible state reached from the initial conditions.
DCF	Dynamically conflict free	A Petri net is has no dynamic conflicts if no state exists, in which two transitions are enabled, which could disable each other by firing.	The occurrence of a reaction inhibits another reaction which could also occur at the same time. The shared reactants are consumed by one of the reaction and no reactants are left or one reaction produces a component that directly inhibits the other reaction.
DSt	Dead states	A Petri net has a dead state if no transition can be enabled any more.	The system can run into a state, where no reaction can occur.
DTr	Dead transition	A transition in a Petri net is dead if it can not be enabled in any marking reachable from the initial marking.	The system can run from the initial state chosen initial state into at least one state, where at least one reaction can not occur any more.

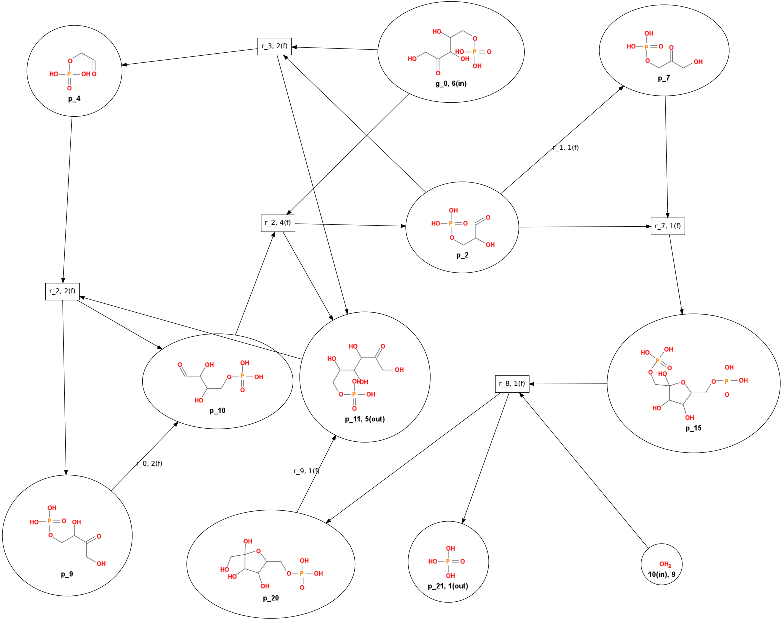
Table 4.4: *Behavioural Properties of a Petri net related to Traps and Siphons and their biological Meaning*

Property		**Informal Definition**	Biological Meaning
STP	Siphon trap property	Every siphon includes an initially marked trap. This excludes input places.	The part of the system that represents an outflow of certain components by a siphon contains also an initial active trap. Thus, the outflow does not stop, because it gets new input from the trap.
CPI	Covered by place invariants	A Petri net is covered by P-invariants if every place belongs to a P-invariant.	Mass Conservation is given in the entire system.
CTI	Covered by transition invariants	A Petri net is covered by T-invariants if every transition belongs to a T-invariant.	The initial state of all sequences of reactions can be restored.

A Self-Initiating Solution



A **NON**-Self-Initiating Solution

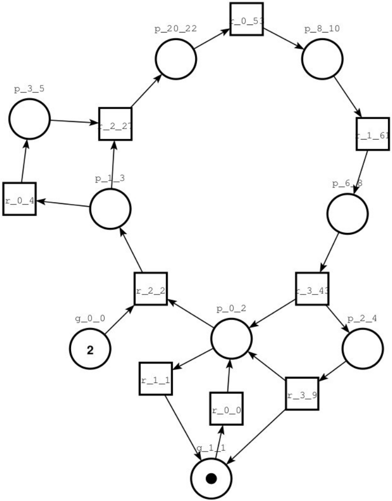


Complexity Questions

- When is the minimal number of tokens to make the goal marking reachable?
  $\rightarrow$ PSPACE-hard

Complexity Questions

- How to classify solutions of the ILP-approach for generative chemistries?



Complexity Questions (Esparaza article)

> Rule of thumb 1:
> All interesting questions about the behaviour of 1-safe Petri nets are PSPACE-hard.

- Is the Petri net live?
- Is some reachable marking a deadlock?
- Is a given marking reachable from the initial marking?
- Is there a reachable marking that puts a token in a given place?
- Is there a reachable marking that does not put a token in a given place?
- Is there a reachable marking that enables a given transition?
- Is there a reachable marking that enables more than one transition?
- Is the initial marking reachable from every reachable marking?
- Is there an infinite run?
- Is there exactly one run?
- Is there a run containing a given transition?
- Is there a run that does not contain a given transition?
- Is there a run containing a given transition infinitely often?
- Is there a run which enables a transition infinitely often but contains it only finitely often?

Complexity Questions

Rule of thumb 2:
Nearly all interesting questions about the behaviour of 1-safe Petri nets can be decided in polynomial space.

Complexity Questions

Rule of thumb 4:
Most interesting questions about the behaviour of *acyclic* 1-safe Petri nets are NP-hard.

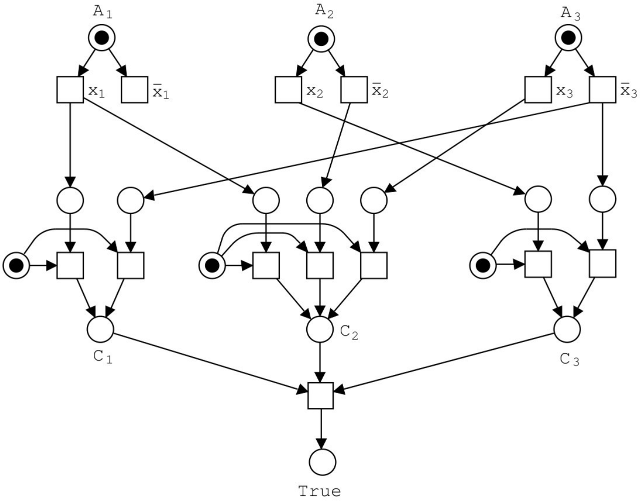
- Is a given marking reachable from the initial marking?
- Is there a reachable marking which marks a given place?
- Is there a reachable marking which does not mark a given place?
- Is there a reachable marking which enables a given transition?
- Is the initial marking reachable from every reachable marking?
- Is there a run containing a given transition?
- Is there a run that does not contain a given transition?

Complexity Questions

Is there a reachable marking which marks a given place?

Complexity Questions

Is there a reachable marking which marks a given place?



Acyclic net corresponding to the formula $(x_1 \lor \overline{x}_3) \land (x_1 \lor \overline{x}_2 \lor x_3) \land (x_2 \lor \overline{x}_3)$

Complexity Questions

Rule of thumb 5:
Many interesting questions about 1-safe conflict-free Petri nets are solvable in polynomial time.
Some interesting questions about *live* 1-safe free-choice Petri nets are solvable in polynomial time (and liveness of 1-safe free-choice Petri nets is decidable in polynomial time too).
Almost no interesting questions for 1-safe net classes substantially larger than free-choice Petri nets are solvable in polynomial time.

– Is there a reachable marking which marks a given place?
– Is there a reachable marking which does not mark a given place?
– Is there a reachable marking which enables a given transition?
– Is the initial marking reachable from every reachable marking?
– Is there a run that does not contain a given transition?

Complexity Questions

Rule of thumb 6:
All interesting questions about the behaviour of (Place/Transition) Petri nets are EXPSPACE-hard. More precisely, they require at least $2^{O(\sqrt{n})}$-space.