Petri Nets
Lecture Notes for SS 2015

Prof. Javier Esparza

July 16, 2015

Contents

Sources

The main sources are:

J. Desel. **Struktur und Analyse von Free-Choice-Petrinetzen**. Deutscher Universitäts Verlag, 1992.

J. Desel und J. Esparza. **Free-choice Petri nets**. Cambridge Tracts in Theoretical Computer Science 40, Cambridge University Press, 1995.

The Petri net model of Peterson's algorithm is taken from

E. Best. **Semantics of Sequential and Parallel Programs**. Prentice-Hall, 1996.

The action-reaction protocol is taken from

R. Walter. **Petrinetzmodelle verteilter Algorithmen – Intuition und Beweistechnik**. Dieter Bertz Verlag, 1996.

The train examples of Chapter 2 belong to the Petri net folklore. They were first introduced by H. Genrich.

Part I

Petri Nets: Syntax, Semantics, Models

Chapter 1

Basic definitions

1.1 Preliminaries

Numbers

$\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ denote the natural, rational, and real numbers.

Relations

Let X be a set and $R \subseteq X \times X$ a relation. R^* denotes the *transitive and reflexive closure* of R. R^{-1} is the *inverse* of R, that is, the relation defined by $(x, y) \in R^{-1} \Leftrightarrow (y, x) \in R$.

Sequences

A *finite sequence* over a set A is a mapping $\sigma\colon \{1, \ldots, n\} \to A$, denoted by the string $a_1 a_2 \ldots a_n$, where $a_i = \sigma(i)$ for every $1 \leq i \leq n$, or the mapping $\epsilon\colon \emptyset \to A$, the *empty sequence*. The *length* of σ is n and the length of ϵ is 0.

An *infinite sequence* is a mapping $\sigma : \mathbb{N} \to A$. We write $\sigma = a_1 a_2 a_3 \ldots$ with $a_i = \sigma(i)$.

The *concatenation* of two finite sequences or of a finite and an infinite sequence is defined as usual. Given a finite sequence σ, we denote by σ^ω the *infinite concatenation* $\sigma\sigma\sigma \ldots$.

σ is a *prefix* of τ if $\sigma = \tau$ or $\sigma\sigma' = \tau$ for some sequence σ'.

The *alphabet* of a sequence σ is the set of elements of A occurring in σ. Given a sequence σ over A and $B \subseteq A$, the *projection* or *restriction* $\sigma|_B$ is the result of removing all occurrences of elements $a \in A \setminus B$ in σ.

Vectors and matrices

Let $A = \{a_1, \ldots, a_n\}$ be a finite set and let K be one of $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$.

We represent a mapping $X\colon A \to K$ by the vector $(X(a_1), \ldots, X(a_n))$. We identify the mapping X and its vector representation.

Given vectors $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$, the *(scalar) product* $X \cdot Y$ is the number $x_1y_1 + \ldots + x_ny_n$ (we do not distinguish between row and column vectors!). We write $X \geq Y$ to denote $x_1 \geq y_1 \wedge \ldots \wedge x_n \geq y_n$,a nd $X > Y$ to denote $x_1 > y_1 \wedge \ldots \wedge x_n > y_n$.

Let $B = \{b_1, \ldots, b_m\}$ be a finite set. A mapping $C\colon A \times B \to K$ is represented by the $n \times m$ matrix

$$\begin{pmatrix} C(a_1,b_1) & C(a_1,b_2) & \cdots & C(a_1,b_m) \\ C(a_2,b_1) & C(a_2,b_2) & \cdots & C(a_2,b_m) \\ \cdots & \cdots & \cdots & \cdots \\ C(a_n,b_1) & C(a_n,b_2) & \cdots & C(a_n,b_m) \end{pmatrix}$$

We also write $C = (c_{ij})_{i=1,\ldots,n,j=1,\ldots,m}$, where $c_{ij} = C(a_i, b_j)$.

Let $X = (x_1, \ldots, x_m)$ be a vector and let C be a $n \times m$ matrix. The *product* $C \cdot X$ is the vector $Y = (y_1, \ldots, y_n)$ given by

$$y(i) = c_{i1}x_1 + \ldots + c_{im}x_m$$

and for $X = (x_1, \ldots, x_n)$ the product $X \cdot C$ is the vector $Y = (y_1, \ldots, y_m)$ given by

$$y(i) = c_{1i}x_1 + \ldots + c_{ni}x_n$$

1.2 Syntax

Definition 1.2.1 (Net, preset, postset)
A *net* $N = (S, T, F)$ consists of a finite set S of *places* (represented by circles), a finite set T of *transitions* disjoint from S (squares), and a *flow relation* (arrows) $F \subseteq (S \times T) \cup (T \times S)$.

The places and transitions of N are called *elements* or *nodes*. The elements of F are called *arcs*.

Given $x \in S \cup T$, the set ${}^\bullet x = \{y \mid (y, x) \in F\}$ is the *preset* of x and $x^\bullet = \{y \mid (x, y) \in F\}$ is the *postset* of x. For $X \subseteq S \cup T$ we denote ${}^\bullet X = \bigcup_{x \in X} {}^\bullet x$ and $X^\bullet = \bigcup_{x \in X} x^\bullet$.

Example. Let $N = (S, T, F)$ be the net

$$\begin{aligned} S &= \{s_1, \ldots, s_6\} \\ T &= \{t_1, \ldots, t_4\} \\ F &= \{(s_1,t_1), (t_1,s_2), (s_2,t_2), (t_2,s_1), \\ &\quad (s_3,t_2), (t_2,s_4), (s_4,t_3), (t_3,s_3), \\ &\quad (s_5,t_3), (t_3,s_6), (s_6,t_4), (t_4,s_5)\} \end{aligned}$$

Figure 1.1 shows the graphical representation of N. For example we have ${}^\bullet t_2 =$

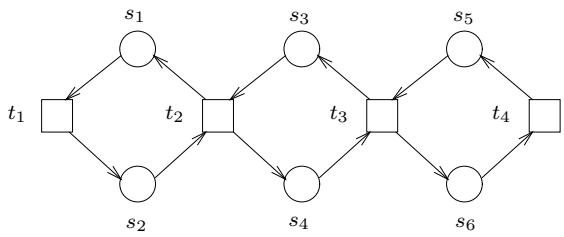


Figure 1.1: Graphical representation of the net N

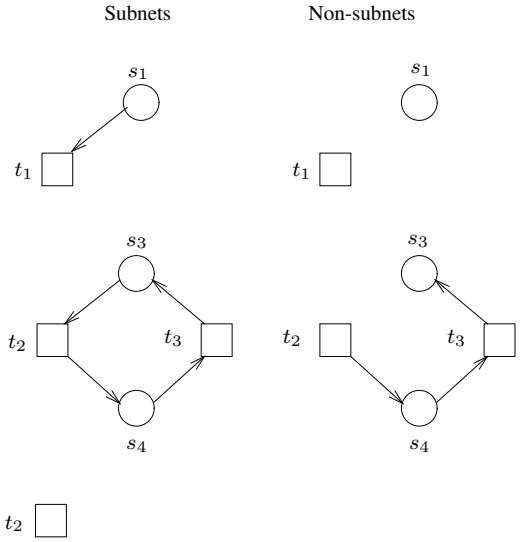


Figure 1.2: Subnets and non-subnets of the net of Figure 1.1

$\{s_2, s_3\}$ and ${}^\bullet S = S^\bullet = T$.

Remark: Nets with empty S, T or F are allowed!

Definition 1.2.2 (Subnet)
$N' = (S', T', F')$ is a *subnet* of $N = (S, T, F)$ if

- $S' \subseteq S$,
- $T' \subseteq T$, and
- $F' = F \cap ((S' \times T') \cup (T' \times S'))$ (not $F' \subseteq F \cap ((S' \times T') \cup (T' \times S'))$!).

Figure 1.2 shows some subnets and non-subnets of the net of Figure 1.1.

Definition 1.2.3 (Path, circuit)
A path of a net $N = (S, T, F)$ is a finite, nonempty sequence $x_1 \dots x_n$ of nodes of N

such that $(x_1, x_2), \ldots, (x_{n-1}, x_n) \in F$. We say that a path $x_1 \ldots x_n$ *leads from* x_1 *to* x_n.

A path is a *circuit* if $(x_n, x_1) \in F$ and $(x_i = x_j) \Rightarrow i = j$ for every $1 \leq i, j \leq n$.

N is *connected* if $(x, y) \in (F \cup F^{-1})^*$ for every $x, y \in S \cup T$, and *strongly connected* if $(x, y) \in F^*$ for every $x, y \in S \cup T$.

Remarks:

- Every net with 0 or 1 node is strongly connected!
- If N is strongly connected then it is also connected.

Proposition 1.2.4 *Let $N = (S, T, F)$ be a net.*

(1) N is connected iff there are no two subnets (S_1, T_1, F_1) and (S_2, T_2, F_2) of N such that

- *$S_1 \cup T_1 \neq \emptyset$, $S_2 \cup T_2 \neq \emptyset$;*
- *$S_1 \cup S_2 = S$, $T_1 \cup T_2 = T$, $F_1 \cup F_2 = F$;*
- *$S_1 \cap S_2 = \emptyset$, $T_1 \cap T_2 = \emptyset$.*

(2) A connected net is strongly connected iff for every $(x, y) \in F$ there is a path leading from y to x.

Proof. Exercise. □

1.3 Semantics

Definition 1.3.1 (Markings)
Let $N = (S, T, F)$ be a net. A *marking* of N is a mapping $M : S \to \mathbb{N}$. Given $R \subseteq S$ we write $M(R) = \sum_{s \in R} M(s)$. A place s is *marked* at M if $M(s) > 0$. A set of places R is *marked* at M if $M(R) > 0$, that is, if at least one place of R is marked at M.

Instead of mappings $S \to \mathbb{N}$ sometimes we use vectors. For this we fix a total order on the places of N. With this convention we can represent a marking $M : S \to \mathbb{N}$ as a vector of dimension $|S|$.

Markings are graphically represented by drawing black dots ("tokens") on the places.

Definition 1.3.2 (Firing rule, dead markings)
A transition is *enabled* at a marking M if $M(s) \geq 1$ for every place $s \in {}^\bullet t$. If t is enabled, then it can *occur* or *fire*, leading from M to the marking M' (denoted $M \xrightarrow{t} M'$) given by:

$$M'(s) = \begin{cases} M(s) - 1 & \text{if } s \in {}^\bullet t \setminus t^\bullet \\ M(s) + 1 & \text{if } s \in t^\bullet \setminus {}^\bullet t \\ M(s) & \text{otherwise} \end{cases}$$

A marking is *dead* if it does not enable any transition.

Example 1.3.3 *Let M be the marking of the net N in Figure 1.1 given by $M(s_1) = M(s_4) = M(s_5) = 1$ and $M(s_2) = M(s_3) = M(s_6) = 0$. We denote this marking by the vector $(1, 0, 0, 1, 1, 0)$.*

The marking enables transitions t_1 and t_3, because ${}^\bullet t_1 = \{s_1\}$ and ${}^\bullet t_3 = \{s_4, s_5\}$. Transition t_2 is not enabled, because $M(s_2) = 0$. Transition t_4 is not enabled, because $M(s_6) = 0$. We have

$$\begin{array}{lcl} (1,0,0,1,1,0) & \xrightarrow{t_1} & (0,1,0,1,1,0) \\ (1,0,0,1,1,0) & \xrightarrow{t_3} & (1,0,1,0,0,1) \end{array}$$

Definition 1.3.4 (Firing sequence, reachable marking)
Let $N = (S, T, F)$ be a net and let M be a marking of N. A finite sequence $\sigma = t_1 \dots t_n$ is *enabled at a marking M* if there are markings $M_1, M_2, \dots, M_n$ such that $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} \dots \xrightarrow{t_n} M_n$. We write $M \xrightarrow{\sigma} M_n$. The empty sequence ϵ is enabled at any marking and we have $M \xrightarrow{\epsilon} M$.

If $M \xrightarrow{\sigma} M'$ for some markings M, M' and some sequence σ, then we write $M \xrightarrow{*} M'$ and say that *M' is reachable from M*. $[M\rangle$ denotes the set of markings that are reachable from M.

An infinite sequence $\sigma = t_1 t_2 \dots$ is *enabled at a marking* if there are markings $M_1, M_2, \dots$ such that $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \longrightarrow \dots$

Example 1.3.5 *Let N be the net of Figure 1.1 and let $M = (1, 0, 0, 1, 1, 0)$ be a marking of N. We have*

$$(1,0,0,1,1,0) \xrightarrow{t_1} (0,1,0,1,1,0) \xrightarrow{t_3} \begin{array}{l} (0,1,1,0,0,1) \\ \qquad \downarrow t_2 \\ (1,0,0,1,0,1) \end{array} \xrightarrow{t_4} (1,0,0,1,1,0)$$

So M enables the finite sequence $t_1\ t_3\ t_2\ t_4$ and the infinite sequence $(t_1\ t_3\ t_2\ t_4)^\omega$.

Proposition 1.3.6 *A (finite or infinite) sequence σ is enabled at M iff every finite prefix of σ is enabled at M.*

Proof. Easy exercise. □

The following simple lemma plays a fundamental role in many results about Petri nets.

Lemma 1.3.7 *[Monotonicity lemma]*
Let M and L be two markings of a net.

(1) If $M \xrightarrow{\sigma} M'$ for a finite sequence σ, then $(M + L) \xrightarrow{\sigma} (M' + L)$ for every marking L.

(2) If $M \xrightarrow{\sigma}$ for an infinite sequence σ, then $(M + L) \xrightarrow{\sigma}$ for every marking L.

Proof. (1): by induction on the length of σ.
Basis: $\sigma = \epsilon$. ϵ is enabled at any marking.
Step: Let $\sigma = \tau t$ (t transition) such that $M \xrightarrow{\tau} M'' \xrightarrow{t} M'$. By induction hypothesis $(M + L) \xrightarrow{\tau} (M'' + L)$. From the firing rule and $M'' \xrightarrow{t} M'$ we get $(M'' + L) \xrightarrow{t} (M' + L)$. So $(M + L) \xrightarrow{\tau t} (M' + L)$.

(2): We show that every finite prefix of σ is enabled at $M + L$. The result then follows from Proposition 1.3.6. By Proposition 1.3.6, every finite prefix of σ is enabled at M. That is, for every finite prefix τ of σ there is a marking M' such that $M \xrightarrow{\tau} M'$. By (1) we get $(M + L) \xrightarrow{\tau} (M' + L)$, and we are done. □

Definition 1.3.8 (Petri nets)
A *Petri net*, *net system*, or just a *system* is a pair (N, M_0) where N is a connected net $N = (S, T, F)$ with nonempty sets of places and transitions, and an *initial marking* $M_0 \colon S \to \mathbb{N}$. A marking M is *reachable in* (N, M_0) or *a reachable marking of* (N, M_0) if $M_0 \xrightarrow{*} M$.

Definition 1.3.9 (Reachability graph)
The *reachability graph* G of a Petri net (N, M_0) where $N = (S, T, F)$ is the directed, labeled graph satisfying:

- The nodes of G are the reachable markings of (N, M_0).
- The edges of G are labeled with transitions from T.
- There is an edge from M to M' labeled by t iff $M \xrightarrow{t} M$, that is, iff M enables t and the firing of t leads from M to M'.

The algorithm of Figure 1.3 computes the reachability graph. It uses two functions:

- $\mathsf{enabled}(M)$: returns the set of transitions enabled at M.
- $\mathsf{fire}(M, t)$: returns the marking M' such that $M \xrightarrow{t} M'$.

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((S, T, F, M_0))
 1  (V, E, v_0) := ({M_0}, ∅, M_0);
 2  Work : set := {M_0};
 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        if M' ∉ V
 9          then V := V ∪ {M'}
10               Work := Work ∪ {M'};
11        E := E ∪ {(M, t, M')};
12  return (V, E, v_0)
```

Figure 1.3: Algorithm for computing the reachability graph

Chapter 2

Modelling with Petri nets

2.1 A buffer of capacity n

We model a buffer with capacity for n items. Figure 2.1 shows the Petri net for $n = 3$. The model consists of n cells, each of them with capacity for one item. The addition

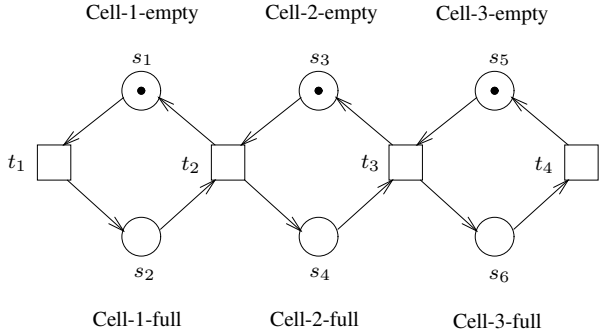


Figure 2.1: A 3-buffer

of a new item is modeled by the firing of t_1. The firing of transition t_i models moving the item in cell $i-1$ to cell i. Firing t_{n+1} models removing one item. Observe that the buffer is concurrent: there are reachable markings at which transitions t_1 and t_{n+1} can occur independently of each other, that is, an item can be added while another one is being removed.

Figure 2.2 shows the reachability graph of the buffer with capacity 3. By inspection of the reachability graph we can see that the following properties hold:

- Consistency: no cell is simultaneously empty and full (that is, no marking puts tokens on s_i and s_{i+1} for $i = 1, 2, 3$).
- 1-boundedness: every reachable marking puts at most one token in a given place.
- Deadlock freedom: every reachable marking has at least one successor marking. Even more: every cell can always be filled and emptied again (every transition can occur again).

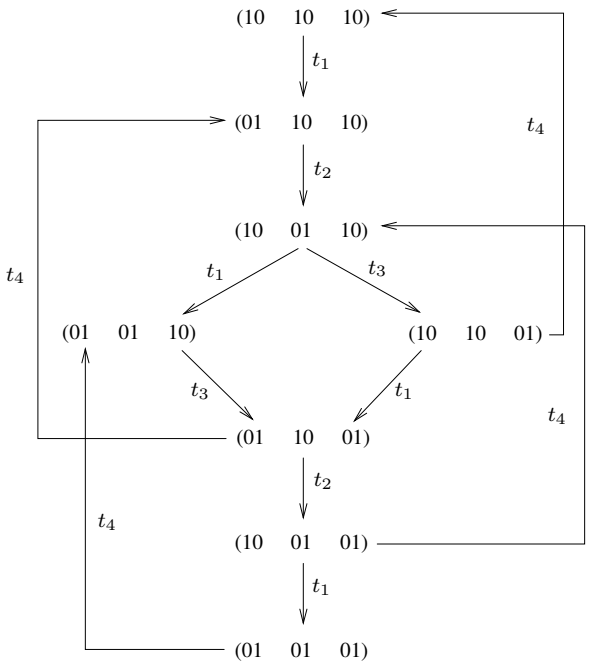


Figure 2.2: Reachability graph of the 3-buffer

- Capacity 3: the buffer has indeed capacity 3, that is, there is a reachable marking that puts one token in s_2, s_4, s_6.
- The initial marking is reachable from any reachable marking (that is, it is always possible to empty the buffer).
- Between any two reachable markings there is a path of length at most 6.

2.2 Train tracks

Four cities are connected by unidirectional train tracks building a circle. Two trains circulate on the tracks. Our task is to ensure that it will never be the case that two trains occupy the same track.

Figure 2.3 shows a solution of the problem modeled as a Petri net. the four tracks are modeled by places $s_1, \ldots, s_4$. A token on s_i means that there is train in the i-th track.

The four control places $l_1, \ldots, l_4$ guarantee that no reachable marking puts more than one token on s_i. This property can be proven by means of the reachability graph shown in Figure 2.4. Since every reachable marking puts at most one token on a place, we denote a marking by the set of places marked by it. For instance, we denote by $\{l_1, s_2, l_3, s_4\}$ the marking that puts a token on l_1, s_2, l_3 and s_4.

Consider now a slightly different system. We have 8 cities connected in a circuit, and three trains use the tracks. To increase safety, we have to guarantee that there

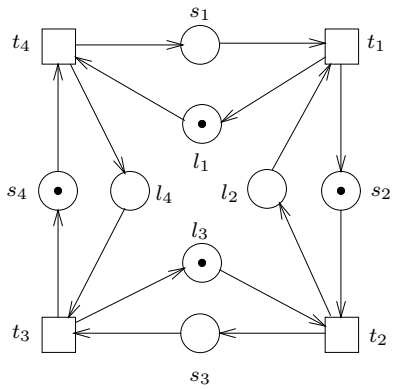


Figure 2.3: Train tracks (first version)

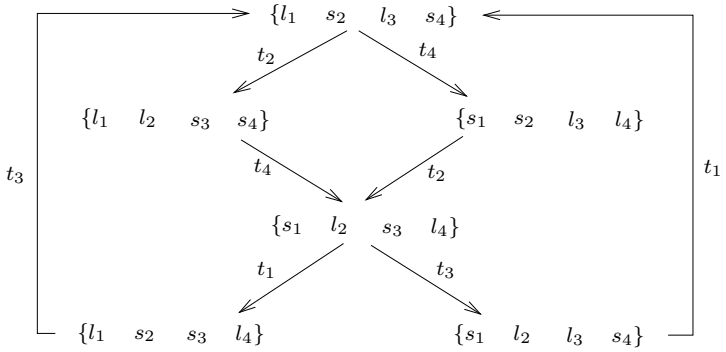


Figure 2.4: Reachability graph of the Petri net of Figure 2.3

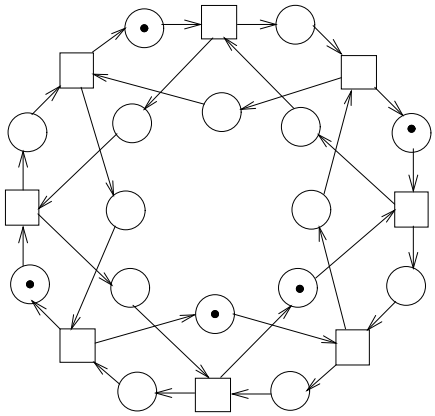


Figure 2.5: Train tracks (second version)

always is at least one empty track between any two trains.

The Petri net of Figure 2.5 is a solution of the problem: The reader can construct the reachability graph and show that the desired property holds. However, the graph is pretty large!

2.3 Dining philosophers

Four philosophers sit 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 (see the top of Figure 2.6). Unfortunately the spaghetti is of a particularly slippery type, and a philosopher needs both forks in order to eat it. The philosophers have agreed on the following protocol to obtain the forks: Initially philosophers think about philosophy, when they get hungry they do the following: (1) take the left fork, (2) take the right fork and start eating, (3) return both forks simultaneously, and repeat from the beginning. Figure 2.6 shows a Petri net model of the system.

Two interesting questions about this systems are:

- Can the philosophers starve to death (because the system reaches a deadlock)?
- Will an individual philosopher eventually eat, assuming she wants to?

2.4 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 (see Figure 2.7)

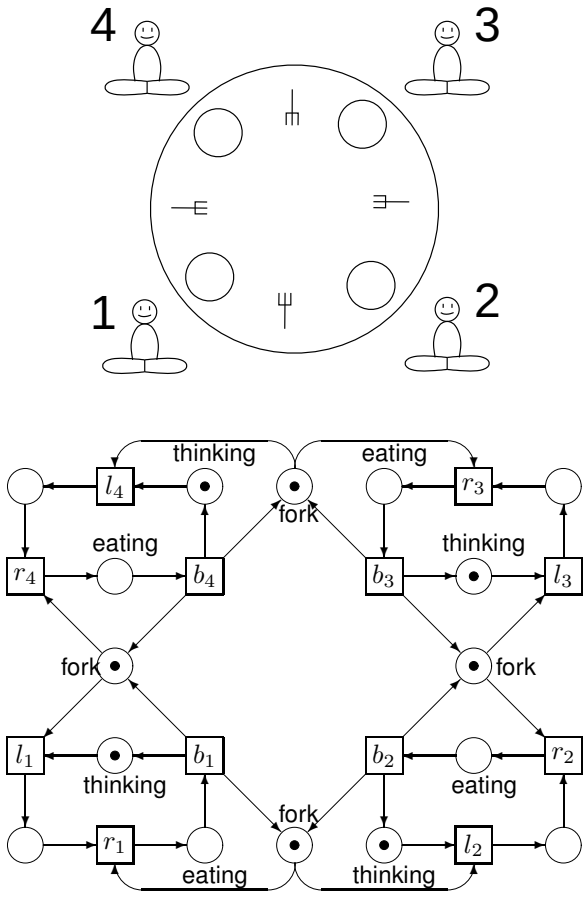


Figure 2.6: Petri net model of the dining philosophers

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

We model the system with a Petri net. The puzzle mentions the following *objects*: Man, wolf, goat, cabbage, boat. Both can be on either side of the river. It also mentions the following *actions*: Crossing the river, wolf eats goat, goat eats cabbage.

Objects and their states are modeled by places. (We can omit the boat, because it is always going to be on the same side as the man.) Actions are modeled by transitions. Figure 2.7 shows the transitions for the three actions.

2.5 Peterson's algorithm

Peterson's algorithm is a well-known solution to the mutual exclusion problem for two processes.

var $m_1, m_2 : \{false, true\}$ (**init** *false*);
$hold : \{1, 2\}$;

while $true$ **do**	**while** $true$ **do**
$m_1 := true$;	$m_2 := true$;
$hold := 1$;	$hold := 2$;
await$(\neg m_2 \vee hold = 2)$;	**await**$(\neg m_1 \vee hold = 1)$;
(critical section);	(critical section);
$m_1 := false$;	$m_2 := false$;
od	**od**

The Petri net of Figure 2.8 models this algorithm. The variable m_i is modeled by the places $m_i = true$ and $m_i = false$. A token on $m_i = true$ means that at the current state of the program (marking) the variable m_i has the value $true$ (so the Petri net must satisfy the property that no reachable marking puts tokens on both $m_i = true$ and $m_i = false$ at the same time). Variable $hold$ is modeled analogously.

A token on p_4 (q_4) indicates that the left (right) process is in its critical section. Mutual exclusion holds if no reachable marking puts a token on p_4 and q_4. The Petri net has 20 reachable markings.

2.6 The action/reaction protocol

Two agents must repeatedly exchange informations. When an agent requests an information from the other one, it must wait for an answer before proceeding. The task is to design a protocol for the exchanges. In particular, the protocol must guarantee that it is not possible to reach a situation in which both processes are waiting from an answer from the other one.

A first attempt at a solution is shown in Figure 2.9. Requests are modeled by the $Action$ transitions, and replies by the $Reaction$ transitions. However, this solution can reach a deadlock: both processes can issue a request simultaneously, after which they

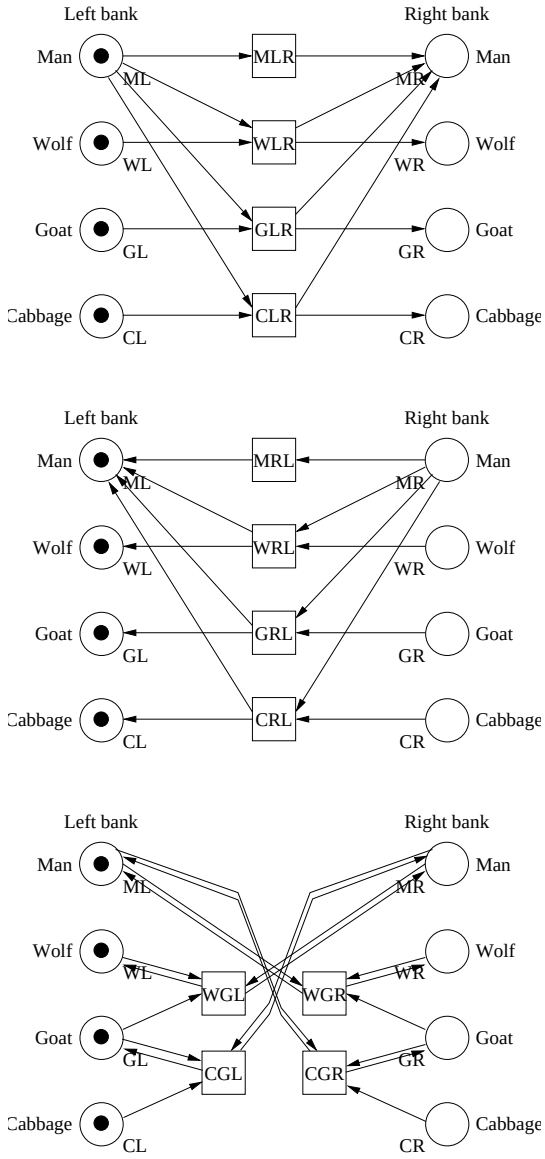


Figure 2.7: Transitions modelling the actions of the puzzle

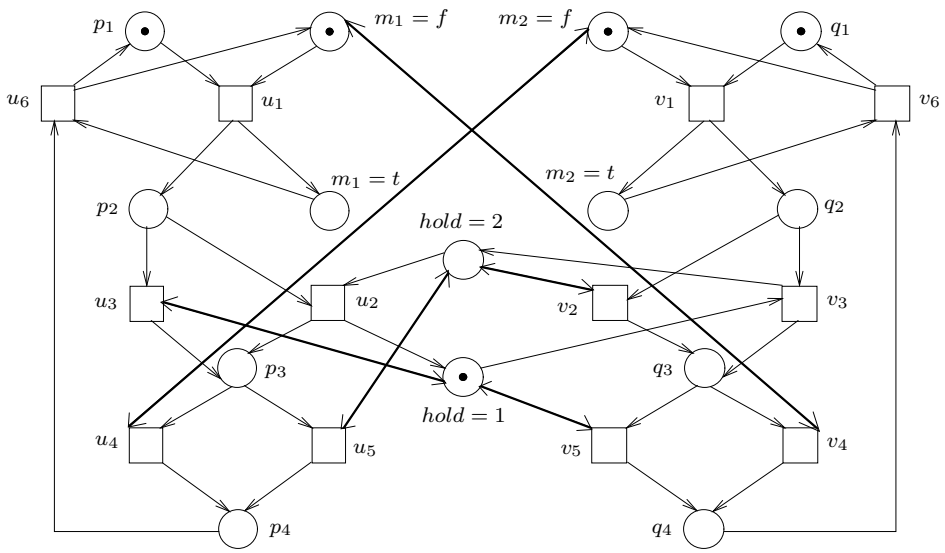


Figure 2.8: Petri net model of Peterson's algorithm

wait forever for an answer. We call such a situation a *crosstalk*. Figure 2.10 shows a second attempt. Now processes can detect that a crosstalk has taken place. If a process detects a crosstalk, it answers the request of its partner, and then continues to wait for an answer to its own request. This solution has no deadlocks (prove it!), but it exhibits the following problem: a non-cooperative process can always get answers to its requests, without ever answering any request from its partner. The solution is deadlock free, but *unfair*. The third attempt (Figure 2.11) is fair. If a process detects a crosstalk, then it answers the request of its partner, as before, but then it moves to a state in which it is only willing to receive an answer to its own question. Unfortunately, the system has again a deadlock (can you find it?).

The final attempt (Figure 2.12) is both deadlock-free and fair. The protocol works in rounds. A "good" round consists of a request and an answer. In a "bad" round both processes issue a request and they reach a crosstalk situation. Such a round continues as follows: both processes detect the crosstalk, send each other an "end-of-round" signal, wait for the same signal from their partner, and then move to their initial states.

The solution is not perfect. In the worst case there are only bad rounds, and no requests are answered at all.

2.7 Variants of the main model

Definition 2.7.1 (Nets with place capacities)
A *net with capacities* $N = (S, T, F, K)$ consists of a net (S, T, F) and a mapping $K : S \to \mathbb{N}$.
A transition t is *enabled* at a marking M of N if

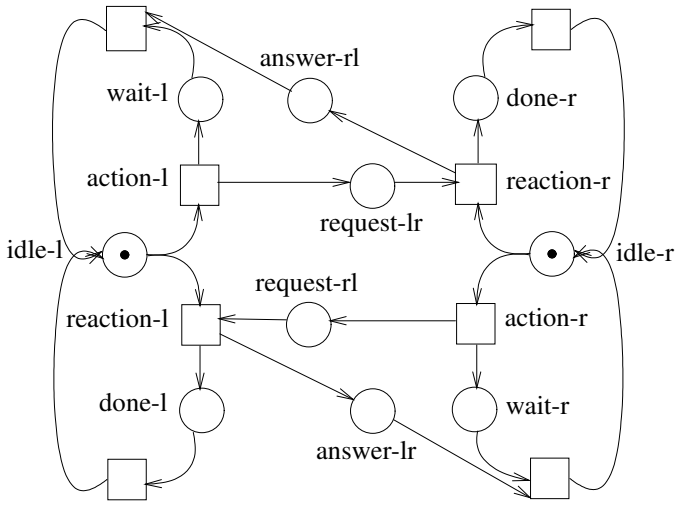


Figure 2.9: First attempt

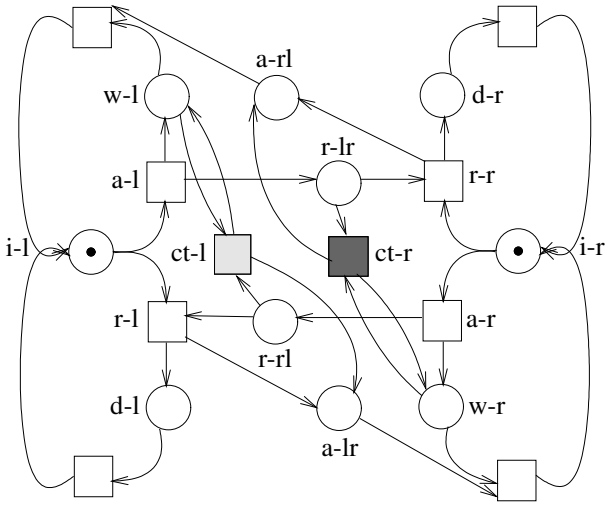


Figure 2.10: Second attempt

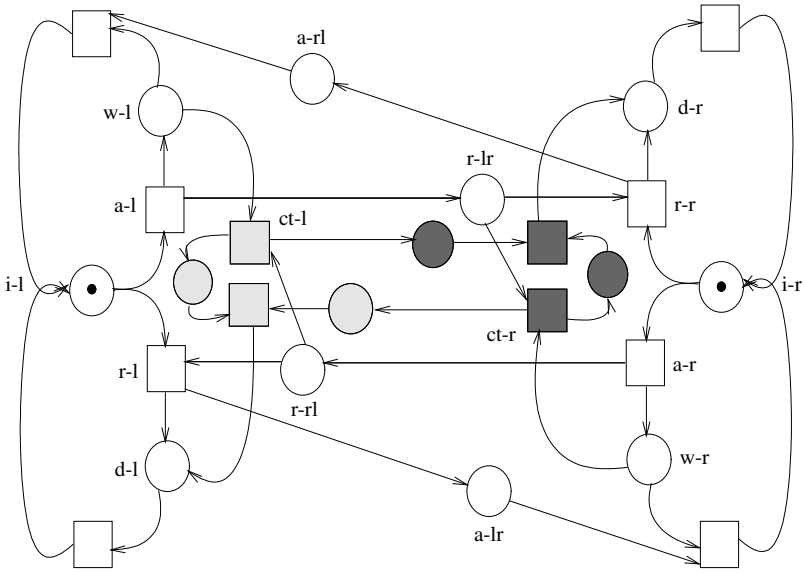


Figure 2.11: Third attempt

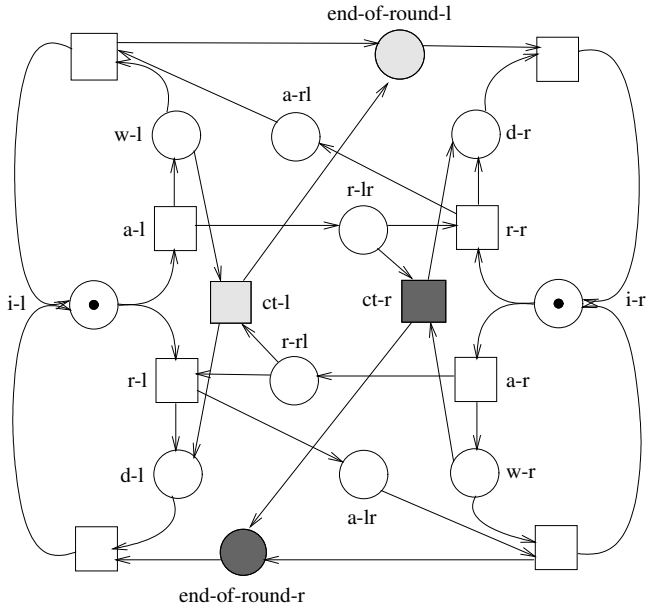
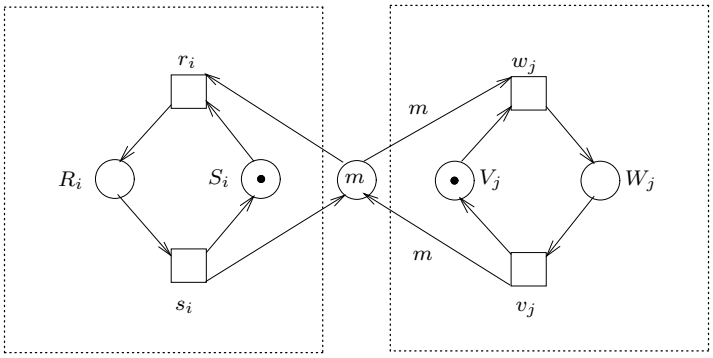


Figure 2.12: Last attempt



R_i:	Process i reads	W_i:	Process j writes
S_i:	Process i idle	V_j:	Process j idle
r_i:	Process i starts reading	w_j:	Process j starts writing
s_i:	Process i stops reading	v_j:	Process j stops writing

m readers n writers

Figure 2.13: Readers and writers

– $M(s) \geq 1$ for every place $s \in {}^\bullet t$ and
– $M(s) < K(s)$ for every place $s \in t^\bullet \setminus {}^\bullet t$

The notions of firing, Petri net with capacities, etc. are defined as in the capacity-free case.

Definition 2.7.2 (Nets with weighted arcs)
A *net with weighted arcs* $N = (S, T, W)$ consists of two disjoint sets of places and transitions and a *weight function* $W : (S \times T) \cup (T \times S) \to \mathbb{N}$. A transition t is *enabled* at a marking M of N if $M(s) \geq W(s,t)$ for every $s \in S$. If t is enabled then it can *occur* leading to the marking M' defined by

$$M'(s) = M(s) + W(t,s) - W(s,t)$$

for every place s. Other notions are defined as in the standard model.

The Petri net with weighted arcs of Figure 2.13 models a solution to the "readers and writers" problem. A set of processes has access to a database. Processes can read concurrently, but a process can only write if no other processes reads nor writes.

Exercise: Modify the Petri net so that reading processes can not indefinitely prevent another process from writing.

Definition 2.7.3 (Nets with inhibitor arcs)
A *net with inhibitor arcs* $N = (S, T, F, I)$ consists of two disjoint sets of places and transitions, a set $F \subseteq (S \times T) \cup (T \times S)$ of arcs, and a set $I \subseteq S \times T$, disjoint with F, of *inhibitor arcs*. A transition t is *enabled* at a marking M of N if $M(s) > 0$ for every

place s such that $(s,t) \in F$, and $M(s) = 0$ for every place s such that $(s,t) \in I$. If t is enabled then it can *occur* leading to the marking M', defined as for standard Petri nets.

Definition 2.7.4 (Nets with reset arcs)
A *net with reset arcs* $N = (S, T, F, R)$ consists of two disjoint sets of places and transitions, a set $F \subseteq (S \times T) \cup (T \times S)$ of arcs, and a set $R \subseteq S \times T$, disjoint with F, of *reset arcs*. A transition t is *enabled* at a marking M of N if $M(s) > 0$ for every place s such that $(s,t) \in F \cup R$. If t is enabled then it can *occur* leading to the marking obtained after the following operations:

- Remove one token from every place s such that $(s,t) \in F$.
- Remove *all* tokens from every place s such that $(s,t) \in R$.
- Add one token to every place s such that $(t,s) \in F$.

2.8 Analysis problems

We introduce a number of properties we are interested in. We assume that nets have at least one place and one transition.

Definition 2.8.1 (System properties)
Let (N, M_0) be a Petri net.

(N, M_0) is *deadlock free* if every reachable marking enables at least one transition (that is, no reachable marking is dead).

(N, M_0) is *live* if for every reachable marking M and every transition t there is a marking $M' \in [M\rangle$ that enables t. (Intuitively: every transition can always fire again).

(N, M_0) is *bounded*, if for every place s there is a number $b \geq 0$ such that $M(s) \leq b$ for every reachable marking M. M_0 is a *bounded marking of* N if (N, M_0) is bounded. The *bound* of a place s of a bounded Petri net (N, M_0) is the number

$$max\{M(s) \mid M \in [M_0\rangle\}$$

(N, M_0) is b*-bounded* if every place has bound b.

In these notes we study the following problems:

- **Deadlock freedom**: is a given Petri net (N, M_0) deadlock-free?
- **Liveness**: is a given Petri net (N, M_0) live?
- **Boundedness**: is a given Petri net (N, M_0) bounded?
- b**-boundedness**: given $b \in \mathbb{N}$ and a Petri net (N, M_0), is (N, M_0) b-bounded?
- **Reachability**: given a Petri net (N, M_0) and a marking M of N, is M reachable?
- **Coverability**: given a Petri net (N, M_0) and a marking M of N, is there a reachable marking $M' \geq M$?

There are some simple connections between these problems:

Proposition 2.8.2

(1) Liveness implies deadlock freedom.

(2) If (N, M_0) is bounded then there is a number b such that (N, M_0) is b-bounded.

(3) If (N, M_0) is bounded, then it has finitely many reachable markings.

Proof. (1) follows immediately from the definitions. (2) and (3) follow from the definitions and from the fact that a Petri net has finitely many places. □

Sometimes we also use the following notion

Definition 2.8.3 (Well-formed nets)
A net N is *well formed* if there is a marking M_0 such that the Petri net (N, M_0) is live and bounded.

and consider the following problem

- **Well-formedness**: is a given net well formed?

Part II

Analysis Techniques for Petri Nets

Chapter 3 shows (sometimes without proofs) that **Deadlock-freedom**, **Liveness**, **Boundedness**, b**-Boundedness**, **Coverability**, and **Reachability** are all decidable. The decision procedures for these problems have high complexity, but, at the same time, results of complexity theory show that no efficient algorithms exist for them.

Since better runtimes are often required in many practical applications, we often use algorithms that can be applied to arbitrary Petri nets, but sometimes answer "don't know", or do not terminate. We call them semi-decision procedures. We also use faster decision procedures for special Petri net classes.

Chapter 4 is devoted to semi-decision procedures. Chapter 5 presents efficient decision algorithms for three classes: S-nets, T-nets, and Free-Choice nets

Chapter 3

Decision procedures

3.1 A decision procedure for Boundedness

The b-**Boundednesss** problem is clearly decidable: if the input Petri net (N, M_0) has n places, then the number of b-bounded markings of N is n^{b+1}. So we can decide b-**Boundedness** by constructing the reachability graph of (N, M_0) until either the construction terminates, or we find a reachable marking that is not b-bounded.

The same idea gives a semi-decision procedure for **Boundedness**: again, we construct the reachability graph. If the input (N, M_0) is bounded, then there are finitely many reachable markings, the construction terminates, and we can return "bounded". However, if the net is unbounded then this procedure does not terminate.

We now give a decision procedure for **Boundedeness**. We need two lemmas. The first one is a simple adaptation of König's Lemma; the second is known as Dickson's Lemma.

Lemma 3.1.1 (Königs lemma) *Let $G = (V, E)$ be the reachability graph of a Petri net (N, M_0). If V is infinite, then G contains an infinite simple path.*

Proof. Assume $V = [M_0\rangle$ is infinite. For every reachable marking M there is a simple path π_M from M_0 to M. Since M_0 has finitely many immediate successors (at most one for each transition of N), and each π_M visits one of them, at least one immediate successor M_1 of M_0 has infinitely many successors in $(V \setminus \{M_0\}, E)$, that is, $[M_1\rangle \setminus \{M_0\}$ is infinite. Iterating this argument we construct an infinite simple path $M_0 M_1 M_2 \cdots$. □

Lemma 3.1.2 (Dickson's lemma) *For every infinite sequence $A_1 A_2 A_3 \ldots$ of vectors of $\mathbb{N}^k$ there is an infinite sequence $i_1 < i_2 < i_3 \ldots$ of indices such that $A_{i_1} \leq A_{i_2} \leq A_{i_3} \ldots$.*

Proof. By induction on k
Basis: $k = 1$. Then the elements of $\mathcal{A}$ are just numbers. The set $\{A_1, A_2, \cdots\}$ has

a minimum, say c_1. Choose i_1 as some index (say, the smallest), such that $A_{i_1} = c_1$. Consider now the set $\{A_{i_1+1}, A_{i_1+2}, \cdots\}$. The set has a minimum c_2, which by definition satisfies $c_1 \leq c_2$. Choose i_2 as the the smallest index $i_2 > i_1$ such that $A_{i_2} = c_2$. Etc.
Step: $k > 1$. Given a vector A_i, let A'_i be the vector of dimension $k-1$ consisting of the first $k-1$ components of A_i, and let a_i be the last component of A_i. We write $A_i = (A'_i \mid a_i)$.
Since the vectors of $A'_1 A'_2 A'_3 \cdots$ have dimension $k-1$, by induction hypothesis there is an infinite subsequence $A'_{i_1} \leq A'_{i_2} \leq A'_{i_3} \cdots$. Consider now the sequence $a_{i_1} a_{i_2} a_{i_3} \cdots$. By induction hypothesis there is a subsequence $a_{j_1} \leq a_{j_2} \leq a_{j_3} \cdots$. But then we have $A_{j_1} \leq A_{j_2} \leq A_{j_3} \cdots$, and we are done. □

Remark: Lemma 3.1.2 shows that the partial order $\leq \subseteq \mathbb{N}^k \times \mathbb{N}^k$ is a *well-quasi-order*. Given a set A, and a partial order $\preceq \subseteq A \times A$, we say that $\preceq$ is a well-quasi-order if every infinite sequence $a_1 a_2 a_3 \cdots \in A^\omega$ contains an infinite chain $a_{i_1} \preceq a_{i_2} \preceq \cdots$. In the next section we examine well-quasi-orders in more detail.

We use König's Lemma and Dickson's lemma to provide the following characterization of unboundedness.

Theorem 3.1.3 (N, M_0) *is unbounded iff there are markings* M *and* L *such that* $L \neq 0$ *and* $M_0 \xrightarrow{*} M \xrightarrow{*} (M + L)$

Proof. $(\Leftarrow)$: Assume there are such markings M, L. By the Monotonicity Lemma we have

$$M_1 \xrightarrow{*} (M_1 + L) \xrightarrow{*} (M_1 + 2 \cdot L) \xrightarrow{*} \ldots$$

So the set $[M_0\rangle$ of reachable markings is infinite and (N, M_0) is unbounded.
$(\Rightarrow)$ Assume (N, M_0) is unbounded. Then the set $[M_0\rangle$ of reachable markings is infinite. By Königs lemma there is an infinite firing sequence $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_1} M_2 \ldots$ that never visits a marking twice. By Dickson's Lemma there are M_i and M_j such that $M_0 \xrightarrow{*} M_i \xrightarrow{*} M_j$ and $M_i \leq M_j$. Let $M \equiv M_i$ and $L \equiv M_j - M_i$. □

Theorem 3.1.4 Boundedness *is decidable.*

Proof. We give an algorithm that always terminates and always returns the correct answer: " bounded" or "unbounded". The algorithm explores the reachability graph of the net using breadth-first search. After adding a new marking M', the algorithm checks if the part of the graph already constructed contains a sequence $M_0 \xrightarrow{*} M \xrightarrow{*} M'$ such that $M \leq M'$ (and $M \neq M'$, because M' is new). The algorithm terminates if it finds such a sequence, in which case it returns "unbounded", or if it cannot add any new marking, in which case it returns "bounded".

If (N, M_0) is bounded, then by Theorem 3.1.3 the algorithm never finds a new marking M' satisfying the condition above. So, since the Petri net has only finitely

many reachable markings, the algorithm terminates because it cannot find any new marking, and correctly returns "bounded".

If (N, M_0) is unbounded, then there are infinitely many reachable markings, and the algorithm cannot terminate because it runs out of reachable markings. On the other hand, by Theorem 3.1.3 the algorithm eventually finds markings M' and M as above, and so it correctly answers "unbounded". □

3.2 Decision procedures for Coverability

The reachability graph of a Petri net can be infinite, in which case the algorithm for computing the reachability graph will not terminate. Therefore, the algorithm cannot decide that a given marking is not coverable. In this section we introduce several decision procedures that overcome this problem.

3.2.1 Coverability graphs

We show how to construct a *coverability graph* of a Petri net (N, M_0). The coverability graph is always finite, and satisfies the following property: a marking M of (N, M_0) is coverable iff some node M' of the coverability graph of (N, M_0) covers M, i.e., satisfies $M' \geq M$.

We introduce a new symbol ω. Intuitively, it stands for an arbitrarily large number. We extend the arithmetic on natural numbers with ω as follows. For all $n \in \mathbb{N}$:

$$
\begin{aligned}
&n + \omega = \omega + n = \omega,\\
&\omega + \omega = \omega,\\
&\omega - n = \omega,\\
&0 \cdot \omega = 0\\
&n \geq 1 \Rightarrow n \cdot \omega = \omega \cdot n = \omega,\\
&n \leq \omega \text{ and } \omega \leq \omega.
\end{aligned}
$$

Observe that $\omega - \omega$ remains undefined, but we will not need it.

We extend the notion of markings to *ω-markings*. An ω-marking of a net $N = (S, T, F)$ is a mapping $M \colon S \to \mathbb{N} \cup \{\omega\}$. Intuitively, in an ω-marking, each place s has either a certain number of tokens or "arbitrarily many" tokens.

The enabledness condition and the firing rule neatly extend to ω-markings with the extended arithmetic rules: recall that a transition t is enabled at a marking M if $M(s) > 0$ for every $s \in {}^\bullet t$. Now $M(s) > 0$ may hold because $M(s) = \omega$. Further, recall that if t is enabled, then it can fire, leading from M to the marking M' given by:

$$
M'(s) = \begin{cases} M(s) - 1 & \text{if } s \in {}^\bullet t \setminus t^\bullet \\ M(s) + 1 & \text{if } s \in t^\bullet \setminus {}^\bullet t \\ M(s) & \text{otherwise} \end{cases}
$$

If $s \in {}^\bullet t \cup t^\bullet$ and $M(s) = \omega$, then we have $M'(s) = \omega$. That is, if a place contains ω tokens, then firing a transition will not change its number of tokens, even if the transition is connected with an arc to the place.

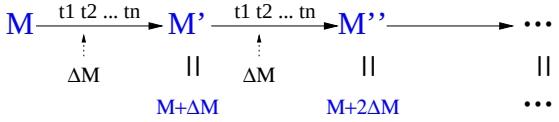


Figure 3.1: Pumping tokens.

COVERABILITY-GRAPH$((S, T, F, M_0))$
1 $(V, E, v_0) := (\{M_0\}, \emptyset, M_0)$;
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 $E := E \cup \{(M, t, M')\}$;
13 **return** (V, E, v_0);

ADDOMEGAS(M, t, M', V, E)
1 **for** $M'' \in V$
2 **do if** $M'' < M'$ and $M'' \xrightarrow{*}_E M$
3 **then** $M' := M' + ((M' - M'') \cdot \omega)$;
4 **return** M';

Figure 3.2: Algorithm for the construction of the coverability graph

Assume $M' \in [M\rangle$ and $M \leq M'$. Then there is some sequence of transitions $t_1 t_2 \dots t_n$ such that $M \xrightarrow{t_1 t_2 \dots t_n} M'$. By the Monotonicity Lemma, there is a marking M'' with $M' \xrightarrow{t_1 t_2 \dots t_n} M''$. Further, if we denote $\Delta M := M' - M$, then $M'' = M' + \Delta M = M + 2\Delta M$ (see Figure 3.1). 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 s for which $\Delta M(s) > 0$.

The main idea for the construction of the coverability graph is to replace the marking M' by the ω-marking $M' + \omega \cdot \Delta M$. The algorithm is shown in Figure 3.2. The following notations are used in the AddOmegas subroutine:

- $M'' \rightarrow_E M$ iff $(M'', t, M) \in E$ for some $t \in T$.
- $M'' \xrightarrow{*}_E M$ iff $\exists n \geq 0 \colon \exists M_0, M_1, \dots, M_n \colon M'' = M_0 \rightarrow_E M_1 \rightarrow_E M_2 \rightarrow_E \cdots \rightarrow_E M_n = M$.

Observe that COVERABILITY-GRAPH is very similar to REACHABILITY-GRAPH, it just adds a call to subroutine AddOmegas(M, t, M', V, E). 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.

We show that COVERABILITY-GRAPH terminates, and that a marking M of (N, M_0) is coverable iff some node M' of the coverabilitygraph of (N, M_0) covers M, i.e., satisfies $M' \geq M$

Theorem 3.2.1 COVERABILITY-GRAPH *terminates.*

Proof. Assume that COVERABILITY-GRAPH does not terminate. We derive a contradiction. If COVERABILITY-GRAPH does not terminate, then it constructs an infinite graph. Since every node of the graph has at most $|T|$ successors, by König's lemma 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 a 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-1}. □

For the rest of the proof we start with a lemma.

Lemma 3.2.2 *For every ω-marking M' added by the algorithm to V and for every $k > 0$, there there is a reachable marking M'_k satisfying $M'_k(s) = M'(s)$ for every place s such that $M'(s) \in \mathbb{N}$, and $M'_k(s) > k$ for every place s such that $M'(s) = \omega$.*

Proof. We prove that if all ω-markings added so far to V satisfy the property, then the next one also does. Assume the algorithm currently explores marking M and transition t, and let $M \xrightarrow{t} M_1$. By induction hypothesis, for every $k > 0$, there there is a reachable marking M_k satisfying $M_k(s) = M(s)$ for every place s such that $M(s) \in \mathbb{N}$, and $M_k(s) > k$ for every place s such that $M(s) = \omega$. If AddOmegas does not add any ωs, then we can take M'_k as the result of firing t from M_k. Assume AddOmegas finds an ω-marking M'' such that $M'' \xrightarrow{*}_E M \xrightarrow{t} M_1$. Then there is a sequence σ such that $M'' \xrightarrow{\sigma} M \xrightarrow{t} M_1$. By induction hypothesis, for every $k > 0$, there is a reachable marking M''_k satisfying $M''_k(s) = M''(s)$ for every place s such that $M''(s) \in \mathbb{N}$, and $M''_k(s) > k$ for every place s such that $M''(s) = \omega$. Then, starting from a sufficiently large k, the marking M''_k enables σt (for instance, take $k = |\sigma t|$, since a σt can remove at most $|\sigma t|$ tokens from a place). We can then choose M_k as the marking satisfying $M''_k \xrightarrow{\sigma t} M'_k$.

If AddOmegas finds several ω-markings M'' such that $M'' \xrightarrow{*}_E M \xrightarrow{t} M_1$, we repeat the argument above. □

Theorem 3.2.3 *Let (N, M_0) be a Petri net and let M be a marking of N. There is a reachable marking $M' \geq M$ iff the coverability graph of (N, M_0) contains an ω-marking $M'' \geq M$.*

Proof. ($\Rightarrow$): Assume there is a reachable marking $M' \geq M$. Then some firing sequence

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \cdots M_{n-1} \xrightarrow{t_n} M'$$

of (N, M_0) leads from M_0 to M'. By the definition of the algorithm, the coverability graph contains a path

$$M_0 \xrightarrow{t_1} M'_1 \xrightarrow{t_2} M'_2 \cdots M'_{n-1} \xrightarrow{t_n} M'_n$$

such that $M'_i \geq M_i$ for every $1 \leq i \leq n$. Take $M'' = M'_n$.

($\Leftarrow$): Assume the coverability graph of (N, M_0) contains an ω-marking $M'' \geq M$. By Lemma 3.2.2, there is a reachable marking M''_k satisfying $M''_k(s) = M''(s)$ for every place s such that $M''(s) \in \mathbb{N}$, and $M''_k(s) > k$ for every place s such that $M''(s) = \omega$. Take k larger than any of the components of M, and set $M' = M''_k$. Then clearly $M' \geq M$. □

Size of the coverability graph

Let $\mathcal{B}_n$ be the set of bounded Petri nets with k places $\{s_1, \ldots, s_k\}$ and an initial marking putting only one token on s_1 and zero tokens elsewhere (observe that $\mathcal{B}_n$ is essentially finite, because the maximal number of transitions with different presets of postsets is 2^{2k}). It has been proved that the function giving for each $n \geq 1$ the maximal size of the reachability graph of the nets in $\mathcal{B}_n$ is not bounded by any primitive recursive function. Since for bounded Petri nets the reachability and coverability graphs coincide, the same result holds for the coverability graphs.

3.2.2 Rackoff's theorem

The coverability graph allows us to answer coverability of *any* marking. However, **Coverability** asks whether a particular marking M can be covered. The question is whether we can give a bound on the size of the *fragment* of the coverability graph we need to construct to find an ω-marking covering M.

Definition 3.2.4 (Integer nets) Let $N = (S, T, F)$ be a net. A *generalized marking* of N (*g-marking* for short) is a mapping $G\colon S \to \mathbb{Z}$. An *integer net* is a pair (N, G_0) where N is a net and G_0 is a g-marking. A g-marking G enables all transitions, and the occurrence of t at G leads to the marking G' given by

$$G'(s) = \begin{cases} G(s) - 1 & \text{if } s \in {}^\bullet t \setminus t^\bullet \\ G(s) + 1 & \text{if } s \in t^\bullet \setminus {}^\bullet t \\ G(s) & \text{otherwise} \end{cases}$$

We denote by $G \overset{t}{\hookrightarrow} G'$ that firing t at G yields to G'.

An *integer firing sequence* of an integer net is a sequence $G_0 \overset{t_1}{\hookrightarrow} G_1 \overset{t_2}{\hookrightarrow} \cdots \overset{t_n}{\hookrightarrow} G_m$.

Clearly, every Petri net is also an integer net, and every firing squence is also an integer firing sequence, but the converse does not hold.

In the rest for the section we fix a net N with places $\{s_1, \ldots, s_k\}$, and identify g-markings with vectors of $\mathbb{Z}^k$.

Definition 3.2.5 *Let $G \in \mathbb{Z}^k$ be a g-marking of N and let $0 \leq i \leq k$. We say that G is i-natural if its first i-components are natural numbers, i.e., if $G(j) \geq 0$ for every $1 \leq j \leq i$. If moreover $G(j) < r$ for every $1 \leq j \leq i$, then we say that G is (i, r)-natural.*

An integer sequence $\sigma = G_0 \stackrel{t_1}{\hookrightarrow} \cdots \stackrel{t_m}{\hookrightarrow} G_m$ is *i-natural* (respectively *(i, r)-natural*) if every generalized marking of σ is i-natural (respectively (i, r)-natural). Given a g-marking $G \in \mathbb{Z}^k$, we say that σ is *(i, G)-covering* if $G_m(j) \geq G(j)$ for every $1 \leq j \leq i$.

Intuitively, G is i-natural if its restriction to the first i places is a "normal" marking, and σ is i-natural if its restriction to the first i places is a "normal" firing sequence. So, in particular, deciding if M is coverable in a Petri net (N, M_0) with k places is equivalent to deciding if (N, M_0) has a (k, M)-covering and k-natural sequence.

We prove the following result:

Theorem 3.2.6 *Let $n = \max(1, |G(1)|, \ldots |G(k)|)$. For every $G_0 \in \mathbb{Z}^k$, if (N, G_0) has a (k, G)-covering, k-natural sequence, then it has one of length at most $(n + 1)^{(2k)^k}$.*

This upper bound is not very precise. The only important aspect is the double exponential dependency on k, the number of places of the net. The proof follows easily from the following lemma, which gives a tighter bound, but in the form of a recursively defined function:

Lemma 3.2.7 *For every $G_0 \in \mathbb{Z}^k$ and for every $1 \leq i \leq k$, if (N, G_0) has an (i, G)-covering, i-natural sequence, then it has one of length at most $f(i)$, where f is inductively defined as follows:*

- *$f(0) = 1$, and*
- *$f(i) = (nf(i-1))^i + f(i-1)$ for every $1 \leq i \leq k$.*

Proof The proof is by induction on i.
Base: $i = 0$. Follows from the fact that the sequence $\sigma = G_0$ is $(0, G)$-covering and 0-natural.
Step: $i > 0$. Assume (N, G_0) has an (i, G)-covering, i-natural sequence. We consider two cases:

Case 1: (N, G_0) has an (i, G)-covering, $(i, nf(i-1))$-natural sequence.
Assume the sequence is

$$\sigma = G_0 \stackrel{t_1}{\hookrightarrow} \cdots \stackrel{t_m}{\hookrightarrow} G_m$$

and assume further that it has minimal length.

We claim that $G_0, G_1, \ldots, G_m$ are pairwise different in the first i places. Assume the contrary: there exist $\alpha < \beta$ such that $G_\alpha(j) = G_\beta(j)$ for every $1 \leq j \leq i$. Then the sequence

$$\sigma' = G_0 \stackrel{t_1}{\hookrightarrow} \cdots \xrightarrow{t_{\alpha-1}} G_\alpha \stackrel{t_{\beta+1}}{\hookrightarrow} G'_{\beta+1} \stackrel{t_{\beta+2}}{\hookrightarrow} \cdots \stackrel{t_m}{\hookrightarrow} G'_m$$

is also (i, G)-covering and $(i, nf(i-1))$-natural, contradicting the minimality of σ. This proves the claim.

Since σ is $(i, nf(i-1))$-natural, for every g-marking G' appearing in σ we have $0 \leq G'(j) < nf(i-1)$ for every $1 \leq j \leq i$. There are at most $(nf(i-1))^i$ g-markings G' different in the first i places satisfying $0 \leq G'(j) < nf(i-1)$. By the claim above the length of σ is at most $(nf(i-1))^i$.

Case 2: (N, G_0) has no (i, G)-covering, $(i, nf(i-1))$-natural sequence.
Then there is an (i, G)-covering, i-natural sequence that is *not* $(i, nf(i-1))$-natural. Let this sequence be

$$\sigma = G_0 \overset{t_1}{\hookrightarrow} G_1 \overset{t_2}{\hookrightarrow} \cdots G_{m-1} \overset{t_m}{\hookrightarrow} G_m$$

Let $G_{\alpha+1}$ be the first vector of σ that is not $(i, nf(i-1))$-natural. Without loss of generality, we can assume $G_{\alpha+1}(i) \geq nf(i-1)$. Then the prefix

$$G_0 \overset{t_1}{\hookrightarrow} \cdots \overset{t_\alpha}{\hookrightarrow} G_\alpha$$

is (i, G_α)-covering and $(i, nf(i-1))$-natural. As in the previous case, we can assume $\alpha \leq (nf(i-1))^i$.

Since

$$G_{\alpha+1} \overset{t_{\alpha+1}}{\hookrightarrow} \cdots \overset{t_m}{\hookrightarrow} G_m$$

is an $(i-1, G)$-covering and $(i-1)$-natural sequence of $(N, G_{\alpha+1})$, by induction hypothesis there exists another $(i-1, G)$-covering and $(i-1)$-natural sequence

$$G_{\alpha+1} \overset{u_1}{\hookrightarrow} H_1 \overset{u_2}{\hookrightarrow} \cdots \overset{u_\ell}{\hookrightarrow} H_\ell$$

of $(N, G_{\alpha+1})$ of length at most $f(i-1)$, that is, $\ell \leq f(i-1)$. Since $G_{\alpha+1}(i) \geq nf(i-1)$, and a sequence of length $f(i-1)$ can remove at most $(f(i-1)-1)$ tokens from the place s_i, after the execution of the new sequence we still have $H_\ell(s_i) \geq n \geq G(s_i)$ and $H_\ell(s_i) \geq 0$. So the sequence

$$\sigma' = G_0 \overset{t_1}{\hookrightarrow} \cdots \overset{t_\alpha}{\hookrightarrow} G_\alpha \overset{t_{\alpha+1}}{\hookrightarrow} G_{\alpha+1} \overset{u_1}{\hookrightarrow} H_1 \overset{u_2}{\hookrightarrow} \cdots \overset{u_\ell}{\hookrightarrow} H_\ell$$

is an (i, G)-covering and i-natural sequence of (N, G_0) of length at most $(nf(i-1))^i + f(i-1)$. □

We can now proceed to prove Theorem 3.2.6:

Proof of Theorem 3.2.6. Define $g(0) = n+1$ and $g(i) = (g(i-1))^{2k}$ for every $1 \leq i \leq k$. Observe that $g(i) \geq n+1$ for every $i \geq 0$. We prove $f(i) \leq g(i)$ for every $0 \leq i \leq k$ by induction on i. For $i = 0$ we have $f(0) = 1 \leq n = g(0)$. For $i > 0$ we

have

$$\begin{aligned} f(i) &= (nf(i-1))^i + f(i-1) \\ &\leq (ng(i-1))^k + g(i-1) \\ &\leq (ng(i-1))^k + g(i-1)^k \\ &= (n^k+1)g(i-1)^k \\ &\leq (n+1)^k g(i-1)^k \\ &\leq g(i-1)^k g(i-1)^k \\ &= g(i) \end{aligned}$$

By Lemma 3.2.7, if (N, G_0) has a (k, G)-covering sequence, then it has one of length at most $g(k) = (n+1)^{(2k)^k}$. □

By Theorem 3.2.6, in order to decide coverability of M we can just construct the reachability graph using breadth-first search up to depth $(n+1)^{(2k)^k}$, where n is the maximal number of tokens in any place of the marking M, and k is the number of places in the net. Clearly, the same holds for the coverability graph, because, loosely speaking, it just "improves" our chances of covering M.

It can be asked whether Rackoff's bound is the best one can hope for. The affirmative answer was essentially proved by Lipton, who showed that a Petri net with $O(k^2)$ places (and at most one token per place in the initial marking) can simulate a counter machine whose counters are bounded by 2^{2^k}. One can use this result to show that the shortest path leading to a marking covering M can have length up to $2^{2^{\sqrt{k}}}$ for a Petri net with k places.

3.2.3 The backwards-reachability algorithm

Definition 3.2.8 (Upward-closed sets of markings)
A set $\mathcal{M}$ of markings of a net N is *upward closed* if $M \in \mathcal{M}$ and $M' \geq M$ imply $M' \in \mathcal{M}$.

A marking M of an upward closed set $\mathcal{M}$ is *minimal* if there is no $M' \in \mathcal{M}$ such that $M' \leq M$ and $M' \neq M$.

Observe that an upward closed set is completely determined by its minimal elements: two upwards closed sets are equal iff their sets of minimal elements are equal.

Lemma 3.2.9 *Every upward-closed set of markings has finitely many minimal elements.*

Proof. Assume $\mathcal{M}$ is upward closed and has infinitely many minimal markings $M_1, M_2, \ldots$. By Dickson's Lemma there are $i \neq j$ such that $M_i \leq M_j$. But then M_j is not minimal. □

An important consequence of the lemma is that every upwards closed set can be *finitely represented* by its set of minimal elements.

Definition 3.2.10 *Let $\mathcal{M}$ be a set of markings of a net $N = (S, T, F)$, and let $t \in T$ be a transition. We define*

$$\begin{aligned} pre(\mathcal{M}, t) &= \{M' \mid M' \xrightarrow{t} M \text{ for some } M \in \mathcal{M}\} \\ pre(\mathcal{M}) &= \bigcup_{t \in T} pre(\mathcal{M}, t) \end{aligned}$$

and further

$$\begin{aligned} pre^0(\mathcal{M}) &= \mathcal{M} \\ pre^{i+1}(\mathcal{M}) &= pre\big(pre^i(\mathcal{M})\big) \text{ for every } i \geq 0 \\ pre^*(\mathcal{M}) &= \bigcup_{i=0}^{\infty} pre^i(\mathcal{M}) \end{aligned}$$

Lemma 3.2.11 *If $\mathcal{M}$ is upward closed, then $pre(\mathcal{M})$ is also upward closed.*

Proof. Let $M' \in pre(\mathcal{M})$. We have to prove that $M' + M'' \in pre(\mathcal{M})$ holds for every marking M''.

Since $M' \in pre(\mathcal{M})$ there is $M \in \mathcal{M}$ and a transition t such that $M' \xrightarrow{t} M$. By the firing rule we have $M' + M'' \xrightarrow{t} M + M''$ for every marking M''. Since $\mathcal{M}$ is upward closed, we have $M + M'' \in \mathcal{M}$. Since $M' + M'' \xrightarrow{t} M + M''$, we finally get $M' + M'' \in pre(\mathcal{M})$. □

Theorem 3.2.12 *Let $\mathcal{M}$ be an upward-closed set of markings of a net N. Then there is $i \geq 0$ such that*

$$pre^*(\mathcal{M}) = \bigcup_{j=0}^{i} pre^j(\mathcal{M})$$

Proof. By Lemma 3.2.11, $pre^j(X)$ is upward closed for every $i \geq 0$. Since a (finite or infinite) union of upward-closed sets is upward closed, $pre^*(\mathcal{M})$ is upward closed as well.

By Lemma 3.2.9, the set m^* of minimal markings of $pre^*(\mathcal{M})$ is finite. Therefore, there exists an index i such that $m^* \subseteq \bigcup_{j=0}^{i} pre^j(\mathcal{M})$. Since this union is upward closed, we get $pre^*(\mathcal{M}) \subseteq \bigcup_{j=0}^{i} pre^j(\mathcal{M})$, and therefore, by definition of $pre^*(\mathcal{M})$, we have $pre^*(\mathcal{M}) = \bigcup_{j=0}^{i} pre^j(\mathcal{M})$. □

This theorem leads to the algorithm on the left of of Figure 3.3. However, this version is not yet directly implementable, because it manipulates infinite sets. For each

BACK1$((S,T,F,M_0),M)$
1 $\mathcal{M} := \{M' \mid M' \geq M\}$;
2 $Old\_\mathcal{M} := \emptyset$;
3 **while true**
4 **do** $Old\_\mathcal{M} := \mathcal{M}$;
5 $\mathcal{M} := \mathcal{M} \cup pre(\mathcal{M})$;
6 **if** $M_0 \in \mathcal{M}$
7 **then return** covered **end**
8 **if** $\mathcal{M} = Old\_\mathcal{M}$
9 **then return** not covered **end**

BACK2$((S,T,F,M_0),M)$
1 $m := \{M\}$;
2 $old\_m := \emptyset$;
3 **while true**
4 **do** $old\_m := m$;
5 $m := \min(m \cup \bigcup_{t\in T} pre(m,t))$;
6 **if** $\exists M' \in m : M_0 \geq M'$
7 **then return** covered **end**
8 **if** $m = old\_m$
9 **then return** not covered **end**

Figure 3.3: Backwards reachability algorithm.

operation (union and pre) or test (the tests $\mathcal{M} \neq Old\_\mathcal{M}$ and $M_0 \notin \mathcal{M}$ of the while-loop), we have to supply an implementation that uses only the *finite representation* of the set, that is, its set of minimal elements. Given a set $\mathcal{M}$, let $\min(\mathcal{M})$ denote the set of minimal elements of $\mathcal{M}$. We then have (exercise):

- $M \in \mathcal{M}$ iff there exists $M' \in \min(\mathcal{M})$ such that $M \geq M'$.
- $\mathcal{M}_1 = \mathcal{M}_2$ iff $\min(\mathcal{M}_1) = \min(\mathcal{M}_2)$.
- $\min(\mathcal{M}_1 \cup \mathcal{M}_2) = \min(\min(\mathcal{M}_1) \cup \min(\mathcal{M}_2))$.
- $\min(pre(\mathcal{M},t)) = pre(\min(\mathcal{M}),t))$.

Using these observations, we obtain the implementable version shown on the right of Figure 3.3.

The abstract backwards-reachability algorithm

The backwards reachability algorithm can be reformulated in more general terms, which allows to apply it to other models of concurrency more general than Petri nets. This is an important advantage of the backwards reachability algorithm over the coverability graph technique.

Definition 3.2.13 *G*iven a set A, and a partial order $\preceq \subseteq A \times A$, we say that $\preceq$ is a well-quasi-order (wqo) if every infinite sequence $a_1a_2a_3\cdots \in A^\omega$ contains an infinite chain $a_{i_1} \preceq a_{i_2} \preceq \cdots$ (where $i_1 < i_2 < i_3 \ldots$).

Here are some examples of well-quasi-orders:

- The pointwise order $\leq$ on $\mathbb{N}^k$.
- The subword order on Σ^* for any finite alphabet Σ.
  We say $w_1 \preceq w_2$ if w_1 is a scattered subword of w_2, that is, if w_1 can be obtained from w_2 by deleting letters. Higman's lemma states that every infinite sequence of words contains an infinite chain with respect to the subword order.

- The subtree order on the set of finite trees over a finite alphabet Σ.
  We say that $t_1 \preceq t_2$ if there is an injective mapping from the nodes of tree t_1 into the nodes of t_2 that preserves reachability: n' is reachable from n in t_1 iff the image of n' is reachable from the image of n in t_2. Kruskal's lemma states that every infinite sequence of trees contains an infinite chain with respect to the subtree order.

Definition 3.2.14 *Let A be a set and let $\preceq A \times A$ be a wqo. A set $X \subseteq A$ is upward closed if $x \in X$ and $x \preceq y$ implies $y \in X$ for every $x, y \in A$. In particular, given $x \in A$, the set $\{y \in A \mid y \succeq x\}$ is upward-closed.*

A relation $\rightarrow \subseteq A \times A$ is monotonic if for every $x \rightarrow y$ and every $x' \succeq x$ there is $y' \succeq y$ such that $x' \rightarrow y'$.

Given $X \subseteq A$, we define

$$pre(X) = \{y \in A \mid y \rightarrow x \text{ and } x \in X\}$$

Further we define:

$$\begin{aligned} pre^0(X) &= X \\ pre^{i+1}(X) &= pre\big(pre^i(X)\big) \text{ for every } i \geq 0 \\ pre^*(X) &= \bigcup_{i=0}^{\infty} pre^i(X) \end{aligned}$$

Theorem 3.2.15 *Let A be a set and let $\preceq A \times A$ be a wqo. Let $X_0 \subseteq A$ be an upward closed set and let $\rightarrow \subseteq A \times A$ be monotonic. Then there is $j \in \mathbb{N}$ such that*

$$pre^*(X) = \bigcup_{i=0}^{j} pre^i(X)$$

This theorem can be used to obtain a backwards reachability algorithm for generalizations of Petri nets, like reset Petri nets, or *lossy channel systems*, whose transition relation is monotonic. Other net models, like Petri nets with inhibitor arcs, do not have a monotonic transition relations (adding tokens may *disable* a transition), and the theorem cannot be applied to them. In fact we have:

Theorem 3.2.16 Deadlock freedom, Liveness, Boundedness, b-boundedness, Reachability, *and* **Coverability** *are all undecidable for Petri nets with inhibitor arcs.*

3.3 Decision procedures for other problems

3.3.1 Reachability

The decidability of **Reachability** was open for about 10 years until it was proved by Mayr in 1980. Kosaraju and Lambert simplied the proof in 1982 and 1992, respectively. All these algorithms and their proofs exceed the framework of this course.

In 2012 Leroux provided a new, very simple algorithm. Its proof is as complicated as the proofs of the previous ones, but the algorithm is very simple to describe.

Definition 3.3.1 (Semilinear set) A set $X \subseteq \mathbb{N}^k$ is *linear* if there is $r \in \mathbb{N}^k$ (the root) and a finite set $P \subseteq \mathbb{N}^k$ (the periods) such that

$$X = \{r + \sum_{p \in P} \lambda_p p\}$$

A *semilinear set* is a finite union of linear sets.

Observe that a semilinear set can be finitely represented as a set of pairs $\{(r_1, P_1), \ldots, (r_n, P_n)\}$ giving the roots and periods of its linear sets.

Theorem 3.3.2 *[Leroux 12] Let (N, M_0) be a Petri net and let M be a marking of M. If M is not reachable from M_0, then there exists a semilinear set $\mathcal{M}$ of markings of N such that*

(a) $M_0 \in \mathcal{M}$,

(b) if $M \in \mathcal{M}$ *and* $M \xrightarrow{t} M'$ *for some transition* t *of* N, *then* $M' \in \mathcal{M}$, *and*

(c) $M \notin \mathcal{M}$.

Given the root r and periods $p_1, \ldots, p_n$ of a semilinear set $\mathcal{M}$, we can check whether $\mathcal{M}$ satisfies (a)-(c). Indeed, checking (a) amounts to solving the linear system of diophantine equations

$$M_0 = r + \sum_{i=1}^{n} \lambda_i p_i$$

with unknowns $\lambda_1, \ldots, \lambda_n$. Similarly, checking (c) amounts to showing that

$$M = r + \sum_{i=1}^{n} \lambda_i p_i$$

has no solution. Finally, checking (b) is more complicated, but reduces to checking validity of a formula of a theory called Presburger arithmetic for which decision procedures exist.

Now, Theorem 3.3.2 can be used to give an algorithm for **Reachability** consisting of two semi-decision procedures, one that explores the reachability graph breadth-first and stops if it finds the goal marking M, and another one that enumerates all semilinear sets, and stops if one of them satisfies (a)-(c). The two procedures run in parallel, and, since one of the two is bound to terminate, yield together a decision procedure for **Reachability**.

3.3.2 Deadlock-freedom

Now we reduce **Deadlock-freedom** to **Reachability**. We proceed in two stages. First, we reduce **Deadlock-freedom** to an auxiliary problem **P**, and then we reduce **P** to reachability.

P: Given a Petri net (N, M_0) and a subset R of places of N, is there a reachable marking M such that $M(s) = 0$ for every $s \in R$?

Theorem 3.3.3 Deadlock-freedom *can be reduced to* **P**.

Proof. Let (N, M_0) be a Petri net such that $N = (S, T, F)$. Define

$$\mathcal{S} = \{R \subseteq S \mid \forall t \in T : {}^\bullet t \cap R \neq \emptyset\}$$

that is, an element of $\mathcal{S}$ contains for every transition t at least on of the input places of t. We have

(1) $\mathcal{S}$ is a finite set.

(2) A marking M of N is dead iff the set of places unmarked at M is an element of $\mathcal{S}$.

Suppose now that there is an algorithm that decides **P**. We can then decide **Deadlock-freedom** as follows. For every $R \in \mathcal{S}$ we use the algorithm for **P** to decide if some reachable marking M satisfies $M(s) = 0$ for every $s \in R$. It follows from (2) that (N, M_0) is deadlock-free if the answer is negative in all cases. Since, by (1), we only have to solve a finite number of instances of **P**, **Deadlock-freedom** is decidable. □

Theorem 3.3.4 P *can be reduced to* **Reachability**.

Proof. Let (N, M_0) be a Petri net where $N = (S, T, F)$, and let R be a set of places of N. We construct a new Petri net (N', M_0') by adding new places, transitions, and arcs to (N, M_0). We proceed in two steps (see Figure 3.4):

- Add two new places s_0 and r_0. Put one token on s_0.
- Add a transition t_0 and arcs (s_0, t_0) and (t_0, r_0).
- For every transition $t \in T$, add two arcs (s_0, t) und (t, s_0).

While s_0 remains marked, all transitions of T can fire. However, transition t_0 can occur at any time, and when this happens all transitions of T become "dead". Intuitively, the firing of t_0 "freezes" (N, M_0).

- For every place $s \in S \backslash R$ add a new transition t_s and arcs (s, t_s), (r_0, t_s), (t_s, r_0).

If r_0 is marked, then the t_s transitions can occur. These transitions "empty" the places of $S \setminus R$.

This concludes the definition of (N', M_0').

Let M_{r_0} be the marking of N' that puts one token on r_0 and no tokens elsewhere. We have

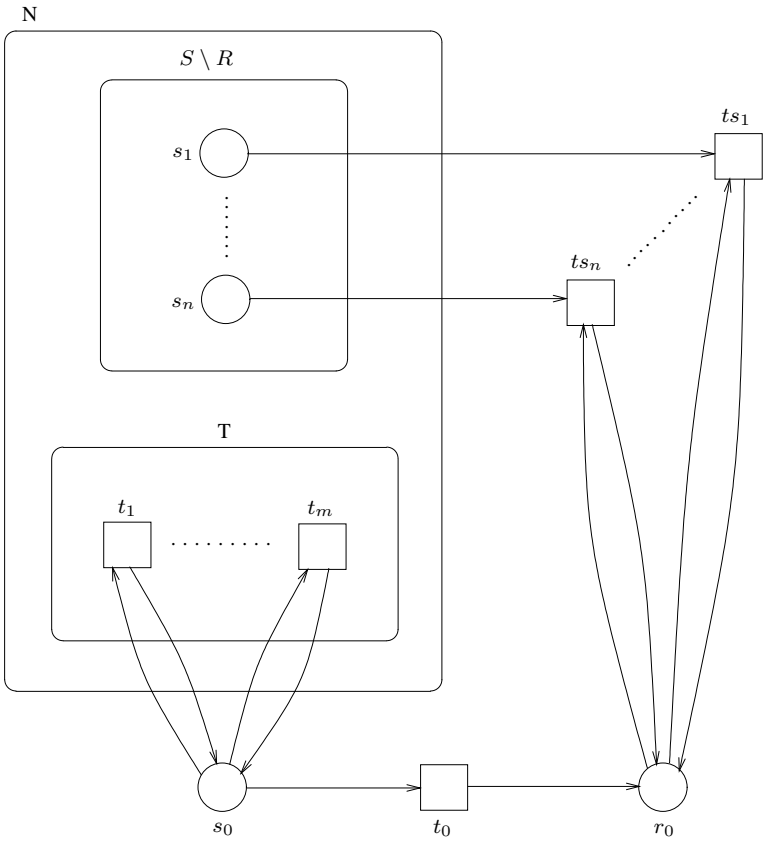


Figure 3.4: Construction of Theorem 3.3.4

(1) If some reachable marking M of (N, M_0) puts no tokens in R, then M_{r_0} is a reachable marking of (N', M_0').

To reach M_{r_0} we first fire transitions of T to reach M, then we fire t_0, and finally we fire t_s transitions until S is empty.

(2) If M_{r_0} is a reachable marking of (N', M_0'), then some marking M reachable from (N, M_0) puts no tokens in R.

M_{r_0} can only be reached by firing t_0 at a marking that puts no tokens in R (because after firing t_0 the places of R cannot be emptied anymore). So we can choose M as the marking reached immediately before firing t_0.

By (1) and (2), we can decide if some reachable marking M of (N, M_0) puts no tokens in R as follows: construct (N', M_0') and decide if M_{r_0} is reachable. Therefore, if there is an algorithm for **Reachability**, then there is also one for **P**. □

3.3.3 Liveness

Liveness can also be reduced to **Reachability**, but the proof is more complex. We sketch the reduction for the problem whether a given transition t of a Petri net (N, M_0) is live.

Let E_t be the set of markings of N that enable t. Clearly, E_t is upward closed. By Lemma 3.2.11, the set $pre^*(E_t)$ is also upward closed. Now, $pre^*(E_t)$ is the set of markings of N that enable some firing sequence ending with t. Let D_t be the complement of $pre^*(E_t)$, that is, the set of markings from which t cannot be enabled anymore. We have: (N, M_0) is live iff $[M_0\rangle \cap D_t = \emptyset$.

If D_t is a finite set of markings, and we are able to compute it, then we are done: we have reduced the liveness problem to a finite number of instances of **Reachability**. However, the set D_t may be infinite, and we do not yet know how to compute it. We show how to deal with these problems.

Every upward-closed set of markings is semilinear (exercise). Using the backwards reachability algorithm, we can compute the finite set $\min(pre^*(E_t))$, and from it we can compute a representation of $pre^*(E_t)$ as a semilinear set. Now we use a powerful result: the complement of a semilinear set is also semilinear; moreover, there is an algorithm that, given a representation of a semilinear set $X \subseteq \mathbb{N}^k$, computes a representation of the complement $\mathbb{N}^k \setminus X$. So we are left with the problem: given a Petri net (N, M_0) and a semilinear set X, decide if some marking of X is reachable from M_0.

This problem can be reduced to **Reachability** as follows (brief sketch). We construct a Petri net that first simulates (N, M_0), and then transfers control to another Petri net which nondeterministically generates a marking of X on "copies" of the places of N. This second net then transfers control to a third, whose transitions remove one token from a place of N and a token from its "copy". If X is reachable, then the first net can produce a marking of X, the second net can produce the same marking, and the third net can then remove all tokens from the first and second nets, reaching the empty marking. Conversely, if the net consisting of the three nets together can reach the empty marking, then (N, M_0) can reach some marking of X.

3.4 Complexity

Unfortunately, all the algorithms we have seen so far have very high complexity: all of them are EXPSPACE-hard. That is, the memory needed by any algorithm solving one of these problems necessarily grows exponentially in the size of the input Petri net. **Boundedness** and **Coverability** have been proved to be EXPSPACE-complete, that is, there exist algorithms for them that "only" require exponential memory. It is conjectured that the same holds for **Deadlock-freedom**, **Liveness**, and **Reachability**, but so far no proof has been found. The known algorithms for this problem have extremely high complexity: there is no primitive-recursive bound for their memory requirements. To undertsand what this means, define inductively the functions $exp_k(x)$ as follows:

- $exp_0(x) = x$;

- $exp_{k+1}(x) = 2^{exp_k(x)}$.

The worst-case time and space complexity of the known algorithms for these three problems grows faster as $\exp_k$ *for every* $k \geq 0$!!

3.5 Algorithms for bounded Petri nets

In many practical cases it is easy to show that a Petri net is bounded. In this case the set of reachable markings is finite, and the reachability graph can be computed and stored, at least in principle. If the reachability graph is available, then it is easy to give algorithms b**-Boundedness**, **Reachability**, and **Deadlock-freedom** running in linear time in the size of the reachability graph. We show now that this is also the case for **Liveness**.

Let $G = (V, E)$ be the reachability graph of a Petri net(N, M_0). We define the relation $\stackrel{*}{\longleftrightarrow} \subseteq V \times V$ as follows: $M \stackrel{*}{\longleftrightarrow} M'$ gdw. $M \stackrel{*}{\longrightarrow} M'$ und $M' \stackrel{*}{\longrightarrow} M$.

$\stackrel{*}{\longleftrightarrow}$ is clearly an equivalence relation on V. Each equivalence class $V' \subseteq V$ of $\stackrel{*}{\longleftrightarrow}$ yields together with $E' = E \cap (V' \times V)$ a *strongly connected component* (SCC) (V', E') of G.

Strongly connected components are partially ordered by the relation $<$ defined as follows: $(V', E') < (V'', E'')$ if $V' \neq V''$ and $\forall M' \in V',\ M'' \in V'' : M'' \in [M'\rangle$. The *bottom* SCCs of the reachability graph are the maximal SCCs with respect to $<$.

Proposition 3.5.1 *Let (N, M_0) be a bounded Petri net. (N, M_0) is live iff for every bottom SCC of the reachability graph of (N, M_0) and for every transition t, some marking of the SCC enables t.*

Proof. Follows easily from the definitions. □

The condition of Proposition 3.5.1 can be checked in linear time using Tarjan's algorithm, whch computes all the SCCs of a directed graph in linear time. The algorithm can be easily adapted to compute the bottom SCCs.

Chapter 4

Semi-decision procedures

4.1 Linear systems of equations and linear programming

In the next two sections we will construct linear systems of equations with integer or rational coefficients that provide partial information about our analysis problems. We will prove propositions like "if the system of equations $A \cdot X \leq b$ (we will see how this system looks like) has a rational positive solution, then the Petri net (N, M_0) is bounded" (sufficient condition), or "if M is reachable in (N, M_0), then the system of equations $B \cdot X = b$ has a solution over the natural numbers" (necessary condition). Such propositions lead to semi-decision procedures to prove or disprove a property. The complexity of these procedures depends on the complexity of solving the different systems of equations.

We define the size of a linear system of equations $A \cdot X = b$ or $A \cdot X \leq b$ where $A = (a_{ij})_{i=1,\ldots n, j=1,\ldots,m}$ and $b = (b_j)_{j=1,\ldots,m}$ as

$$\sum\{log_2|a_{ij}| \mid 1 \leq i \leq n, 1 \leq j \leq m\} + \sum\{log_2|b_j| \mid 1 \leq j \leq m\}$$

The problem of deciding whether $A \cdot X = b$ has

- a rational solution can be solved in polynomial time (though not by means of Gauss elimination!).
- an integer solution can be solved in polynomial time.
- a nonnegative integer solution is NP-complete.

The problem of deciding whether $A \cdot X \leq b$ has

- a rational solution can be solved in polynomial time. [1]

[1] In practice we often use the Simplex algorithm, which has exponential worst-case complexity, but is very efficient for most instances.

- an integer solution is NP-complete.
- a nonnegative integer solution is NP-complete.

Given a linear objective function $f(X) = c_1x_1 + \ldots c_m$ we can decide with the same runtime whether there is a solution X_{op} that maximizes $f(X)$ and, if so, the value $f(X_{op})$.

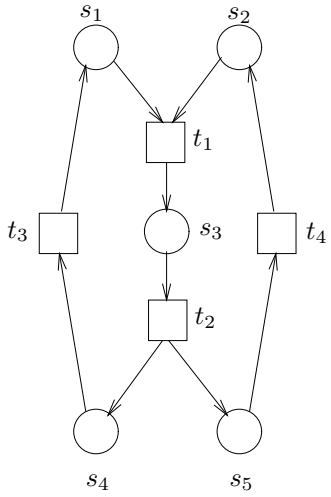
4.2 The Marking Equation

Definition 4.2.1 (Incidence matrix)
Let $N = (S, T, F)$ be a net. The *incidence matrix* $\mathbf{N} : (S \times T) \to \{-1, 0, 1\}$ is given by

$$\mathbf{N}(s,t) = \begin{cases} 0 & \text{if } (s,t) \notin F \text{ and } (t,s) \notin F \text{ or} \\ & \quad\ (s,t) \in F \text{ and } (t,s) \in F \\ -1 & \text{if } (s,t) \in F \text{ and } (t,s) \notin F \\ 1 & \text{if } (s,t) \notin F \text{ and } (t,s) \in F \end{cases}$$

The column $\mathbf{N}(-,t)$ is denoted by $\mathbf{t}$, and the row $\mathbf{N}(s,-)$ by $\mathbf{s}$.

Example 4.2.2



	t_1	t_2	t_3	t_4
s_1	-1	0	1	0
s_2	-1	0	0	1
s_3	1	-1	0	0
s_4	0	1	-1	0
s_5	0	1	0	-1

Definition 4.2.3 (Parikh-vector of a sequence of transitions)
Let $N = (S, T, F)$ be a net and let σ be a finite sequence of transitions. The *Parikh-vector* $\vec{\sigma} : T \to \mathbb{N}$ *von* σ is defined by

$$\vec{\sigma}(t) = \text{ number of occurrences of } t \text{ in } \sigma$$

Lemma 4.2.4 (Marking Equation Lemma)
Let N be a net and let $M \xrightarrow{\sigma} M'$ be a firing sequence of N. Then $M' = M + \mathbf{N} \cdot \vec{\sigma}$.

Proof. By induction on the length of σ.
Basis: $\sigma = \epsilon$. Then $M = M'$ and $\vec{\sigma} = 0$
Step: $\sigma = \tau t$ for some sequence τ and transition t. Let $M \xrightarrow{\tau} L \xrightarrow{t} M'$. We have

$$\begin{array}{lll} M' & = L + \mathbf{t} & \text{(Definition of } \mathbf{t}) \\ & = L + \mathbf{N} \cdot \vec{t} & \text{(Definition of } \vec{t}) \\ & = M + \mathbf{N} \cdot \vec{\tau} + \mathbf{N} \cdot \vec{t} & \text{(Induction hyp.)} \\ & = M + \mathbf{N} \cdot (\vec{\tau} + \vec{t}) & \\ & = M + \mathbf{N} \cdot \vec{\tau t} & \text{(Definition of Parikh-vector)} \\ & = M + \mathbf{N} \cdot \vec{\sigma} & (\sigma = \tau t) \end{array}$$

□

Example 4.2.5 *In the previous net we have* $(11000) \xrightarrow{t_1 t_2 t_3} (10001)$*, and*

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

The marking reached by firing a sequence σ from a marking M depends *only* on the Parikh-vector $\vec{\sigma}$. In other words, if M enables two sequences σ and τ with $\vec{\sigma} = \vec{\tau}$, then both σ and τ lead to the same marking.

Definition 4.2.6 (The Marking Equation)
The Marking Equation of a Petri net (N, M_0) is $M = M_0 + \mathbf{N} \cdot X$ with variables M and X.

The Marking equation leads to the following semi-algorithms for **Boundedness**, b**-Boundedness**, (Non)-**Reachability**, and **Deadlock-freedom**:

Proposition 4.2.7 (A sufficient condition for boundedness)
Let (N, M_0) *be a Petri net. If the optimization problem*

$$\begin{array}{ll} \textit{maximize} & \sum_{s \in S} M(s) \\ \textit{subject to} & M = M_0 + \mathbf{N} \cdot X \end{array}$$

has an optimal solution, then (N, M_0) *is bounded.*

Proof. Let n be the optimal solution of the problem. Then $n \geq \sum_{s \in S} M(s)$ holds for every marking M for which there exists a vector X such that $M = M_0 + \mathbf{N} \cdot X$. By Lemma 4.2.4 we have $n \geq \sum_{s \in S} M(s)$ for every *reachable* marking M, and so $n \geq M(s)$ for every reachable marking M and every place s. □

Exercise: Change the algorithm so that it checks whether a given place is bounded.

Proposition 4.2.8 (A sufficient condition for non-reachability)
Let (N, M_0) be a Petri net and let L be a marking of N. If the equation

$$L = M_0 + \mathbf{N} \cdot X \qquad (\textit{with only } X \textit{ as variable})$$

has no solution, then L is not reachable from M_0.

Proof. Immediate consequence of Lemma 4.2.4. □

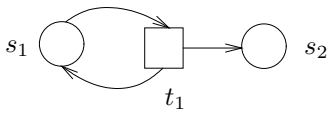
Proposition 4.2.9 (A sufficient condition for deadlock-freedom)
Let (N, M_0) be a 1-bounded Petri net where $N = (S, T, F)$. If the following system of inequations has no solution then (N, M_0) is deadlock-free.

$$\begin{array}{l} M = M_0 + \mathbf{N} \cdot X \\ \sum\limits_{s \in \cdot t} M(s) < |^\bullet t| \textit{ for every transition } t. \end{array}$$

Proof. We show: if there is a reachable dead marking M, then M is a solution of the system. By Lemma 4.2.4 and the reachability of M there is a vector X satisfying $M = M_0 + \mathbf{N} \cdot X$. Since (N, M_0) is 1-bounded, we have $M(s) \leq 1$ for every place s. Let t be an arbitrary transititon. Since M does not enable t, we have $M(s) = 0$ for at least one place $s \in {}^\bullet t$. Since M does not enable any transition, we get $\sum\limits_{s \in \cdot t} M(s) < |^\bullet t|$. □

Remark 4.2.10 *The converses of these propositions do not hold (that is why they are semi-algorithms!). Counterexamples are:*

- *To Proposition 4.2.7:*



	t_1
s_1	0
s_2	1

  (N, M_0) *ist bounded but*

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

  for every n (that is, the Marking Equation has a solution for every marking of the form $(0, n)$).

- *To Proposition 4.2.8:*

  Peterson's algorithm: the marking $(p_4, q_4, m_1 = true, m_2 = true, hold = 1)$ ist not reachable, but the Marking Equation has a solution (Exercise: find a smaller example).

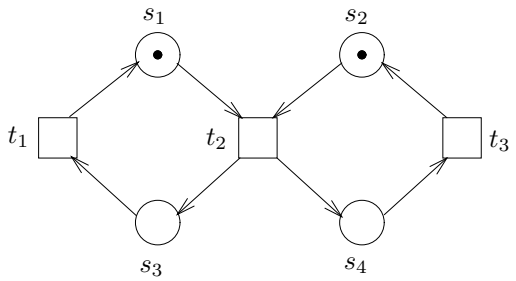


Figure 4.1

- *To Proposition 4.2.9:*

  Peterson's algorithm with an additional transition t satisfying ${}^\bullet t = \{p_4, q_4\}$ and $t^\bullet = \emptyset$. The Petri net is deadlock free, but the Marking Equation has a solution for $(m_1 = true, m_2 = true, hold = 1)$ that satisfies the conditions of Proposition 4.2.9 (Exercise: find a smaller example).

4.3 S- and T-invariants

4.3.1 S-invariants

Definition 4.3.1 (S-invariants)
Let $N = (S, T, F)$ be a net. An S-invariant of N is a vector $I : S \to \mathbb{Q}$ such that $I \cdot \mathbf{N} = 0$.

Proposition 4.3.2 (Fundamental property of S-invariants)
Let (N, M_0) be a Petri net and let I be a S-invariant of N. If $M_0 \xrightarrow{*} M$, then $I \cdot M = I \cdot M_0$.

Proof. We have $M_0 \xrightarrow{\sigma} M$ for some firing sequence σ. By the Marking Equation Lemma we get

$$M = M_0 + \mathbf{N} \cdot \vec{\sigma}$$

and so

$$\begin{array}{lll} I \cdot M & = I \cdot M_0 + I \cdot \mathbf{N} \cdot \vec{\sigma} & \text{(Marking Equation)} \\ & = I \cdot M_0 & (I \cdot \mathbf{N} = 0) \end{array}$$

□

The value of the expression $I \cdot M$ is therefore the same for every reachable marking M, and so it constitutes an invariant of (N, M_0).

Example 4.3.3 *We compute the S-invariants of the net of Figure 4.1*

The incidence matrix is:

	t_1	t_2	t_3
s_1	1	−1	0
s_2	0	−1	1
s_3	−1	1	0
s_4	0	1	−1

We compute the solutions of the system of equations

$$(i_1, i_2, i_3, i_4) \cdot \begin{pmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \\ -1 & 1 & 0 \\ 0 & 1 & -1 \end{pmatrix} = 0$$

The general form of the S-invariants is therefore (x, y, x, y) *with* $x, y \in \mathbb{Q}$

The following propositions are an immediate consequence of the definition of S-invariants:

Proposition 4.3.4 *The S-invariants of a net form a vector space over the real numbers.*

This definition of S-invariant is very suitable for machines, but not for humans, who can only solve very small systems of equations by hand. There is an equivalent definition which allows people to decide, even for nets with several dozens of places, if a given vector is an S-invariant.

Proposition 4.3.5 *I is an S-invariant of $N = (S, T, F)$ iff.* $\forall t \in T : \sum_{s \in {}^\bullet t} I(s) = \sum_{s \in t^\bullet} I(s)$.

Proof. $I \cdot \mathbf{N} = 0$ is equivalent to $I \cdot \mathbf{t} = 0$ for every transition t. So for every transition t we have: $I \cdot \mathbf{t} = \sum_{s \in t^\bullet} I(s) - \sum_{s \in {}^\bullet t} I(s)$. □

Example 4.3.6 *We show that $I = (1, 1, 2, 1)$ is an S-invariant of the net of Figure 4.2. The condition of Proposition 4.3.5 must hold for transitions t_1, t_2 und t_3.*

- *Transition t_1:* $I(s_1) + I(s_2) = I(s_3) = 2$.
- *Transition t_2:* $I(s_3) = I(s_1) + I(s_4) = 2$.
- *Transition t_3:* $I(s_3) = I(s_4) + I(s_2) = 2$.

With the help of S-invariants we can give sufficient conditions for boundedness and necessary conditions for liveness and for the reachability of a marking.

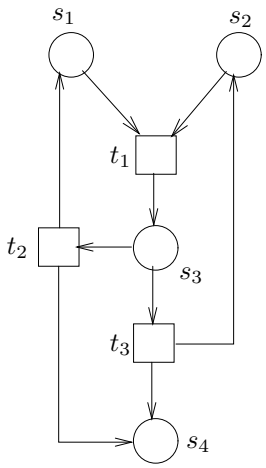


Figure 4.2

Definition 4.3.7 (Semi-positive and positive S-invariants)
Let I be an S-invariant of $N = (S, T, F)$. I is *semi-positive* if $I \geq 0$ and $I \neq 0$, and *positive* if $I > 0$ (that is, if $I(s) > 0$ for every $s \in S$). The *support* of an S-invariant is the set $\langle I \rangle = \{s \in S \mid I(s) > 0\}$.

Proposition 4.3.8 *[A sufficient condition for boundedness]*
Let (N, M_0) be a Petri net. If N has a positive S-invariant I, then (N, M_0) is bounded. More precisely: (N, M_0) is n-bounded for

$$n = max \left\{ \frac{I \cdot M_0}{I(s)} \mid s \text{ is a place of } N \right\}$$

Proof. Let M be any reachable marking. By the fundamental property of S-invariants we have $I \cdot M = I \cdot M_0$.
Let s be an arbitrary place of N. Since $I > 0$ we have $I(s) \cdot M(s) \leq I \cdot M = I \cdot M_0$ and $M(s) \leq \frac{I \cdot M_0}{I(s)}$. □

Proposition 4.3.9 *[A necessary condition for liveness]*
If (N, M_0) is live, then $I \cdot M_0 > 0$ for every semi-positive S-invariant of N.

Proof. Let I be a semi-positive S-invariant and let s be a place of $\langle I \rangle$. Since (N, M_0) is live, some reachable marking M satisfies $M(s) > 0$. Since I is semi-positive, we have $I \cdot M \geq I(s) \cdot M(s) > 0$. Since I is a S-invariant, we get $I \cdot M_0 = I \cdot M > 0$ □

These two propositions lead immediately to semi-algorithms for **Boundedness** and **Liveness**.

Definition 4.3.10 (The $\sim$ relation)
Let M and L be markings and let I be a S-invariant of a net N. M und L *agree on* I if $I \cdot M = I \cdot L$. We write $M \sim L$ if M and L agree on all invariants of N.

Proposition 4.3.11 *[A necessary condition for reachability]*
Let (N, M_0) be a Petri net. $M \sim M_0$ holds for every $M \in [M_0\rangle$.

Proof. Follows from the fundamental property of S-invariants. □

The following theorem allows one to decide if $M \sim L$ holds for two given markings M and L.

Theorem 4.3.12 *Let N be a net and let M, L be two markings of N.*
$M \sim L$ iff the equation $M = L + \mathbf{N} \cdot X$ has a rational solution.

Proof. ($\Rightarrow$): Since $M \sim L$, we have $I \cdot (L - M) = 0$ for every S-invariant I.

We now recall a well-known theorem of linear algebra. Given a $n \times m$ matrix A, let $U = \{u \in \mathbb{N}^n \mid u \cdot A = 0\}$, and let $V = \{v \in \mathbb{N}^m \mid u \cdot v = 0 \text{ for every } u \in U\}$. Then both U and V are vector spaces, and the columns of A contain a basis of V.

If we take $A := \mathbf{N}$, then U is the set of S-invariants of N, and so, by the theorem, the columns of $\mathbf{N}$ contain a basis of the vector space of vectors v satisfying $I \cdot v = 0$ for every S-invariant I. In particular, since $(L - M)$ is one of these vectors, $(L - M)$ is a linear combination over $\mathbb{Q}$ of the columns of $\mathbf{N}$, and so the equation $\mathbf{N} \cdot X = (L - M)$ has a rational solution.

($\Leftarrow$) : Let I be an S-invariant of N. Since $I \cdot \mathbf{N} = 0$ we have $I \cdot L = I \cdot M + I \cdot \mathbf{N} \cdot X = I \cdot M$. □

We also have the following consequences:

$$
\begin{array}{c}
M \text{ is reachable from } L \\
\not\Uparrow \quad \Downarrow \\
M = L + \mathbf{N} \cdot X \text{ has a solution } X \in \mathbb{N}^{|T|} \\
\not\Uparrow \quad \Downarrow \\
M = L + \mathbf{N} \cdot X \text{ has a solution } X \in \mathbb{Q}^{|T|} \\
\Updownarrow \\
M \sim L
\end{array}
$$

4.3.2 T-invariants

Definition 4.3.13 (T-invariants)
Let $N = (S, T, F)$ be a net. A vector $J : T \to \mathbb{Q}$ is a T-invariant of N if $\mathbf{N} \cdot J = 0$.

Proposition 4.3.14 *J is a T-invariant of $N = (S, T, F)$ iff $\forall s \in S : \sum_{t \in {}^\bullet s} J(t) = \sum_{t \in s^\bullet} J(t)$.*

Proposition 4.3.15 *[Fundamental property of T-invariants]*
Let N be a net, let M be a marking of N, and let σ be a sequence of transitions of N enabled at M. The vector $\vec{\sigma}$ is a T-invariant of N iff $M \xrightarrow{\sigma} M$.

Proof. ($\Rightarrow$) : Let M' be the marking satisfying $M \xrightarrow{\sigma} M'$. By the Marking Equation we have $M' = M + \mathbf{N} \cdot \vec{\sigma}$. Since $\mathbf{N} \cdot \vec{\sigma} = 0$ we get $M' = M$

($\Leftarrow$) : By the Marking Equation we have $M = M + \mathbf{N} \cdot \vec{\sigma}$ and so $\mathbf{N} \cdot \vec{\sigma} = 0$. □

Example 4.3.16 *We compute the T-invariants of the net of Figure 4.1 as the solutions of the system of equations*

$$\begin{pmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \\ -1 & 1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} j_1 \\ j_2 \\ j_3 \end{pmatrix} = 0$$

The general form of the T-invarints is (x, x, x), where $x \in \mathbb{Q}$.

Using T-invariants we obtain a necessary condition for well-formedness of a net:

Theorem 4.3.17 *[Necessary condition for well-formedness]*
Every well-formed net has a positive T-invariant.

Proof. Let N be a well-formed net and let M_0 be a live and bounded marking of N. By liveness there is an infinite firing sequence $\sigma_1\sigma_2\sigma_3\cdots$ such that every σ_i is a finite firing sequence containing all transitions of N. We have

$$M_0 \xrightarrow{\sigma_1} M_1 \xrightarrow{\sigma_2} M_2 \xrightarrow{\sigma_3} \dots$$

By boundedness there are indices $i < j$ such that $M_i = M_j$. So the sequence $\sigma_{i+1} \dots \sigma_j$ satisfies

$$M_i \xrightarrow{\sigma_{i+1} \dots \sigma_j} M_i$$

and so $J = \vec{\sigma}_{i+1} + \dots + \vec{\sigma}_j$ is a T-invariant of N. Further, J is positive because every transition occurs at least once in $\sigma_{i+1} \dots \sigma_j$. □

4.4 Siphons and Traps

4.4.1 Siphons

Definition 4.4.1 (Siphon)
Let $N = (S, T, F)$ be a net. A set $R \subseteq S$ of places is a *siphon* of N if ${}^\bullet R \subseteq R^\bullet$. A siphon R is *proper* if $R \neq \emptyset$.

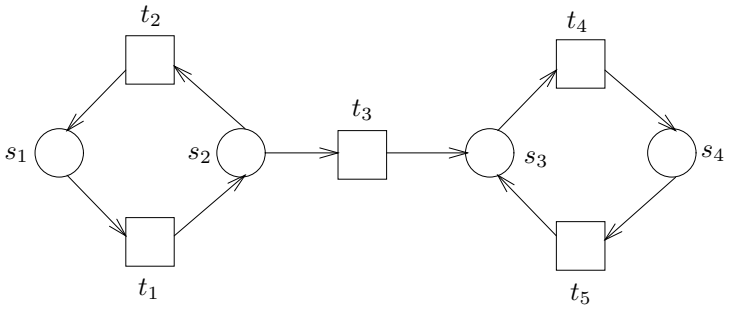


Figure 4.3

$\{s_1, s_2\}$ is a siphon of the net of Figure 4.3 because

$$^\bullet\{s_1, s_2\} = {}^\bullet s_1 \cup {}^\bullet s_2 = \{t_2\} \cup \{t_1\} = \{t_1, t_2\}$$

und

$$\{s_1, s_2\}^\bullet = s_1^\bullet \cup s_2^\bullet = \{t_1\} \cup \{t_2, t_3\} = \{t_1, t_2, t_3\}$$

Proposition 4.4.2 *[Fundamental property of siphons]*
Let R be a siphon of a net N, and let $M \xrightarrow{\sigma} M'$ be a firing sequence of N. If $M(R) = 0$, then $M'(R) = 0$.

Proof. Since $^\bullet R \subseteq R^\bullet$, the transitions that can mark R can only occur at markings that already mark R. □

Loosely speaking, a siphon that becomes unmarked (or "empty"), remains unmarked forever.

Corollary 4.4.3 *[A necessary condition for reachability]*
If M is reachable in (N, M_0), then for every siphon R, if $M_0(R) = 0$ then $M(R) = 0$.

We can easily check in polynomial time if this condition holds. For this we first observe that, if R_1 and R_2 are siphons of N, then so is $R_1 \cup R_2$ (exercise). It follows that there exists a unique largest siphon Q_0 unmarked at M_0 (more precisely, $R \subseteq Q_0$ for every siphon R such that $M_0(R) = 0$). We claim that the condition holds if and only if $M(Q_0) = 0$.

- If the condition holds, then, since $M_0(Q_0) = 0$ by definition, we get $M(Q_0) = 0$.
- If the condition does not hold, then there is a siphon R such that $M_0(R) = 0$ and $M(R) > 0$. Since $R \subseteq Q_0$, we also have $M(Q_0) > 0$.

The siphon Q_0 can be determined with the help of the following algorithm, which computes the largest siphon Q contained in a given set R of places—it suffices then to choose R as the set of places unmarked at M_0.

Input: A net $N = (S, T, F)$ and $R \subseteq S$.
Output: The largest siphon $Q \subseteq R$.
Initialization: $Q := R$.

begin
  while there are $s \in Q$ and $t \in {}^\bullet s$ such that $t \notin Q^\bullet$ **do**
    $Q := Q \setminus \{s\}$
  endwhile
end

Exercise: Show that the algorithm is correct. That is, prove that the algorithm terminates, and that after termination Q is the largest siphon contained in R.

Proposition 4.4.4 *[A necessary condition for liveness]*
If (N, M_0) is live, then M_0 marks every proper siphon of N.

Proof. Let R be a proper siphon of N and let $s \in R$. Since we assume that N is connected, ${}^\bullet s \cup s^\bullet \neq \emptyset$, and, since R is a siphon, $s^\bullet \neq \emptyset$. Let $t \in s^\bullet \neq \emptyset$. By liveness some reachable marking enables t, and so some reachable marking marks s, and therefore also the siphon R. By Proposition 4.4.3 the initial marking M_0 also marks R. □

Again, the condition can be checked with the help of the algorithm above: the condition holds if and only if $Q_0 = \emptyset$. We now look at deadlock-freedom. We can obtain a sufficient condition for it, but not one that is easy to check.

Proposition 4.4.5 *If M is a dead marking of N, then the set of places unmarked at M is a siphon of N.*

Proof. Let $R = \{s \mid M(s) = 0\}$. For every transition t there is a place $s \in {}^\bullet t$ such that $M(s) = 0$ (otherwise t would be enabled). So $R^\bullet$ contains all transitions of N, and therefore ${}^\bullet R \subseteq R^\bullet$. □

Corollary 4.4.6 *[A sufficient condition for deadlock-freedom] Let (N, M_0) be a Petri net. If every reachable marking marks all siphons of N, then (N, M_0) is deadlock-free.*

4.4.2 Traps

Definition 4.4.7 (Trap)
Let $N = (S, T, F)$ be a trap. A set $R \subseteq S$ of places is a *trap* if $R^\bullet \subseteq {}^\bullet R$. A trap R is *proper* if $R \neq \emptyset$.

$\{s_3, s_4\}$ is a trap of the net of Figure 4.3.

Proposition 4.4.8 *[Fundamental property of traps]*
Let R be a trap of a net N and let $M \xrightarrow{\sigma} M'$ be a firing sequence of N. If $M(R) > 0$, then $M'(R) > 0$.

Proof. Since ${}^\bullet R \subseteq {}^\bullet R$, transitions that take tokens from R put tokens in R. □

So, loosely speaking, marked traps stay marked. Notice, however, that this does not mean that the number of tokens of a trap cannot decrease. The number can go up or down, just not become 0.

Corollary 4.4.9 *[A necessary condition for reachability]*
If M is reachable in (N, M_0), then for every trap R, if $M_0(R) > 0$ then $M(R) > 0$.

As in the case of siphons, we can check in polynomial time if this condition holds. If R_1 and R_2 are traps of N, then so is $R_1 \cup R_2$ (exercise). So there exists a unique largest trap Q_0 marked at M_0 (more precisely, $R \subseteq Q_0$ for every trap R such that $M(R) > 0$). It is easy to see that the condition holds if and only if $M_0(Q_0) > 0$ (exercise).

To compute the largest trap unmarked at M, we can transform the algorithm that computes the largest siphon contained in a given set of places into an algorithm for computing the largest trap (exercise).

Recall that checking the sufficient condition for deadlock-freedom was computationally expensive, because of the form "for every reachable marking ...". Combining siphons and traps we obtain an easier-to-check condition.

Proposition 4.4.10 *[A sufficient condition for deadlock-freedom]*
Let (N, M_0) be a Petri net. If every proper siphon of N contains a trap marked at M_0, then (N, M_0) is deadlock-free.

Proof. Easy consequence of Corollary 4.4.6 and Proposition 4.4.8. □

The siphon-trap condition cannot be checked in polynomial time unless P=NP (whether every proper siphon contains a marked trap is an NP-complete problem), but can be checked with the help of a SAT-solver (see "New algorithms for deciding the siphon-trap property" by O. Oanea, H. Wimmel, and K. Wolf).

We finally show how to combine S-invariants and traps to prove that Peterson's algorithm satisfies the mutual exclusion property. For the Petri net model of Figure 2.8 mutual exclusion means that no reachable marking M satisfies $M(p_4) \geq 1 \wedge M(q_4) \geq 1$. We first compute three S-invariants:

(1) $M(hold = 1) + M(hold = 2) = 1$

(2) $M(p_2) + M(p_3) + M(p_4) + M(m_1 = f) = 1$

(3) $M(q_2) + M(q_3) + M(q_4) + M(m_1 = f) = 1$

and two constraints derived from traps:

(4) $M(m_1 = f) + M(p_2) + M(hold = 1) + M(q_3) > 0$

(5) $M(m_2 = f) + M(q_2) + M(hold = 2) + M(p_3) > 0$

Assume now $M(p_4) \geq 1 \wedge M(q_4) \geq 1$ holds. We have:

$$M(p_4) \geq 1 \wedge M(q_4) \geq 1$$

$$\Rightarrow \{(2), (3)\}$$

$$M(p_2) + M(p_3) + M(m_1 = f) = 0 \quad \wedge \quad M(q_2) + M(q_3) + M(m_2 = f) = 0$$

$$\Rightarrow \{(1)\}$$

$$\begin{array}{lcl} M(m_1 = f) + M(p_2) + & & M(m_2 = f) + M(q_2) + \\ M(hold = 1) + M(q_3) = 0 & \vee & M(hold = 2) + M(p_3) = 0 \end{array}$$

Contradicts (4) Contradicts (5)

Chapter 5

Petri net classes with efficient decision procedures

In the three sections of this chapter we study three classes of Petri nets: S-systems, T-systems, and free-choice systems. The sections have a similar structure. After the definition of the class, we introduce three theorems: the Liveness, Boundedness, and Reachability Theorem. The Liveness Theorem characterizes the live Petri nets in the class. The Boundedness Theorem characterizes the live and bounded systems. The Reachability Theorem characterizes the reachable markings of the live and bounded systems. The proof of the theorems requires some results about the structure of S- and T-invariants of the class, which we also present.

The theorems immediately yield decision procedures for **Liveness**, **Boundedness** and **Reachability** whose complexity is much lower than those for general Petri nets.

At the end of the section we present a final theorem, the Shortest Path Theorem, which gives an upper bound for the length of the shortest firing sequence leading to a given reachable marking.

The reader may ask why boundedness only for live Petri nets, and why reachability only for live and bounded Petri nets. A first reason is that, in many application areas, a Petri net model of a correct system must typically be live *and* bounded, and so, when one of these properties fails, it does not make much sense to further analyze the model. The second reason is that, interestingly, the general characterization of the bounded systems or the reachable markings is more complicated and less elegant than the corresponding characterization for live or live and bounded Petri nets.

The proofs of the theorems are very easy for S-systems, a bit more involved for T-systems, and relatively complex for free-choice systems. For this reason we just sketch the proofs for S-systems, explain the proofs in some detail for T-systems, and omit them for free-choice systems.

5.1 S-Systems

Definition 5.1.1 (S-nets, S-systems) A net $N = (S, T, F)$ is a *S-net* if $|{}^\bullet t| = 1 = |t^\bullet|$ for every transition $t \in T$. A Petri net (N, M_0) is a *S-system* if N if N is a S-net.

Proposition 5.1.2 (Fundamental property of S-systems)
Let (N, M_0) be a S-system with $N = (S, T, F)$. Then $M_0(S) = M(S)$ for every reachable marking M.

Proof. Every transition consumes one token and produces one token. □

Theorem 5.1.3 *[Liveness Theorem] A S-system (N, M_0) where $N = (S, T, F)$ is live iff N is strongly connected and $M_0(S) > 0$.*

Proof. (Sketch.)
($\Rightarrow$): If N is not strongly connected, then there is an arc (s, t) such that N has no path from t to s. For every marked place s' such that there is a path from s' to s, we fire the transitions of the path to bring the tokens in s' to s, and then fire t to empty s. We have then reached a marking from which no tokens can "travel" back to s, and so a marking from which t cannot occur again. So (N, M_0) is not live.

If M_0 marks no places, then no transition can occur, and (N, M_0) is not live.

($\Leftarrow$): If N is strongly connected and M_0 puts at least one token somewhere, then the token can freely move, reach any other place, and so enable any transition again. □

Theorem 5.1.4 *[Boundedness Theorem] A live S-system (N, M_0) where $N = (S, T, F)$ is b-bounded iff $M_0(S) \leq b$.*

Proof. Trivial. □

Exercise: give a counterexample for non-live S-systems.

Theorem 5.1.5 *[Reachability Theorem] Let (N, M_0) be a live S-system and let M be a marking of N. M is reachable from M_0 iff $M_0(S) = M(S)$.*

Proof. N is strongly connected by Proposition 5.1.3. So we are free to distribute the tokens of M_0 in an arbitrary way, and reach any marking M, as long as $M(S) = M_0(S)$. □

Proposition 5.1.6 *[S-invariants of S-nets] Let $N = (S, T, F)$ be a connected S-net. A vector $I : S \to \mathbb{Q}$ is a S-invariant of N iff $I = (x, \ldots, x)$ for some $x \in \mathbb{Q}$.*

Proof.

Each transition $t \in T$ has exactly one input place s_t and an output place s'_t. So we have

$$\sum_{s \in {}^\bullet t} I(s) = I(s_t) \quad \text{and} \quad \sum_{s \in t^\bullet} I(s) = I(s'_t)$$

and therefore

$$\begin{array}{ll} & I \text{ is a S-invariant} \\ \Leftrightarrow & \{\text{Proposition 4.3.5 (alternative definition of S-invariant)}\} \\ & \forall t \in T : I(s_t) = I(s'_t) \\ \Leftrightarrow & \{N \text{ is connected}\} \\ & \forall s_1, s_2 \in S : I(s_1) = I(s_2) \\ \Leftrightarrow & \{\,\} \\ & \exists x \in \mathbb{Q} \forall s \in S : I(s) = x. \end{array}$$

□

5.2 T-systems

Definition 5.2.1 (T-nets, T-systems) A net $N = (S; T, F)$ is a *T-net* if $|{}^\bullet s| = 1 = |s^\bullet|$ for every place $s \in S$. A system (N, M_0) is a T-system if N is a T-net.

Notation: Let γ be a circuit of a net N and let M be a marking of N. We denote by $M(\gamma)$ the number if tokens of γ under M, that is, $M(\gamma) = \sum_{s \in \gamma} M(s)$.

Proposition 5.2.2 (Fundamental property of T-systems) *Let γ be a circuit of a T-systems (N, M_0) and let M be a reachable marking. Then $M(\gamma) = M_0(\gamma)$.*

Proof. Firing a transition does not change the number of tokens of γ. If the transition does not belong to the circuit, then the distribution of tokens in the circuit does not change. If the transition belongs to the circuit, then it removes one token from a place of the circuit, and adds a token to another place. The token count does not change. □

5.2.1 Liveness

Theorem 5.2.3 *[Liveness Theorem] A T-system (N, M_0) is live iff $M_0(\gamma) > 0$ for every circuit γ of N.*

Proof.

($\Rightarrow$) Let γ be a circuit with $M_0(\gamma) = 0$. By Proposition 5.2.2 we have $M(\gamma) = 0$ for every reachable marking M. So no transition of γ can ever occur.

($\Leftarrow$) Let t be an arbitrary transition and let M be a reachable marking. We show that some marking reachable from M enables t. Let S_M be the set of places s of N

satisfying the following property: there is a path from s to t that contains no place marked at M. We proceed by induction on $|S_M|$. **Basis:** $|S_M| = 0$. Then $M(s) > 0$ for every place $s \in {}^\bullet t$, and so M enables t.
Step: $|S_M| > 0$. By the fundamental property of T-systems, every circuit of N is marked at M. So there is a path Π such that:

(1) Π leads to t;

(2) M marks no place of Π;

(3) Π has maximal length (that is, no path longer than Π satisfies (1) and (2)).

Let u be the first element of Π. By (3) u is a transition and M marks all places of ${}^\bullet u$. So M enables u. Moreover, we have $u \neq t$ because M does not enable t. Let $M \xrightarrow{u} M'$. We show that $S_{M'} \subset S_M$, and so that $|S_{M'}| < |S_M|$.

1. $S_{M'} \subseteq S_M$
   Let $s \in S_{M'}$. We show $s \in S_M$. There is a path $\Pi' = s \ldots t$ containing no place marked at M'. Assume Π' contains a place r marked at M. Since $M'(r) = 0$ and $M \xrightarrow{u} M'$ we have $u \in r^\bullet$ and so $\{u\} = r^\bullet$. So u is the successor of r in Π'. Since $u \neq t$, M' marks the successor of u in Π', contradicting the definition of Π'.

2. $S_{M'} \neq S_M$. Let s be the successor of u in Π. Then $s \in S_M$ but $s \notin S_{M'}$, because $M'(s) > 0$.

By induction hypothesis there is a firing sequence $M' \xrightarrow{\sigma} M''$ such that M'' enables t. It follows $M \xrightarrow{u} M' \xrightarrow{\sigma} M''$, and so M'' is a marking reachable from M that enables t. □

5.2.2 Boundedness

Theorem 5.2.4 *[Boundedness Theorem] A place s of a live T-system (N, M_0) is b-bounded iff it belongs to some circuit γ such that $M_0(\gamma) \leq b$.*

Proof. ($\Leftarrow$) Follows from the fundamental property of T-systems (Proposition 5.2.2).
($\Rightarrow$) Let M be a reachable marking such that $M(s)$ is maximal. We have $M(s) \leq b$. Define the marking L as follows:

$$L(r) = \begin{cases} M(r) & \text{if } r \neq s \\ 0 & \text{if } r = s \end{cases}$$

We claim that (N, L) is not live. Otherwise there would be a firing sequence $L \xrightarrow{\sigma} L'$ such that $L'(s) > 0$, and by the Monotonicity Lemma we would have $M \xrightarrow{\sigma} M'$ for some marking M' satisfying $M'(s) = L'(s) + M(s) > M(s)$, contradicting the maximality of $M(s)$. By the Liveness Theorem some circuit γ is unmarked at L but marked

at M. Since L and M only differ in the place s, the circuit γ contains s. Further, s is the only place of γ marked at M. So $M(\gamma) = M(s)$, and since $M(s) \leq b$ we get $M(\gamma) \leq b$. □

Corollary 5.2.5 *Let (N, M_0) be a live T-system*

1. *A place of N is bounded iff it belongs to some circuit.*
2. *Let s be a bounded place. Then*

$$max\{M(s) \mid M_0 \xrightarrow{*} M\} = min\{M_0(\gamma) \mid \gamma \text{ contains } s\}$$

3. *(N, M_0) is bounded iff N is strongly connected.*

Proof. Exercise □

5.2.3 Reachability

We need to have a closer look at the T-invariants of T-systems.

Proposition 5.2.6 *[T-invariants of T-nets] Let $N = (S, T, F)$ be a connected T-net. A vector $J : T \to \mathbb{Q}$ is a T-invariant iff $J = (x \dots x)$ for some $x \in \mathbb{Q}$.*

Proof. Dual of the proof of Proposition 5.1.6. □

Theorem 5.2.7 *[Reachability Theorem] Let (N, M_0) be a live T-system. A marking M is reachable from M_0 iff $M_0 \sim M$.*

Proof. ($\Rightarrow$) Proposition 4.3.11
($\Leftarrow$) By Theorem 4.3.12 there is a rational vector X such that

$$M = M_0 + \mathbf{N}.X \tag{5.1}$$

The vector $J = (1, 1, \dots, 1)$ is a T-invariant of N (Proposition 5.2.6). So we have

$$\mathbf{N} \cdot (X + \lambda J) = \mathbf{N} \cdot X$$

for every $\lambda \in \mathbb{Q}$. So without loss of generality we can assume $X \geq 0$.
Let T be the set of transitions of N. We show:

(1) There is a vector $Y : T \to \mathbb{N}$ such that $M = M_0 + \mathbf{N} \cdot Y$. Let Y be the vector with $Y(t) = \lceil X(t) \rceil$ for every transition t ($\lceil x \rceil$ denotes the smallest integer larger than or equal to x). By (5.1) we have

$$M(s) = M_0(s) + X(t_1) - X(t_2)$$

for every place s, where $\{t_1\} = {}^\bullet s$ and $\{t_2\} = s^\bullet$. Both $M(s)$ and $M_0(s)$ are integers. By the definition of Y we get

$$X(t_1) - X(t_2) = Y(t_1) - Y(t_2)$$

So $M(s) = M_0 + Y(t_1) - Y(t_2)$, which implies $M = M_0 + \mathbf{N} \cdot Y$.

(2) $M_0 \xrightarrow{*} M$
By induction over $|Y| = \sum_{t \in T} Y(t)$.
Basis: $|Y| = 0$. Then $Y = 0$ and $M = M_0$.
Step: $|Y| > 0$.
We show that M_0 enables some transition of $\langle Y \rangle$. Let

$$Sy = \{s \in {}^\bullet\langle Y \rangle \mid M_0(s) = 0\}$$

Let $s \in Sy$. By $M_0(s) = 0$ and $M_0 + \mathbf{N} \cdot Y = M \geq 0$ we have:

> if some transition of $s^\bullet$ belongs to $\langle Y \rangle$, then some transition of ${}^\bullet s$ belongs to $\langle Y \rangle$. (*)

Let Π be a path of maximal length containing places of *Sy* and transitions of $\langle Y \rangle$ (such a path exists, because otherwise N would contain a circuit unmarked at M_0). By (*), the first node of Π is a transition $t \in \langle Y \rangle$, and no place of ${}^\bullet t$ belongs to *Sy*. So M_0 marks every place of ${}^\bullet t$, that is, M_0 enables t.

Let $M_0 \xrightarrow{t} M_1$. We have

$$M_1 + \mathbf{N}(Y - \vec{t}) = M$$

where

$$|Y - \vec{t}| = |Y| - 1 < |Y|$$

By induction hypothesis we have $M_1 \xrightarrow{*} M$. Since $M_0 \xrightarrow{t} M_1 \xrightarrow{*} M$, we get $M_0 \xrightarrow{*} M$.

□

5.2.4 Other properties

The theorems we have introduced have many interesting consequences. Her are two of them.

Theorem 5.2.8 *Let N be a strongly connected T-net. For every marking M_0 the following statements are equivalent:*

(1) (N, M_0) is live.

(2) (N, M_0) *is deadlock-free.*

(3) (N, M_0) *has an infinite firing sequence.*

Proof. (1) $\Rightarrow$ (2) $\Rightarrow$ (3) follow immediately from the definitions. We show (3) $\Rightarrow$ (1).

Let $M_0 \xrightarrow{\sigma}$ be an infinite firing sequence. We claim that every transition of N occurs in σ. Since N is strongly connected, (N, M_0) is bounded (Theorem 5.2.4). Let $\sigma = t_1\ t_2\ t_3 \ldots$, and $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \xrightarrow{t_3} \ldots$. Since (N, M_0) is bounded, there are indices i and j with $i < j$ such that $M_i = M_j$. Let σ_{ij} be the subsequence of σ containing the transitions between M_i and M_j. By the fundamental property of T-invariants (Proposition 4.3.15) $\vec{\sigma_{ij}}$ is a T-Invariant . By Proposition 5.2.6 there is $n \in \mathbb{N}$ such that $\vec{\sigma_{ij}} = (n \ldots n)$. So every transition of N occurs in σ_{ij}, and so the same holds for σ.

Since every transition of N occurs in σ, for every place and every circuit of N some marking reached during the execution of N marks the place or the circuit. By the fundamental property of T-systems, all circuits of N are marked at M_0. By the Liveness Theorem (Theorem 5.2.3), (N, M_0) is live. □

Theorem 5.2.9 *[Genrich's Theorem] Let N be a strongly connected T-net with at least one place and one transition. There is a marking M_0 such that (N, M_0) is live and 1-bounded.*

Proof. Since N is strongly connected, any marking that puts tokens on all places of N is live, because it marks all circuits (Liveness Theorem), and bounded, because all markings of N are (Corollary 5.2.5).

Let (N, M) be live and bounded, but not 1-bounded. We construct another live marking L of N satisfying the following two conditions:

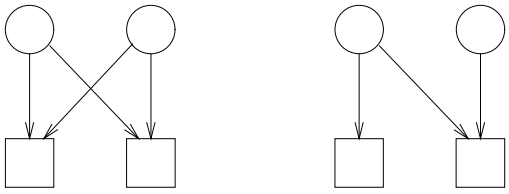
(1) $L(\gamma) \leq M(\gamma)$ for every circuit γ of N, and

(2) $L(\gamma) < M(\gamma)$ for at least one circuit γ.

By Theorem 5.2.4, at least one place of N has a smaller bound under L as under M. Iterating this construction we obtain a 1-bounded marking of N.

Let s be a non-1-bounded place of (N, M). Some reachable marking M' satisfies $M'(s) \geq 2$. Let L be the marking that puts exactly one token in s, and as many tokens as M elsewhere.

Since M is live, it marks all circuits of N. By construction L also marks all circuits, and so L is also live. Condition (1) is a consequence of the definition of L. Condition (2) holds for all circuits containing s (and there is at least one, because N is strongly connected). □



free-choice not free-choice

Figure 5.1

5.3 Free-Choice Systems

Definition 5.3.1 (Free-Choice nets, Free-Choice systems) A net $N = (S, T, F)$ is *free-choice* if $s^\bullet \times {}^\bullet t \subseteq F$ for every $s \in S$ and $t \in T$ such that $(s, t) \in F$. A Petri net (N, M_0) is *free-choice* if N is a free-choice net..

This definition is very concise and moreover symmetric with respect to places and transitions. If the reader finds it cryptic, the following equivalent definitions may help.

Proposition 5.3.2 *[Alternative definitions of free-choice nets]*

(1) A net is free-choice if for every two transitions t_1, t_2:

$$(t_1 \neq t_2 \wedge {}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset) \Rightarrow {}^\bullet t_1 = {}^\bullet t_2$$

(2) A net is free-choice if for every two places s_1, s_2:

$$(s_1 \neq s_2 \wedge s_1^\bullet \cap s_2^\bullet \neq \emptyset) \Rightarrow s_1^\bullet = s_2^\bullet$$

Proof. Exercise. □

Figure 5.1 illustrates these definitions.
Clearly, S- and T-systems are special cases of free-choice systems (see Figure 5.2).

5.3.1 Liveness

We showed in the last chapter that a Petri net in which every siphon contains an initially marked trap is deadlock-free, but the converse does not hold. For free-choice systems we obtain Commoner's Theorem, a much stronger result characterizing liveness.

Theorem 5.3.3 *[Commoner's Liveness Theorem] A free-choice system (N, M_0) is live iff every siphon of N contains a trap marked at M_0.*

Proof. We sketch the following direction: if every siphon of N contains a trap marked at M_0, then (N, M_0) is live. We need the following definitions. Let M be a

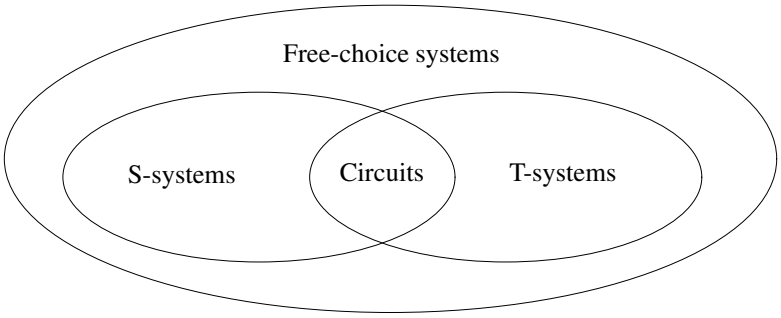


Figure 5.2: Net classes

marking of N. A transition t is *dead at* M if it is not enabled at any marking of $[M\rangle$. Let D_M denote the set of transitions dead at M. A transition t is *live at* M if $t \notin D_{M'}$ for every marking $M' \in [M\rangle$. Let L_M be the set of transitions live at M. Notice that a transition may be neither live nor dead at a marking. We have

- If $t \in L_M$ and $M' \in [M\rangle$, then $t \in L_{M'}$, that is, live transitions stay live.
- If $t \in D_M$ and $M' \in [M\rangle$, then $t \in L_{M'}$, that is, dead transitions stay dead.
- If $t \notin L_M \cup D_M$ then there is a marking M' reachable from M such that $t \in D_{M'}$. That is, transitions that are neither live nor dead may die.

Let T be the set of transitions of N. By the definitions above, there is a marking M reachable from M_0 such that $T = D_M \cup L_M$, that is, every transition is either live or dead at M. For every $t \in D_M$ there is a place $s_t \in {}^\bullet t$ such that $M(s_t) = 0$. Since N is free-choice, we have (exercise: prove it!): no input transition of s_t is live a M, that is, ${}^\bullet s_t \subseteq D_M$. It follows: the set $R = \{s_t \mid t \in T\}$ is a siphon unmarked at M. But then R does not contain any initially marked trap, because marked traps stay marked. □

A siphon is *minimal* if it does not properly contain any proper siphon. Clearly, the Liveness Theorem still holds if we replace "siphon" by "minimal siphon". The net of Figure 5.3 has four minimal siphons: $R_1 = \{s_1, s_3, s_5, s_7\}$, $R_2 = \{s_2, s_4, s_6, s_8\}$, $R_3 = \{s_2, s_3, s_5, s_7\}$ and $R_4 = \{s_1, s_4, s_6, s_8\}$. R_1, R_2, R_3 and R_4 are also traps, and so, in particular, they contain traps. By the Liveness Theorem, every marking that marks R_1, R_2, R_3 and R_4 is live.

The liveness problem for free-choice systems is NP-complete, and so we cannot expect to find a polynomial algorithm to check the condition of Commoner's Theorem:

Theorem 5.3.4 *[Complexity] The problem*

Given: A free-choice system (N, M_0)
Decide: Is (N, M_0) *not live?*

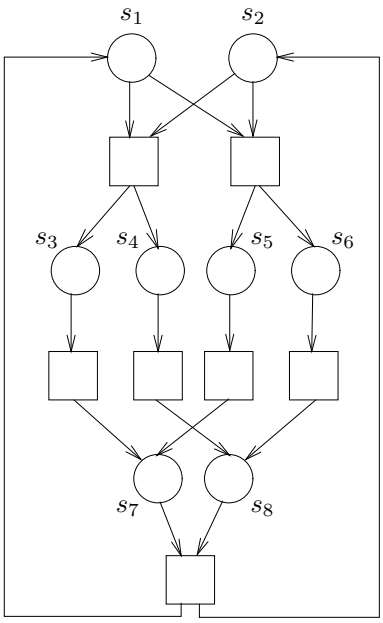


Figure 5.3: A free-choice system

is NP-complete.

Proof. The proof is by reduction from SAT, the satisfiability problem for boolean formulas. The reduction is illustrated in Figure 5.4, which shows the free-choice system for the formula

$$\Phi = (x_1 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee \overline{x_3})$$

□

5.3.2 Boundedness

Definition 5.3.5 (S-component) Let $N = (S, T, F)$ be a net. A subnet $N' = (S', T', F')$ of N is an *S-component* of N if

1. $T' = {}^\bullet S' \cup S'^\bullet$ (where ${}^\bullet s = \{t \in T \mid (t, s) \in F\}$, and analogously for $s^\bullet$).

2. N' is a strongly connected S-net.

Figure 5.5 shows two S-components of the net of Figure 5.3.

S-components are for free-choice systems what circuits are for T-systems: firing a transition does not change the number of tokens of an S-component.

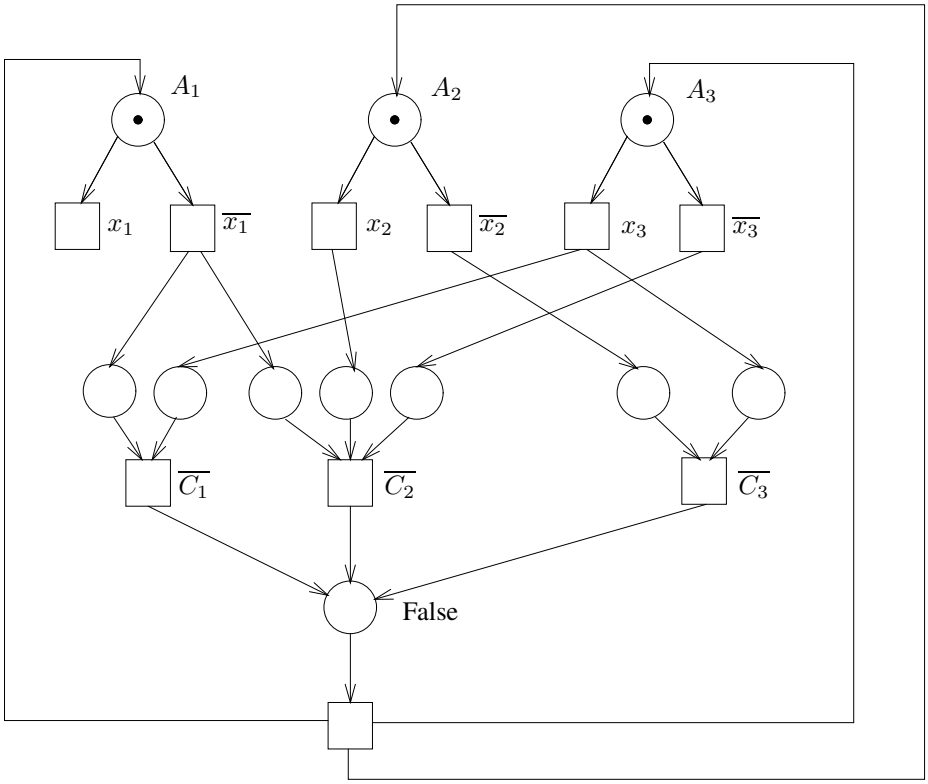


Figure 5.4: Free-choice system for the formula Φ

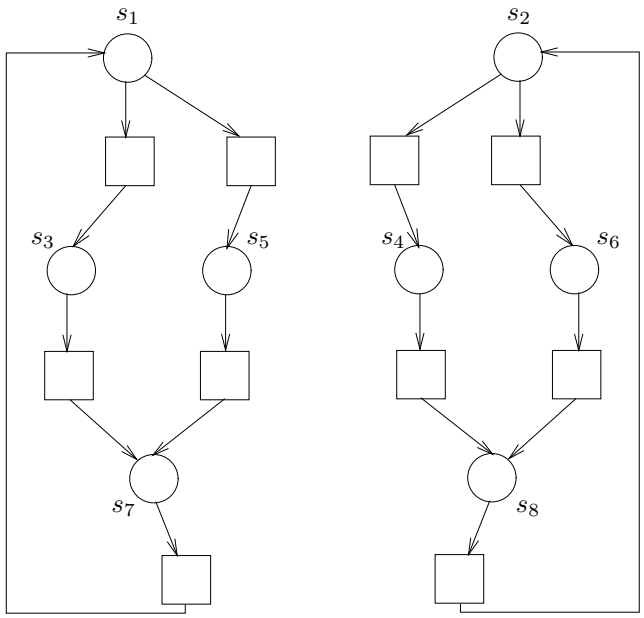


Figure 5.5: S-components of the net of Figure 5.3

Proposition 5.3.6 *Let (N, M_0) be a Petri net and let $N' = (S', T', F')$ be an S-component of N. Then $M_0(S') = M(S')$ for every marking M reachable from M_0.*

Proof. Firing a transition either takes no tokens from a place of the component and adds none, or it takes exactly one token and adds exactly one token. □

Theorem 5.3.7 *[Hack's Boundedness Theorem] Let (N, M_0) be a live free-choice system. (N, M_0) is bounded iff every place of N belongs to a S-component.*

Proof. ($\Leftarrow$) Exercise

($\Rightarrow$) (Sketch). We first show that every minimal siphon N is the set of places of a S-component. Then we show that every place is contained in some minimal siphon. □

Proposition 5.3.8 *[Place bounds] Let (N, M_0) be a live and bounded free-choice system and let s be a place of N. We have*

$$max\{M(s) \mid M_0 \xrightarrow{*} M\} =$$
$$min\{M_0(S') \mid S' \text{ is the set of places of a S-component of } N\}$$

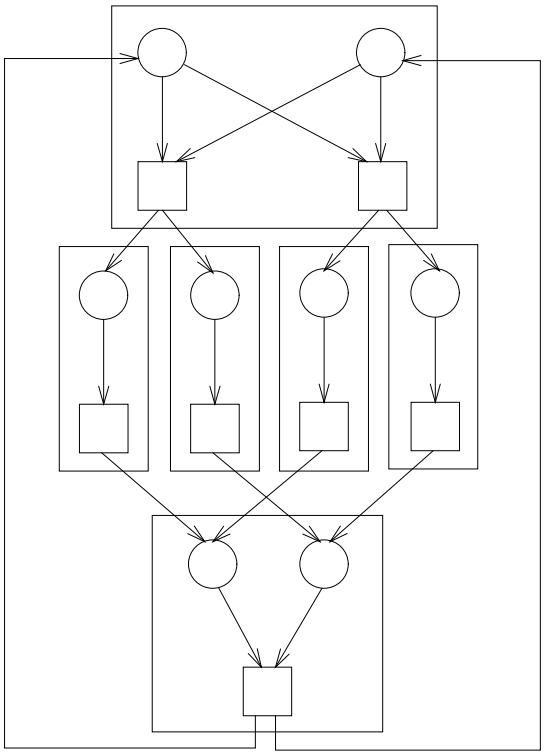


Figure 5.6: Clusters of the net of Figure 5.3

Proof. Analogous to the Boundedness Theorem for T-systems. □

Theorem 5.3.4 shows that there is no polynomial algorithm for **Liveness** (unless $P = NP$). Now we ask ourselves what is the complexity of deciding if a free-choice system is simultaneously live and bounded. We can of course first use the decision procedure for liveness, and then, if the net is live, check the condition of the Boundedness Theorem. But there are more efficient algorithms.[1]. The fastest known algorithm runs in $O(n \cdot m)$ time for a net with n places and m transitions. A not so efficient but simpler algorithm follows immediately from the next theorem:

Definition 5.3.9 (Cluster) Let $N = (S, T, F)$ be a net. A *cluster* is an equivalence class of the equivalence relation $((F \cap (S \times T)) \cup (F \cap (S \times T))^{-1})^*$.

Figure 5.6 shows the clusters of the net of Figure 5.3.

[1]Compare with this: in order to decide if a number is divisible by 100.000, we can first check if it is divisible by 3125, and, if so, if it is divisible by 32. However, there is a faster procedure: check if the last five digits are zeros.

Theorem 5.3.10 *[The Rank Theorem] A free-choice system (N, M_0) is live and bounded iff*

1. *N has a positive S-invariant.*
2. *N has a positive T-invariant.*
3. *The rank of the incidence matrix $(\mathbf{N})$ is equal to $c - 1$, where c is the number of clusters of N.*
4. *Every siphon of N is marked under M_0.*

Proof. Omitted. □

Conditions (1) and (2) can be checked using linear programming, condition (3) using well-known algorithms of linear algebra, and condition (4) with the algorithm of Section 4.4.1.

5.3.3 Reachability

Theorem 5.3.11 Reachability *is NP-complete for live and bounded free-choice nets.*

Proof. Exercise: find a reduction from SAT. □

However, for systems satisfying an additional condition there is a polynomial algorithm. A Petri net (N, M_0) is *cyclic* if, loosely speaking, it is always possible to return to the initial marking. Formally: $\forall M \in [M_0\rangle : M_0 \in [M\rangle$. We have:

Theorem 5.3.12 *[Reachability Theorem] Let (N, M_0) be a live, bounded, and cyclic free-choice system. A marking M of N is reachable from M_0 iff $M_0 \sim M$.*

Proof. Omitted. □

Corollary 5.3.13 *The problem*

> *Given: a live, bounded, and cyclic free-choice system (N, M_0) and a marking M*
> *Decide: Is M reachable?*

can be solved in polynomial time.

This result is only useful if we are able to check efficiently if a live and bounded free-choice system is cyclic. The following theorem shows that this is the case:

Theorem 5.3.14 *A live and bounded free-choice system (N, M_0) is cyclic iff M_0 marks every proper trap of N.*

Proof. Omitted. □