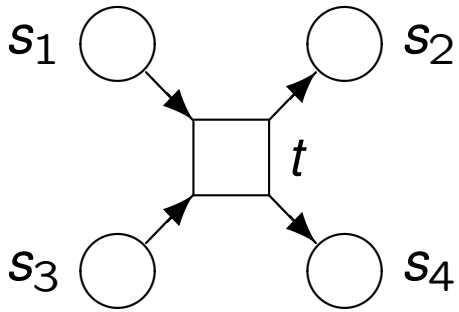
Petri nets

Petri nets

Petri nets are a basic model of parallel and distributed systems (named after Carl Adam Petri). The basic idea is to describe state changes in a system with transitions.



Petri nets contain places ○ and transitions □ that may be connected by directed arcs.

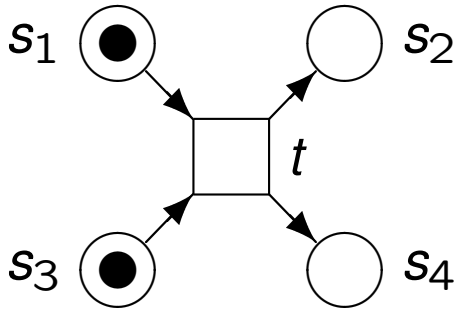
Places symbolise states, conditions, or resources that need to be met/be available before an action can be carried out.

Transitions symbolise actions.

Behaviour of Petri nets

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

A token on a place means that the corresponding condition is fulfilled or that a resource is available:



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.

Why Petri Nets?

low-level model for concurrent systems

expressly models concurrency, conflict, causality, ...

finite-state or infinite-state models

Content:

Semantics of Petri nets

Modelling with Petri nets

Analysis methods: finite/infinite-state case, structural analysis

Remark: Many variants of Petri nets exist in the literature; we regard a special simple case also called P/T nets.

Petri Net

A Petri 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,
- $W : ((P \times T) \cup (T \times P)) \to \mathbb{N}$ is the arc weight mapping (where $W(f) = 0$ for all $f \notin F$, and $W(f) > 0$ for all $f \in F$), and
- $m_0 : P \to \mathbb{N}$ is the initial marking representing the initial distribution of tokens.

Semantics

Let $N = \langle P, T, F, W, m_0 \rangle$ be a Petri net. We associate with it the transition system $\mathcal{M} = \langle S, \Sigma, \Delta, I, AP, \ell \rangle$, where:

$$S = \{\, m \mid m \colon P \to \mathbb{N} \,\}, \quad I = \{m_0\}$$

$$\Sigma = T$$

$$\Delta = \{\, (m, t, m') \mid \forall p \in P : m(p) \geq W(p,t) \wedge m'(p) = m(p) - W(p,t) + W(t,p) \,\}$$

$$AP = P, \quad \ell(m) = \{\, p \in P \mid m(p) > 0 \,\}$$

When $(m, t, m') \in \Delta$, we say that t is enabled in m and that its firing produces the successor marking m'; we also write $m \xrightarrow{t} m'$.

Semantics (remark)

The semantics given on the previous slide is also called interleaving semantics (one transition fires at a time).

Alternatively, one could define a step semantics, which better expresses the concurrent behaviours.

In step semantics, one allows a *multiset* of transitions to fire simultaneously; i.e. a multiset A is enabled in marking m if m contains enough tokens to fire all transitions in A.

However, for our purposes the interleaving semantics is sufficient.

Petri 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,

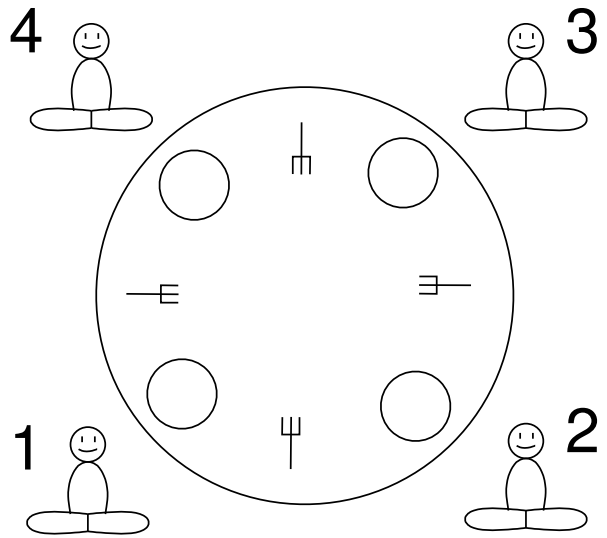
Let $a \in P \cup T$. The set ${}^\bullet a = \{a' \mid \langle a', a \rangle \in F\}$ is called the pre-set of a, and the set $a^\bullet = \{a' \mid \langle a, a' \rangle \in F\}$ is its post-set.

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.

Example: Dining philosophers

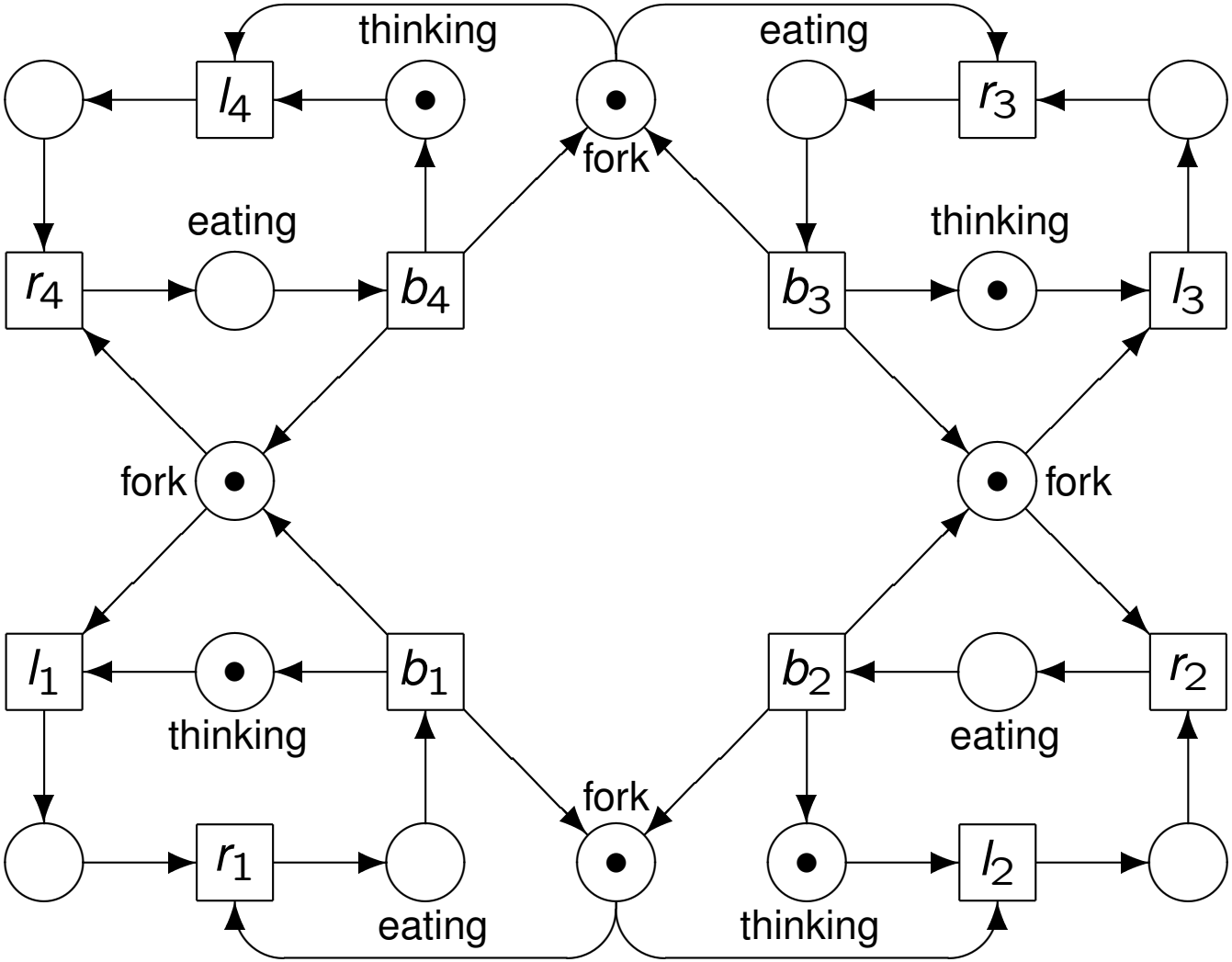
There are philosophers sitting around a round table.

There are forks on the table, one between each pair of philosophers.



The philosophers want to eat spaghetti from a large bowl in the center of the table.

Dining philosophers: Petri net



Synchronization by rendez-vous

Assume that we have a number of components with local actions and actions $!m$ (send message m) and $?m$ (receive message m).

Transition into Petri net:

Places = union of local states

Transitions:

– for local actions (p, a, p') build a Petri transition t labelled with a and ${}^\bullet t = \{p\}$, $t^\bullet = \{p'\}$;

– for pairs of actions $(p, !m, p')$ and $(q, ?m, q')$ build a Petri transition t labelled with m and ${}^\bullet t = \{p, q\}$, $t^\bullet = \{p', q'\}$.

Similar translations possible for other models discussed in the course (asynchronous product, TS with variables, . . .)

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. $m_0 = \{p_1, p_1, p_2, p_2, p_2, p_2, p_2\}$.

Reachable markings

Let m be a marking of a Petri 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 $m'' \in reach(m)$.

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

Let $N = \langle P, T, F, W, m_0 \rangle$ be a Petri net with associated transition system $\mathcal{M} = \langle S, \Sigma, \Delta, I, AP, \ell \rangle$.

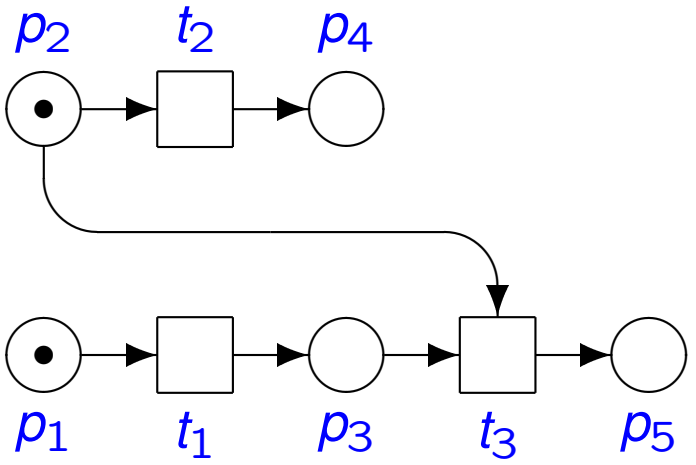
The reachability graph of N is the rooted, directed graph $G = \langle S', \Delta', m_0 \rangle$, where S' and Δ' are the restrictions of S and Δ to $reach(N)$.

The reachability graph can be constructed in iterative fashion, starting with the initial marking and then adding, step for step, all reachable markings.

k-safeness

Definition: Let N be a net. If no reachable marking of N can contain more than k tokens in any place (where $k \geq 0$ is some constant), then N is said to be k-safe.

Example: The following net is 1-safe.



Other example: the nets resulting from translating synchronous rendez-vous

k-safeness and Termination

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

In this case, there are finitely many reachable markings, and the construction of the reachability graph terminates.

On the other hand, if a net is not k-safe for any k, then there are infinitely many markings, and the construction will not terminate.

Reachability problem for 1-safe nets

Let N be a Petri net and m be a marking. The *reachability problem* for N, m is to determine whether $m \in reach(N)$.

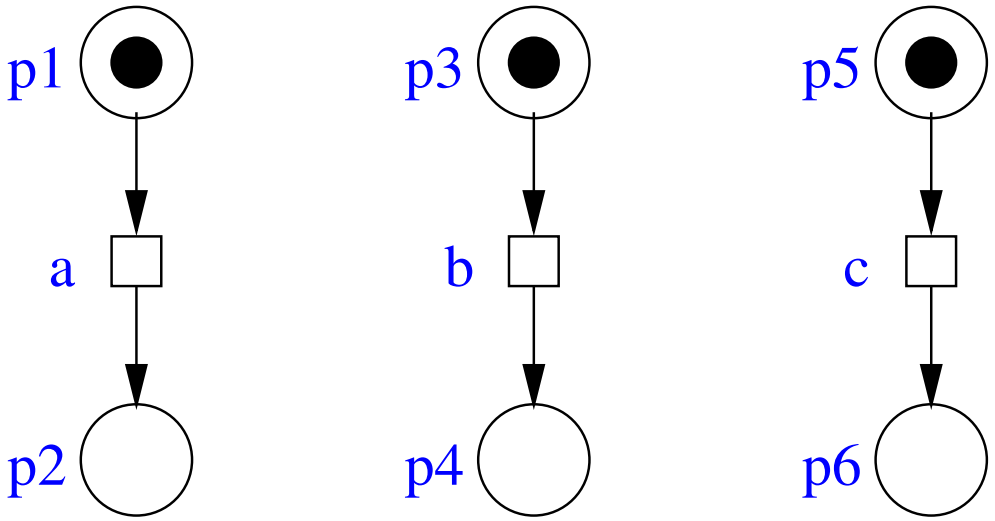
Theorem: The reachability problem for 1-safe Petri nets is PSPACE-complete.

Proof: (sketch)
upper bound: non-deterministically simulate net for at most $2^{|P|}$ steps;
hardness by reduction from QBF.

Corollary: Given a 1-safe net N and a place p, it is PSPACE-complete to determine whether $reach(N)$ contains a marking m such that $m(p) = 1$.

Algorithms for the reachability problem

The most straightforward way to solve the reachability problem on Petri nets is to construct the reachability graph. However, this can be very inefficient:



If there are n such components, then the reachability graph has size 2^n.

In fact, the reachability graph does not take advantage of the concurrent nature of a Petri net.

Later, we shall study a method more adapted to concurrent systems. It constructs a concise representation of the reachable markings called unfolding.

Given the unfolding of N, it is NP-complete to determine whether a given marking is reachable in N. One can thus take advantage of the advances made in SAT-checking.

First, we discuss a method to determine reachability information for non-safe nets.

Unbounded nets: Coverability graphs

Use of reachability graphs

If the net is not k-safe for any k, then it has infinitely many reachable markings, and one cannot effectively compute the reachability graph.

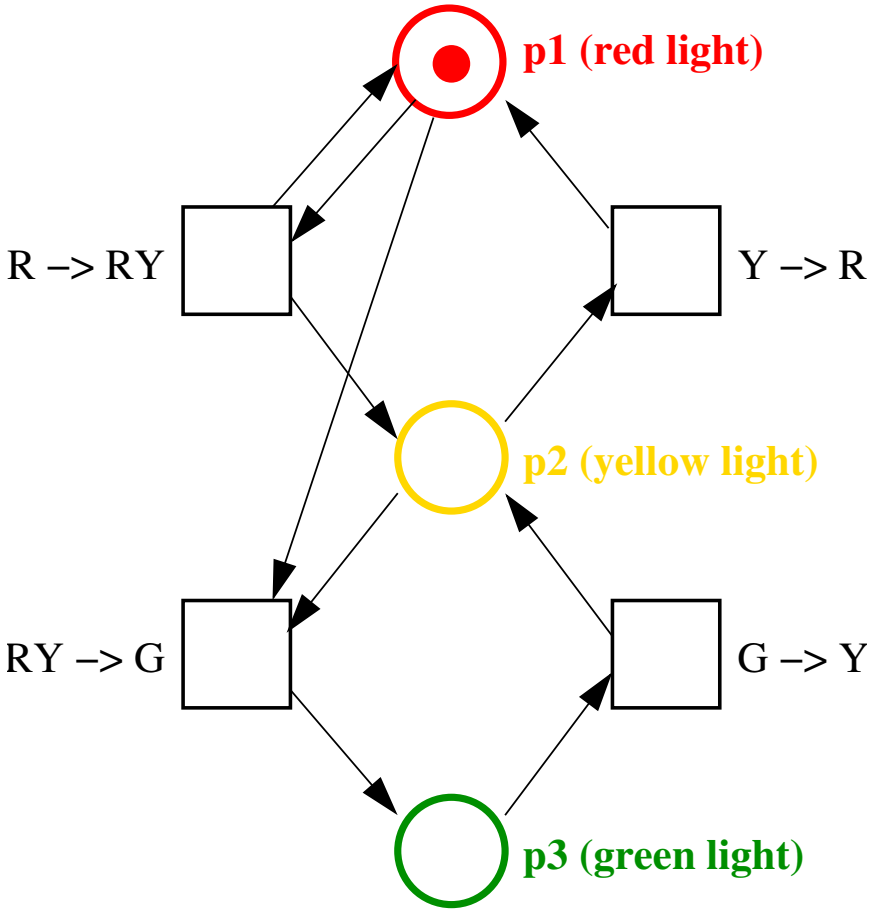
Nevertheless, the following problem is decidable: Given a (non-safe) net $\mathcal{P}$ and a marking m, is m reachable in $\mathcal{P}$?

This result is due to Mayr and Kosaraju (1981/82). However, the complexity of the problem is in general non-elementary, and no efficient methods are known.

Most of the time, though, one is interested in checking whether m is *part of* a reachable marking (one says that m is *coverable* in this case). This problem is somewhat easier to solve in practice, and we shall discuss it.

Example

Consider the following (slightly inept) attempt at modelling a traffic light:



Coverability Graphs

The reachability graph of the preceding net is infinite.

We will show the construction of a so-called coverability graph for it.

The coverability graph has the following properties:

- It can be used to find out whether the reachability graph is infinte.
- It is always finite, and its construction always terminates.
- Even for unbounded nets, it still gathers some information about reachable markings.

Computing with ω

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.

ω-Markings

We 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 (arbitrarily many).

Note: This is a technical definition that we will need for constructing the coverability graph! The nets that we use only have *finite* markings.

An ω-marking such as $(1, \omega, 0)$ can also be interpreted as the set of (non-ω)-markings that have one token on the first place, no token on the third place, and any number of tokens on the second place.

Firing Rule with 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

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

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

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

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 $M' \xrightarrow{t_1 t_2 \ldots t_n}$.

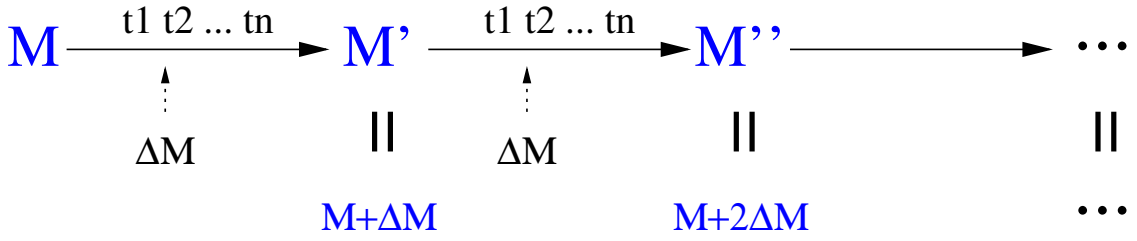
Coverability and Transition Sequences (2/2)

Let M, M' be markings such that $M < M'$, and assume that there is a sequence of transitions such that $M \xrightarrow{t_1 t_2 \ldots t_n} M'$ holds.

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

```
 1  ⟨V, E, v0⟩ := ⟨{M0}, ∅, M0⟩;
 2  Work : set := {M0};
 3  while Work ≠ ∅
 4  do select M from Work;
 5       Work := Work \ {M};
 6       for t ∈ enabled(M)
 7       do M' := fire(M, t);
 8            M' := AddOmegas(M, M', V);
 9            if M' ∉ V
10              then V := V ∪ {M'}
11                   Work := Work ∪ {M'};
12            E := E ∪ {⟨M, t, M'⟩};
13  return ⟨V, E, v0⟩;
```

The subroutine AddOmegas(M, M', V) will check if the sequences leading to M' can be repeated, strictly increasing the number of tokens on some places, and replace their values with ω.

Coverability Graph Algorithm (2/2)

The following notation us used in the AddOmegas subroutine:

- $M'' \rightarrow^* M$ iff the coverability graph currently contains a path (including the empty path!) leading from M'' to M.

ADDOMEGAS(M, M', V)

```
1  repeat saved := M';
2          for all M'' ∈ V s.t. M'' →* M
3          do if M'' < M'
4                then M' := M' + ((M' − M'') · ω);
5  until saved = M';
6  return M';
```

In other words, repeated check all the predecessor markings of the new marking M' to see if they are strictly covered by M'. Line 5 causes all places whose number of tokens in M' is strictly larger than in the "parent" M'' to contain ω.

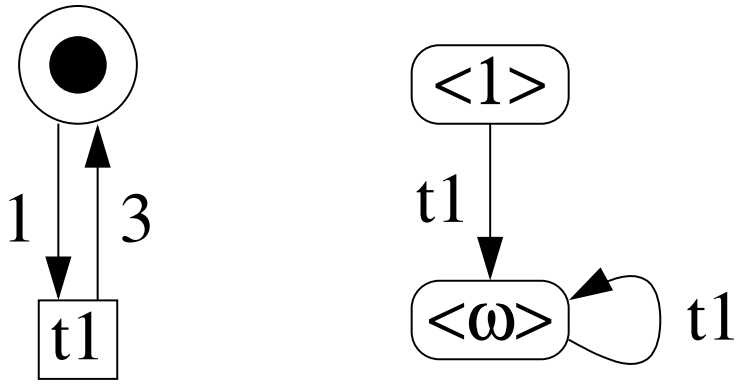
Properties of the coverability graph (1)

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

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.

Note that 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:



Properties of the coverability graph (2)

The coverability graph could thus be said to compute an overapproximation of the reachable markings.

The construction of the coverability graph always terminates.
If N is bounded, then the coverabilibility graph is identical to the reachability graph.

Coverability graphs are not unique,
i.e. for a given net there may be more than one coverability graph, depending on the order of the worklist and the order in which firing transitions are considered.