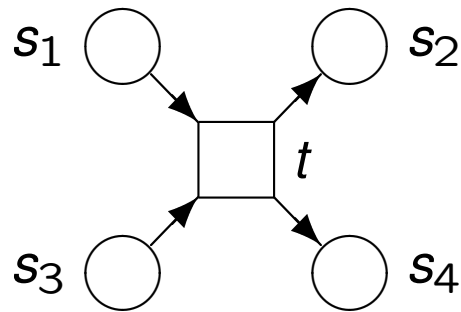




Petri nets

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.

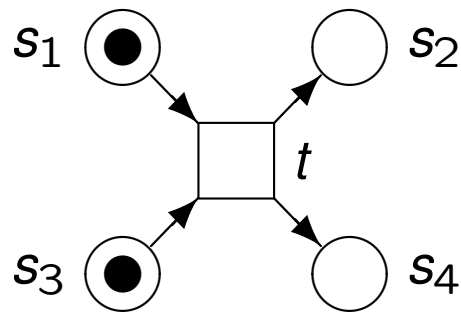
Places symbolise **states**, **conditions**, or **resources** that need to be met/be available before an action can be carried out.

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 .

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**,
- $W: ((P \times T) \cup (T \times P)) \rightarrow \mathbb{N}$ is the **arc weight** mapping (where $W(f) = 0$ for all $f \notin F$, and $W(f) > 0$ for all $f \in F$), and
- $M_0: P \rightarrow \mathbb{N}$ is the **initial marking** representing the initial distribution of tokens.

P/T nets: Remarks

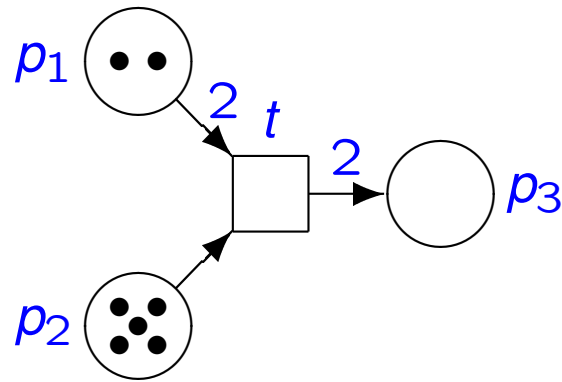
If $\langle p, t \rangle \in F$ for a transition t and a place p , then p is an **input place** of t ,

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.

Place/Transition Net: Example



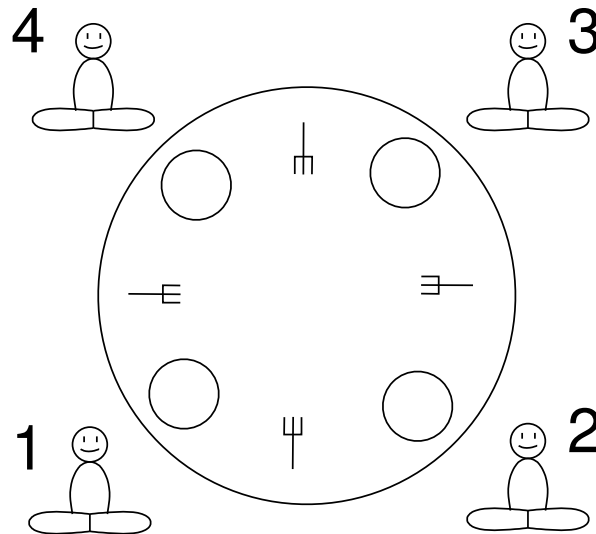
The place/transition net $\langle P, T, F, W, M_0 \rangle$ above is defined as follows:

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

Bigger example: Dining philosophers

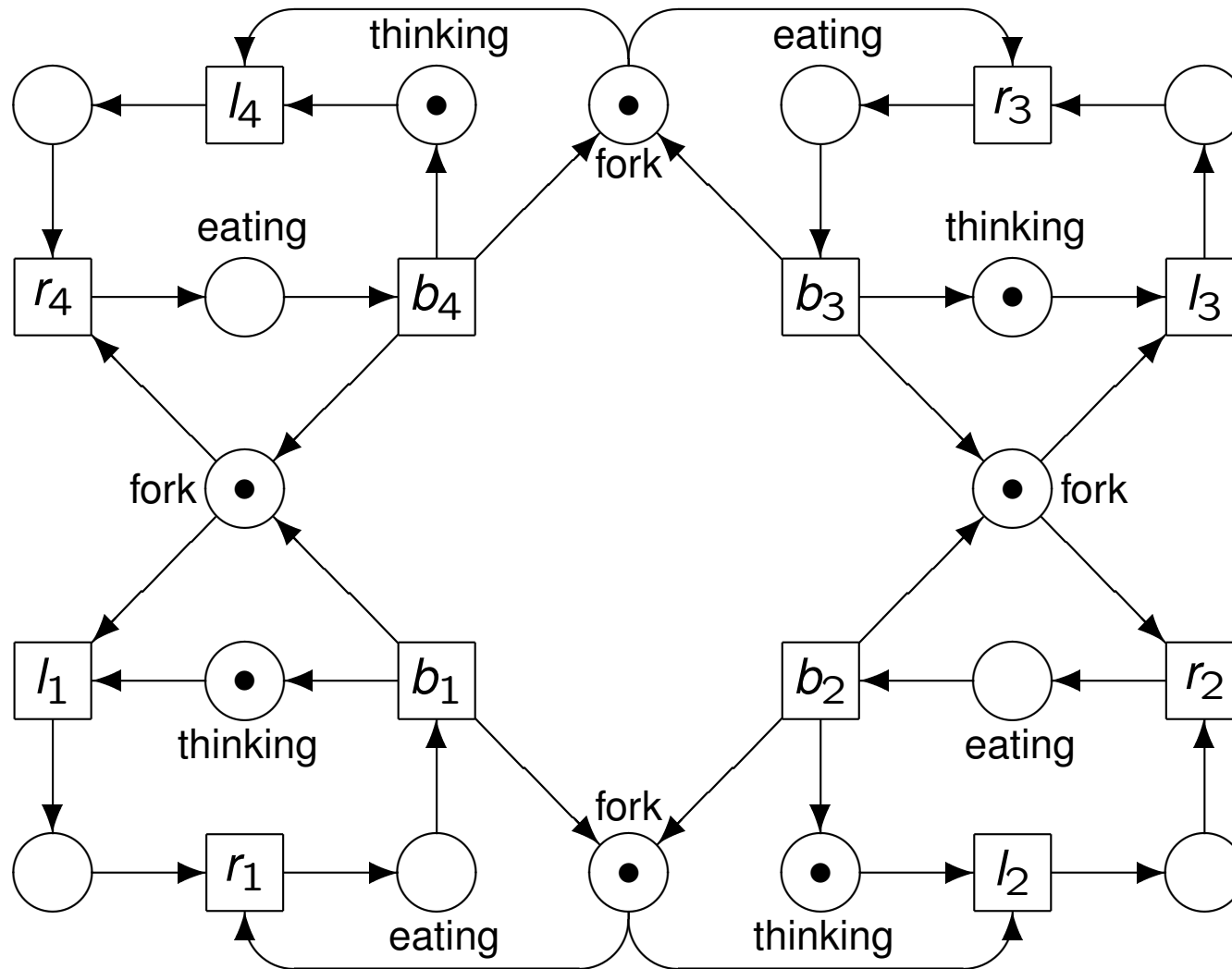
There are philosophers sitting around a round table.

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



Literature about Petri Nets

For more in-depth coverage, a lot of literature about Petri nets is available, for instance:

Reisig, *Elements of Distributed Algorithms: Modelling and Analysis with Petri Nets*, Springer, 1998

Tools: The PEP tool

<http://theoretica.informatik.uni-oldenburg.de/~pep/>

Internet resources: Petri Nets World

<http://www.informatik.uni-hamburg.de/TGI/PetriNets/>

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

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

Firing condition:

Transition $t \in T$ is **M -enabled** (or: **enabled in M**), written $M \xrightarrow{t}$, iff

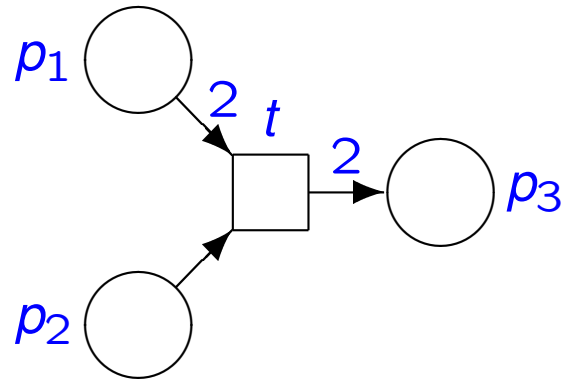
$$\forall p \in \bullet t : M(p) \geq W(p, t).$$

Firing rule:

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

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

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:

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

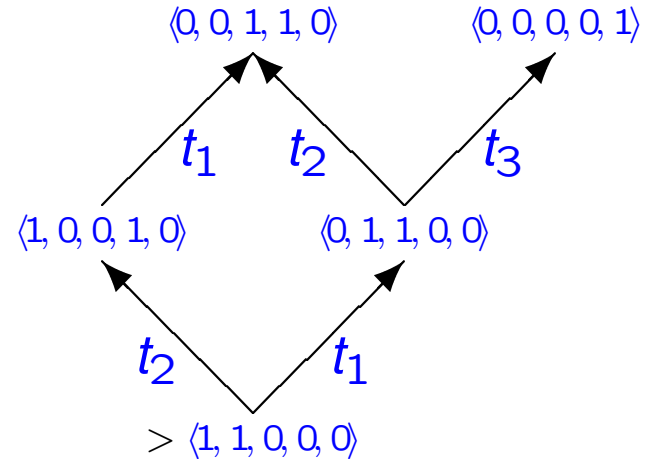
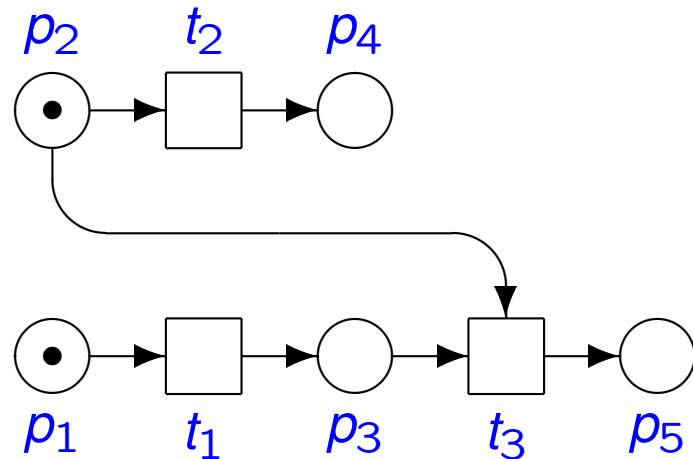
The set of reachable markings $reach(N)$ of a net $N = \langle P, T, F, W, M_0 \rangle$ is defined to be $reach(M_0)$.

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

- $V = \text{reach}(N)$ 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;
- $E = \{ \langle M, t, M' \rangle \mid M \in V \text{ and } M \xrightarrow{t} M' \}$ 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 .

Computing the reachability graph

```
REACHABILITY-GRAPH( $\langle P, T, F, W, M_0 \rangle$ )
1   $\langle V, E, v_0 \rangle := \langle \{M_0\}, \emptyset, M_0 \rangle;$ 
2   $Work : set := \{M_0\};$ 
3  while  $Work \neq \emptyset$ 
4  do select  $M$  from  $Work$ ;
5      $Work := Work \setminus \{M\};$ 
6     for  $t \in \text{enabled}(M)$ 
7     do  $M' := \text{fire}(M, t);$ 
8         if  $M' \notin V$ 
9         then  $V := V \cup \{M'\}$ 
10             $Work := Work \cup \{M'\};$ 
11             $E := E \cup \{\langle M, t, M' \rangle\};$ 
12 return  $\langle V, E, v_0 \rangle;$ 
```

The algorithm makes use of two functions:

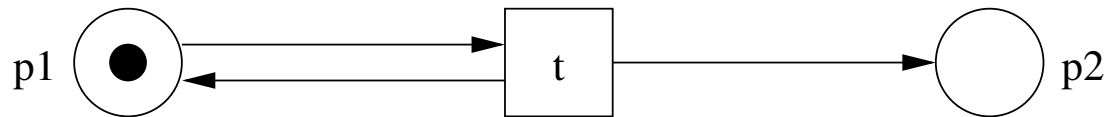
- $\text{enabled}(M) := \{t \mid M \xrightarrow{t}\}$
- $\text{fire}(M, t) := M'$
if $M \xrightarrow{t} M'$

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.

Reachability Graph: Termination

In general, the algorithm may not terminate!

This is because the graph may be infinite. Example: Example:



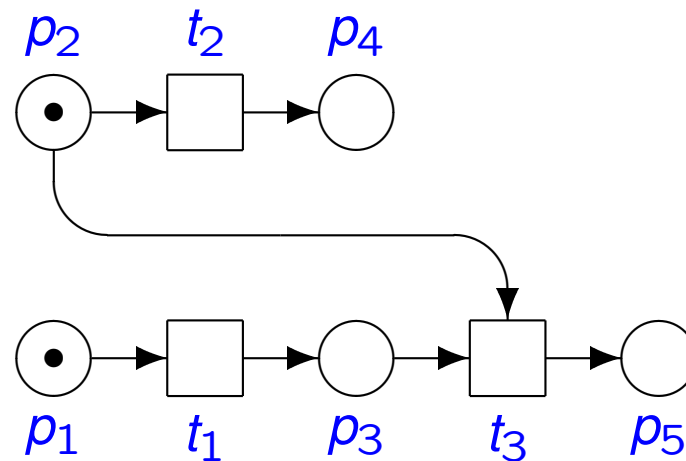
The reachable markings in this net are $\langle 1, 0 \rangle$, $\langle 1, 1 \rangle$, $\langle 1, 2 \rangle$, ...

It is quite straightforward to see that the graph is finite if and only if we can put a *bound* on the number of tokens in 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 (1): The following net is 1-safe.



Example (2): The net from the previous slide is not k -safe for any k .

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.

Use of reachability graphs

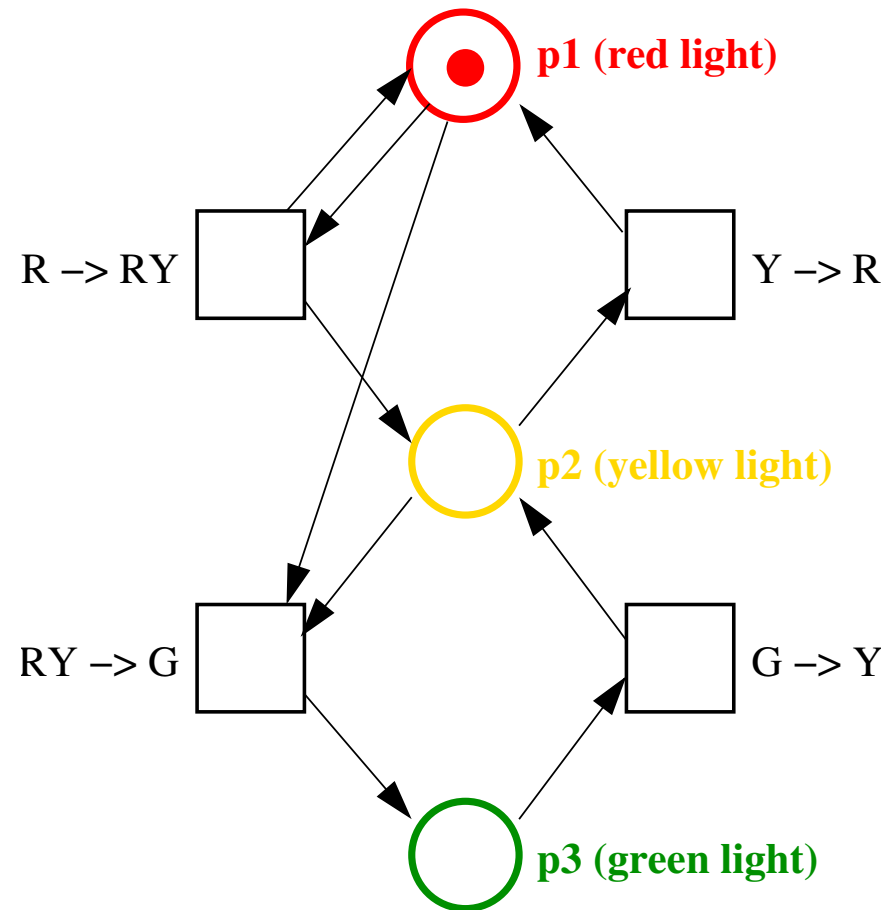
In practice, all analysis tools and methods for Petri nets compute (some representation of) the reachability graph. The reachability graph can be effectively computed if the net is k -safe for some k .

If the net is not k -safe for any k , we may compute the **coverability graph** instead (see upcoming slides).

Coverability graphs

Example

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



Coverability Graphs

The reachability graph of the preceding net is infinite. As we have mentioned before, the algorithm for computing the reachability graph will not terminate in this case.

We will show the construction of a different graph:
the so-called **coverability graph**.

The coverability graph has the following properties:

- It can be used to find out whether the reachability graph is infinite.

- It is always finite, and its construction always terminates.

- It gathers some information about reachable markings.

- However, it is slightly more complicated than the reachability graph!

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 $n \in \mathbb{N}$:

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

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

$$\omega - n = \omega,$$

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

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

$$n \leq \omega, \text{ and } \omega \leq \omega.$$

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

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

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 $M \xrightarrow{t_1 t_2 \dots t_n} M'$ to denote:

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

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

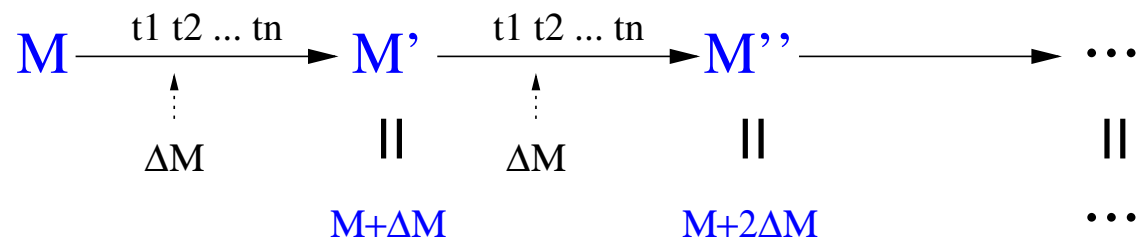
Coverability and Transition Sequences (2/2)

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

Thus, there is a marking M'' with $M' \xrightarrow{t_1 t_2 \dots t_n} M''$.

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 $t_1 t_2 \dots t_n$ repeatedly we can “pump” an arbitrary number of tokens to all the places having a non-zero marking in ΔM .

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 ΔM are replaced by ω .

Coverability Graph Algorithm (1/2)

```
COVERABILITY-GRAPH( $\langle P, T, F, W, M_0 \rangle$ )
1   $\langle V, E, v_0 \rangle := \langle \{M_0\}, \emptyset, M_0 \rangle$ ;
2   $Work : set := \{M_0\}$ ;
3  while  $Work \neq \emptyset$ 
4  do select  $M$  from  $Work$ ;
5      $Work := Work \setminus \{M\}$ ;
6     for  $t \in enabled(M)$ 
7     do  $M' := fire(M, t)$ ;
8          $M' := AddOmegas(M, t, M', V, E)$ ;
9         if  $M' \notin V$ 
10        then  $V := V \cup \{M'\}$ 
11             $Work := Work \cup \{M'\}$ ;
12         $E := E \cup \langle M, t, M' \rangle$ ;
13 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 $AddOmegas(M, t, M', V, E)$, 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 notation is 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, t, M', V, E)

```
1  repeat  $saved := M'$ ;  
2      for all  $M'' \in V$  s.t.  $M'' \rightarrow^* M$   
3      do if  $M'' < M'$   
4          then  $M' := M' + ((M' - M'') \cdot \omega)$ ;  
5  until  $saved = M'$ ;  
6  return  $M'$ ;
```

In other words, repeatedly 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 .

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:

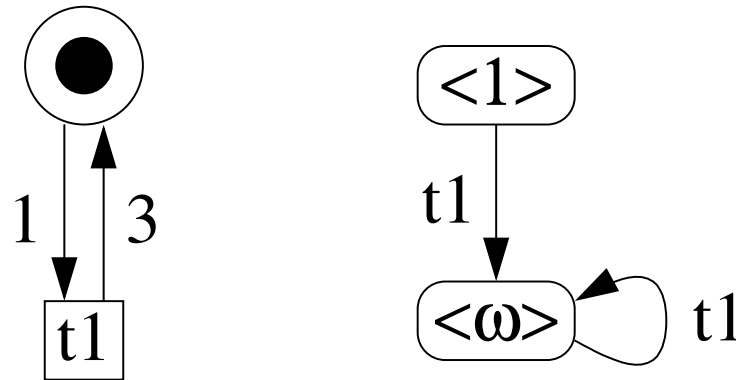
A marking M of N is reachable *if and only if* M is a vertex of the reachability graph of N .

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 construction of the **reachability graph** may not terminate.
It terminates if and only if N is bounded.

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

Reachability and coverability graphs: Comparison (3)

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).

Reachability and coverability graphs: Comparison (4)

Reachability graphs are **unique**,

i.e. for a given net there is exactly one reachability graph (modulo isomorphism).

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.