Low level Petri nets

Miklós Gerzson

Introduction

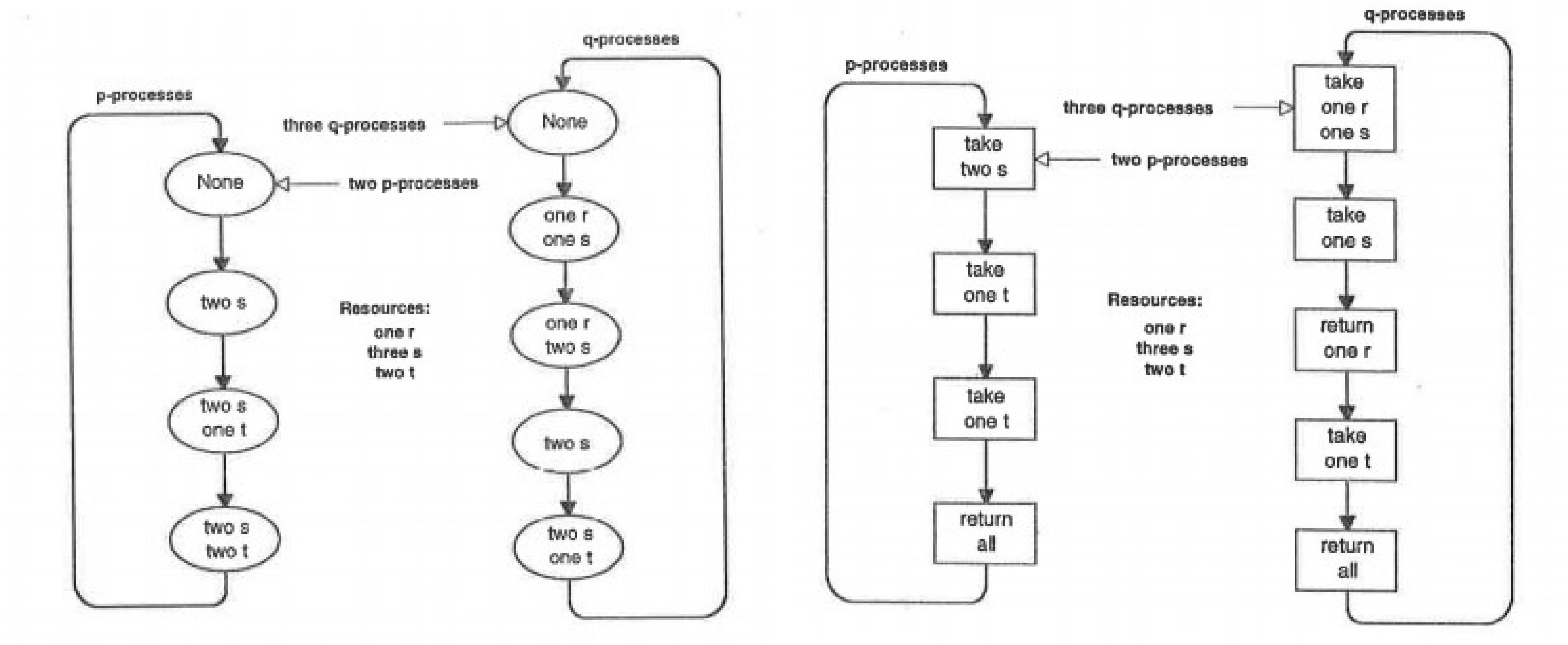
- Petri nets: graphical and mathematical modelling tool for the description of dynamic systems
- system types: concurrent, asynchronous, distributed, parallel, nondeterministic, stochastic
- graphical representation: structural description and dynamic characterization
- mathematical description: state equations, algebraic equations
- analysis tool: behavioral and structural features of systems

History

- Carl Adam Petri: Kommunikation mit Automaten, PhD Dissertation, 1962,
- very popular modelling tool
- conference series: International Conferences on Application and Theory of petri Nets and Concurrency, e.g.
- papers: Petri Net Newsletter, e.g.
- software: HPSim, CPN Tools, e.g.

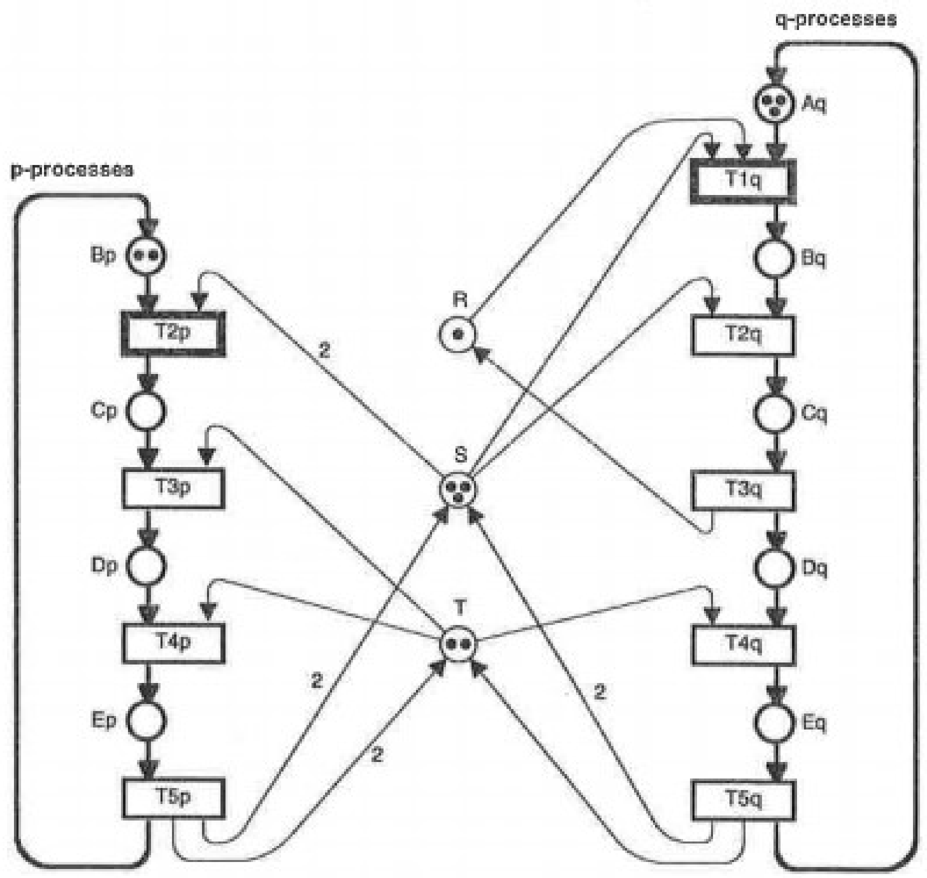
Introductory example

- Problem description



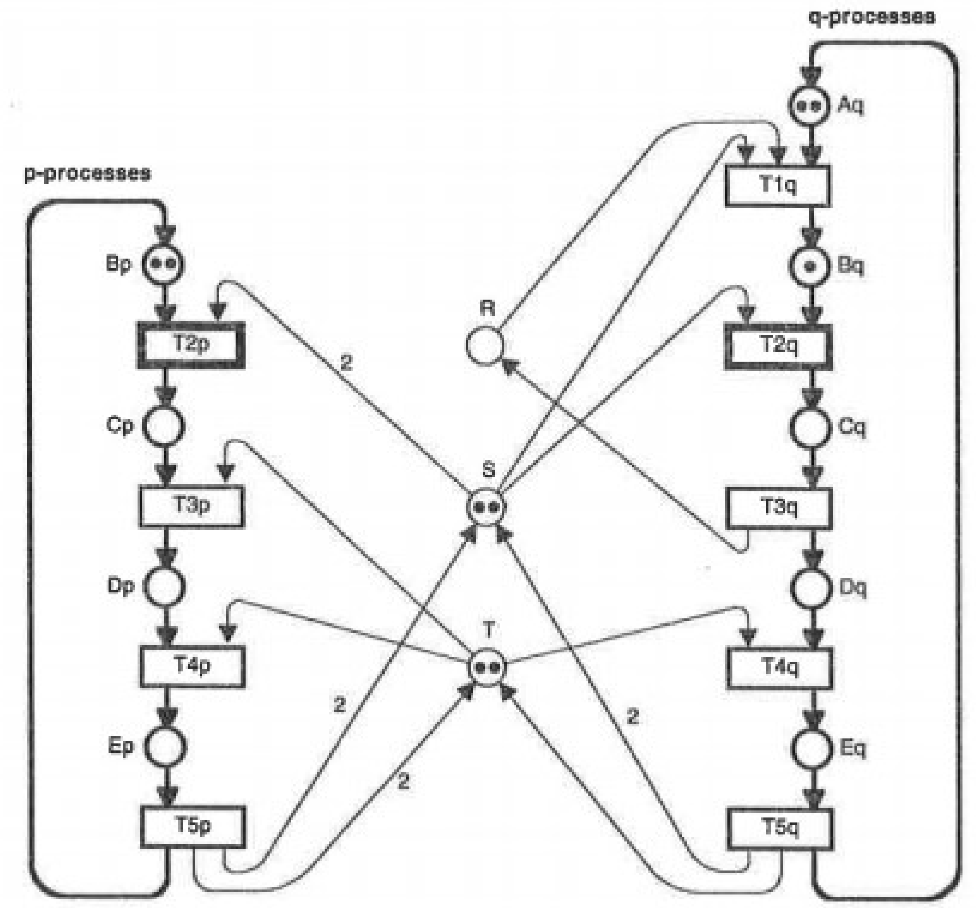
Introductory example

- Petri net of example



Introductory example

- Firing of a transition



Basic definitions

- Petri nets are the abstract models of information flow in the form of directed graph
- two types of elements:
  - transitions – occurring events
  - places – pre- and postconditions
- graphical representation
  - transitions: bars or boxes
  - places: circles
  - logical connections: arcs

Examples

Input places	**Transitions**	**Output places**
Preconditions	**Events**	**Postconditions**
input data	computation step	output data
input signals	signal processor	output signals
resources needed	task or job	resources released
conditions	clause in logic	conclusion(s)
buffers	processor	buffers

Dynamic behavior

- tokens on places: description of the state of a place
  - availability of a given resource
  - number of data
  - number of resources
- representation of tokens: black dots
- marking vector: defines the number of tokens on places
- weight function: assigns weight (positive integers) to the arcs

Formal definition

- Petri net is a 5-tuple,

$$PN = (P, T, F, W, M_0)$$

where

$P = \{p_1, p_2, ..., p_m\}$ – set of places;

$T = \{t_1, t_2, ..., t_n\}$ – set of transitions;

$F \subseteq (P \times T) \cup (T \times P)$ – set of arcs;

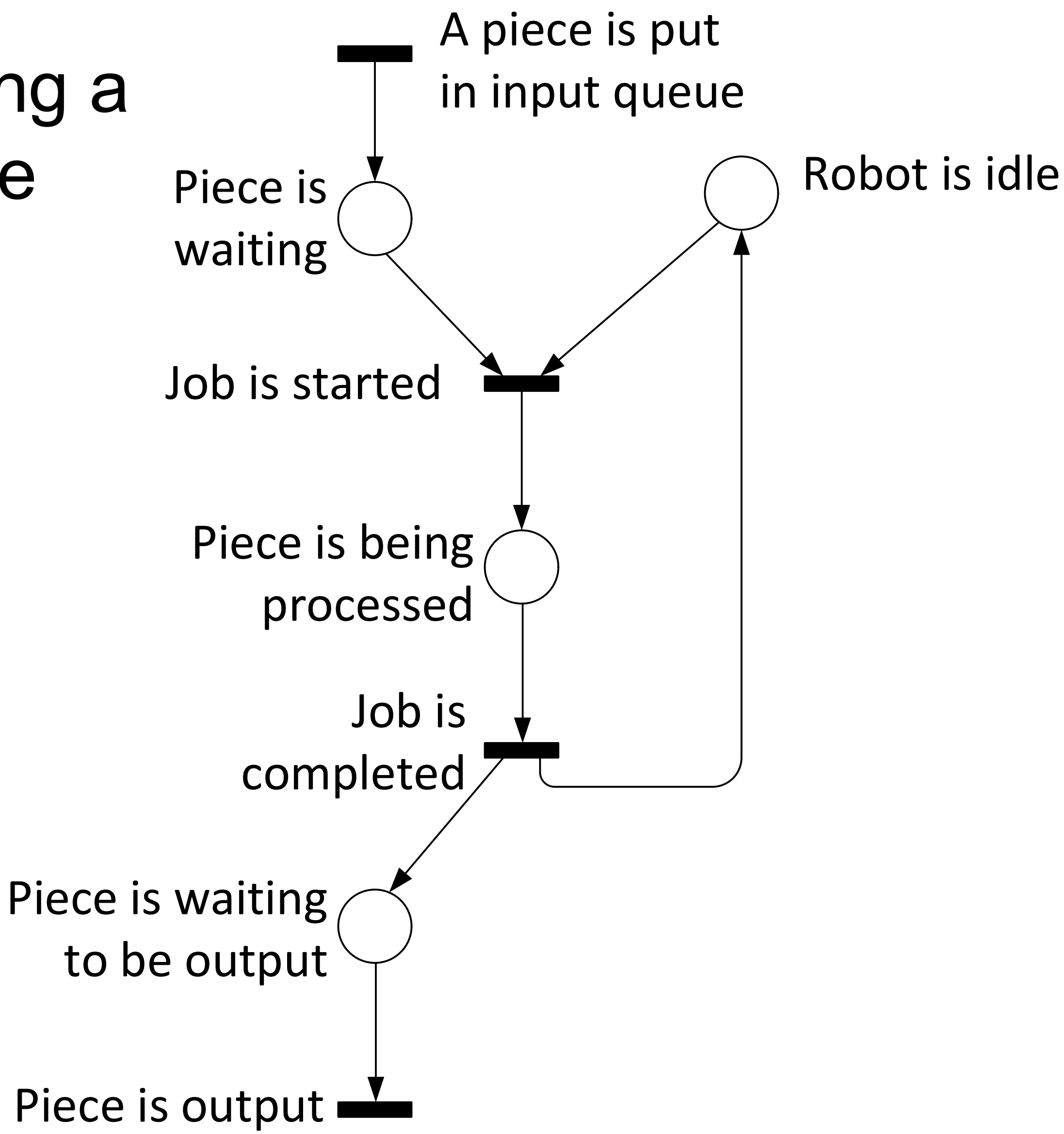
$W: F \rightarrow \{0, 1, 2, 3, ...\}$ – weight function;

$M_0: P \rightarrow \{0, 1, 2, 3, ...\}$ – initial marking.

$P \cap T = \emptyset$ and $P \cup T \neq \emptyset$

Example

- Processing a workpiece



Formal description of the example

$$PN = (P, T, F, W, M_0)$$

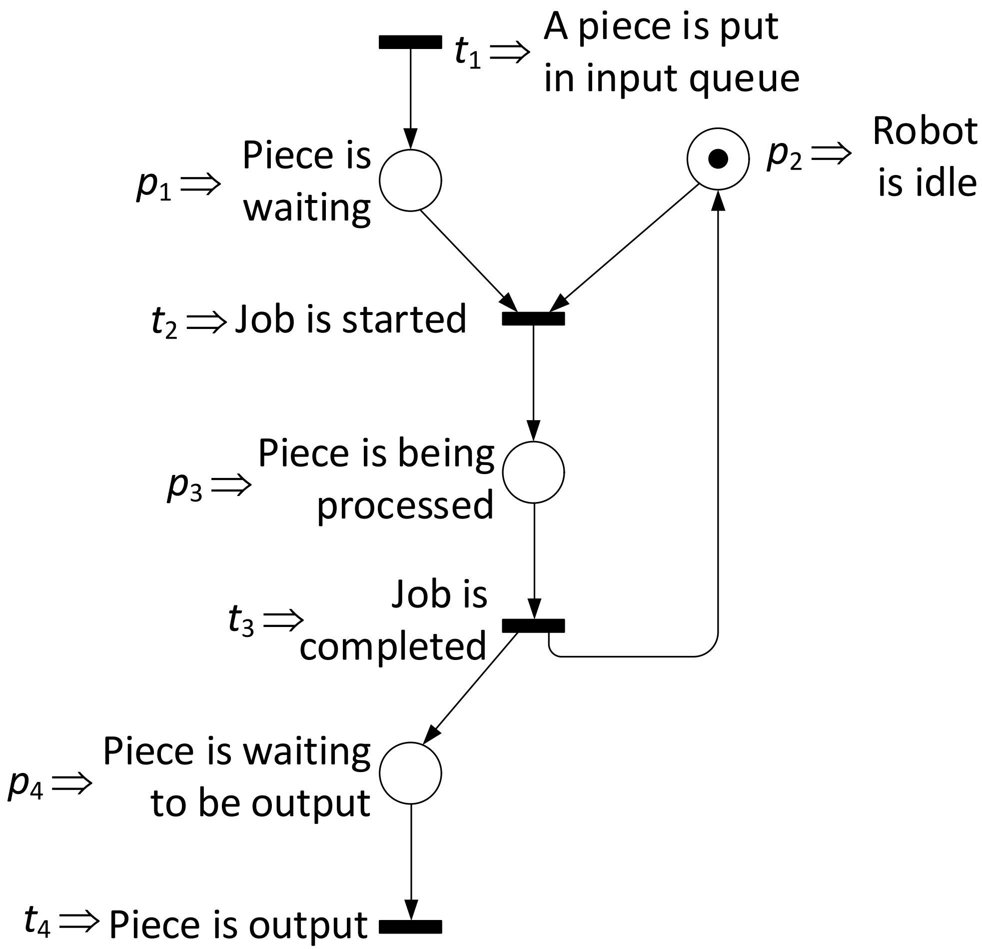
$$P = \{p_1, p_2, p_3, p_4\}$$

$$T = \{t_1, t_2, t_2, t_3\}$$

$$F = \{(t_1, p_1), (p_1, t_2), (p_2, t_2), (t_2, p_3), (p_3, t_3), (t_3, p_2), (t_3, p_4), (p_4, t_4)\}$$

$$W: w(t_1, p_1) = 1, w(p_1, t_2) = 1, w(p_2, t_2) = 1, \ldots w(p_4, t_4) = 1.$$

$$M_0: P \to \{0, 1, 0, 0\}$$



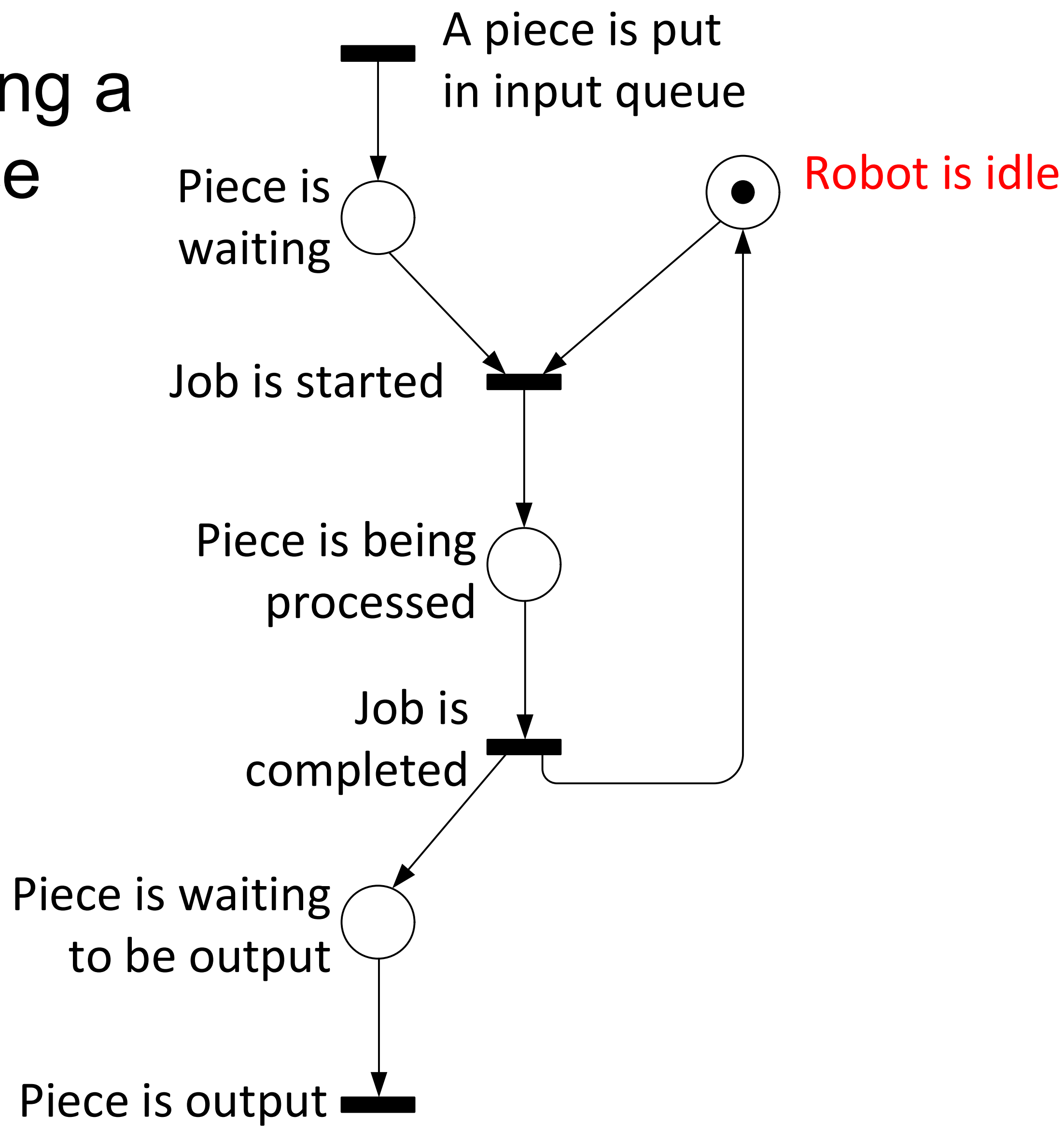
Firing of transitions

- Behavior of systems: state of their elements and changes in the system
- Simulation of changes: firing rules

1. A transition *t* is enabled if each of its input place is marked with at least *w*(*p*, *t*) tokens (*w*(*p*, *t*) is the weight of the arc from the place *p* to the transition *t*).
2. An enabled transition may or may not fire.
3. A firing of an enabled transition *t* removes *w*(*p*, *t*) tokens from each of its input place and adds *w*(*t*, *p*) tokens to its output places (*w*(*t*, *p*) is the weight of the arc from the transition *t* to the place *p*).

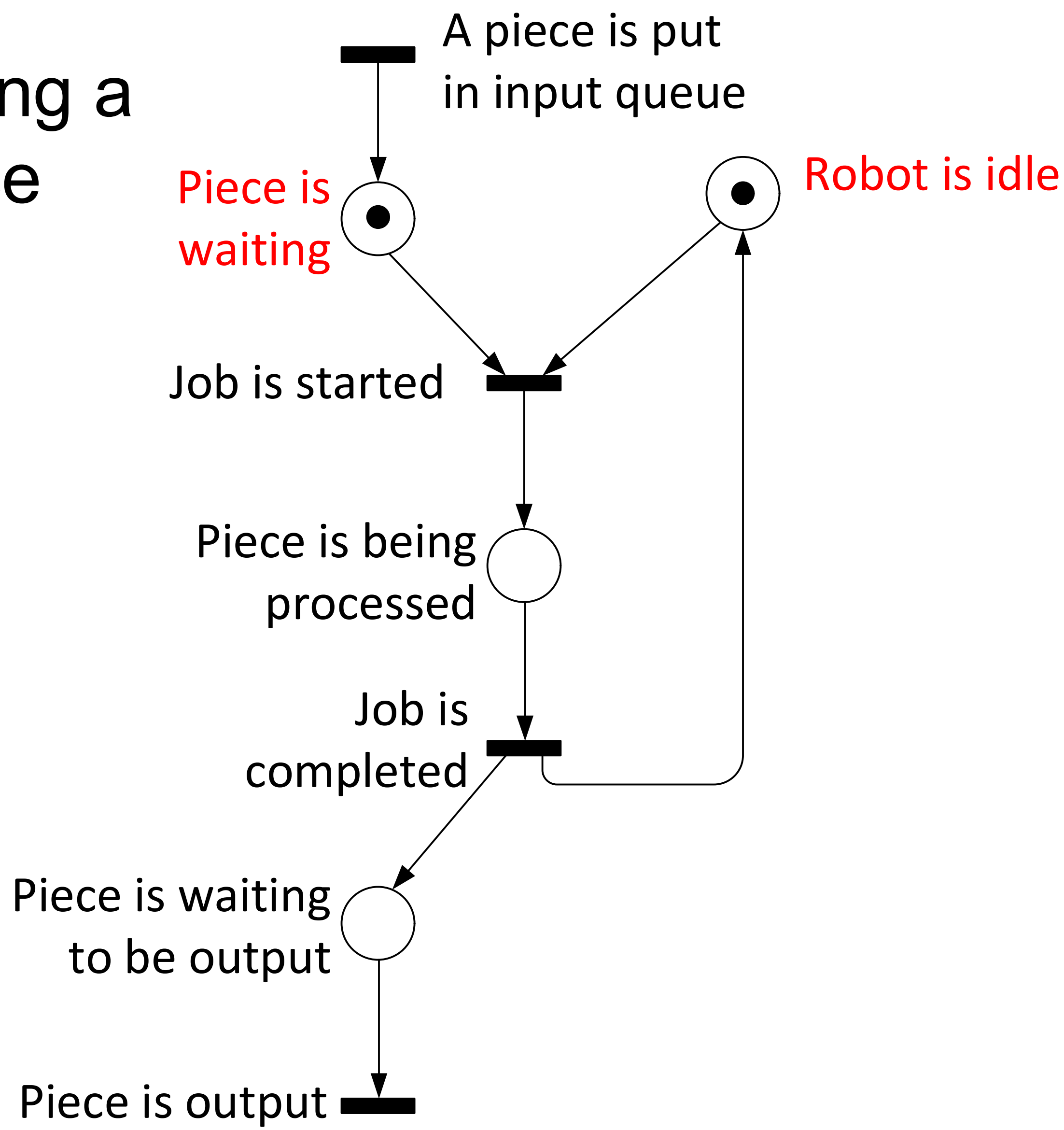
Example

- Processing a workpiece



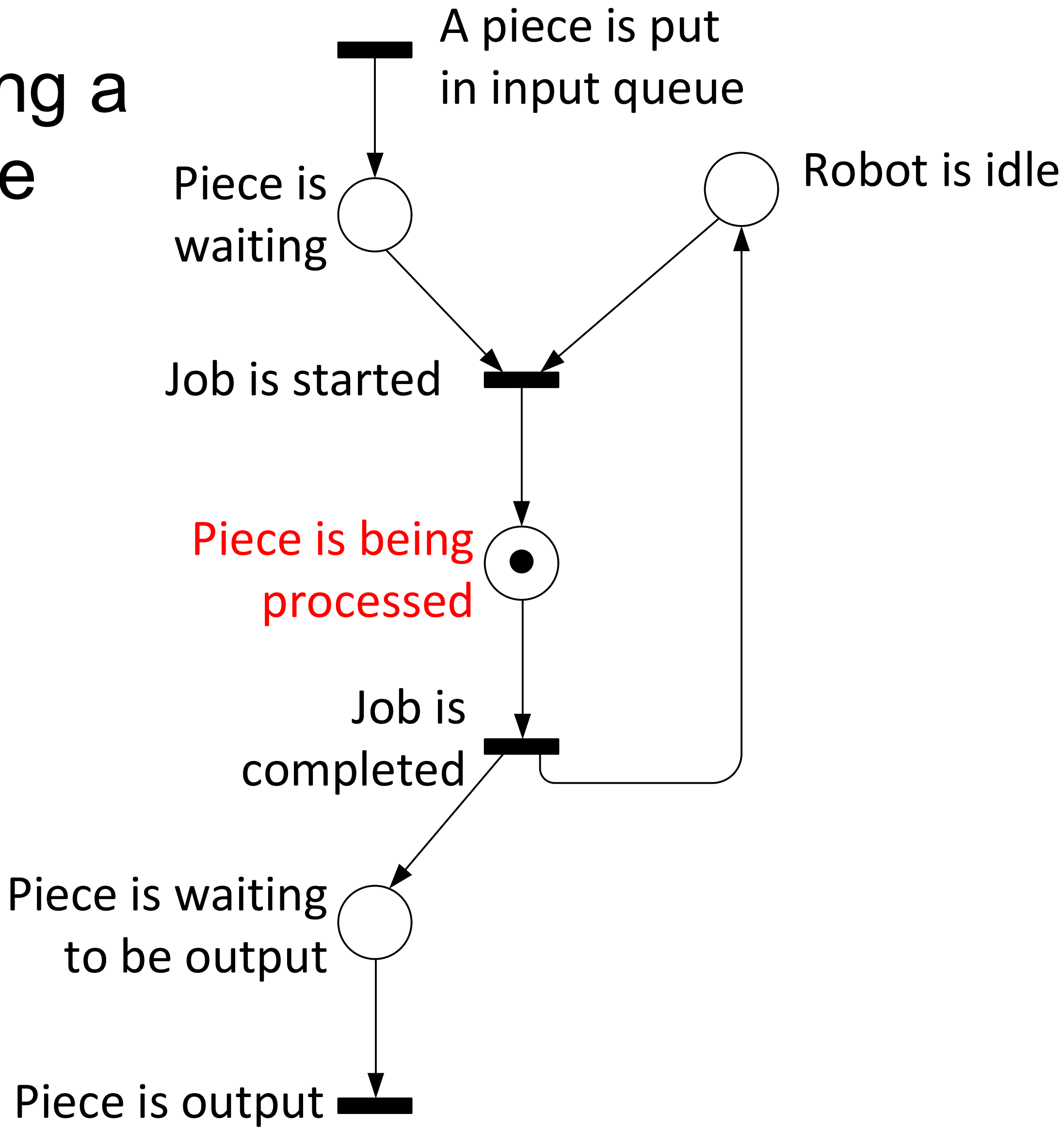
Example

- Processing a workpiece



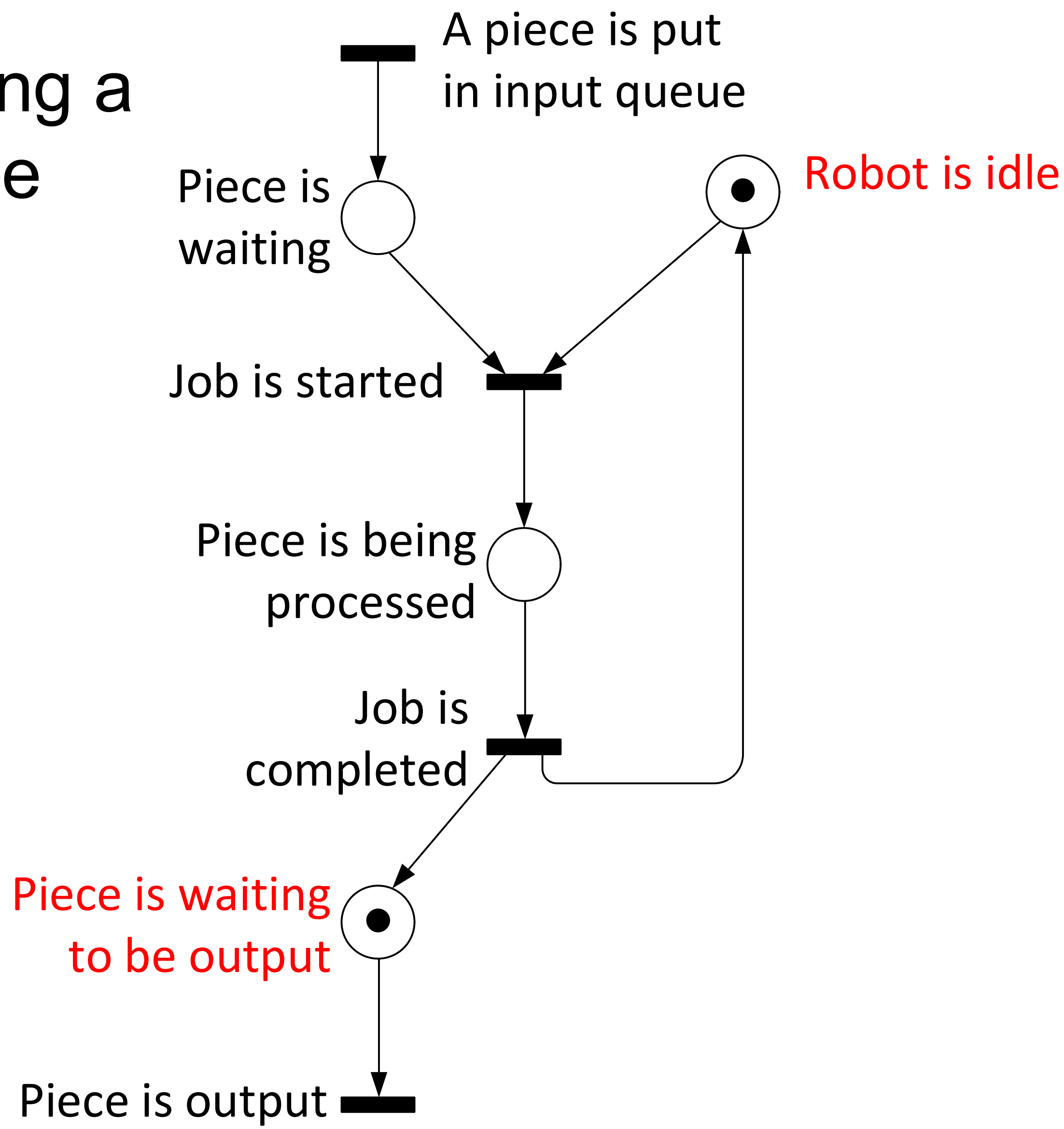
Example

- Processing a workpiece



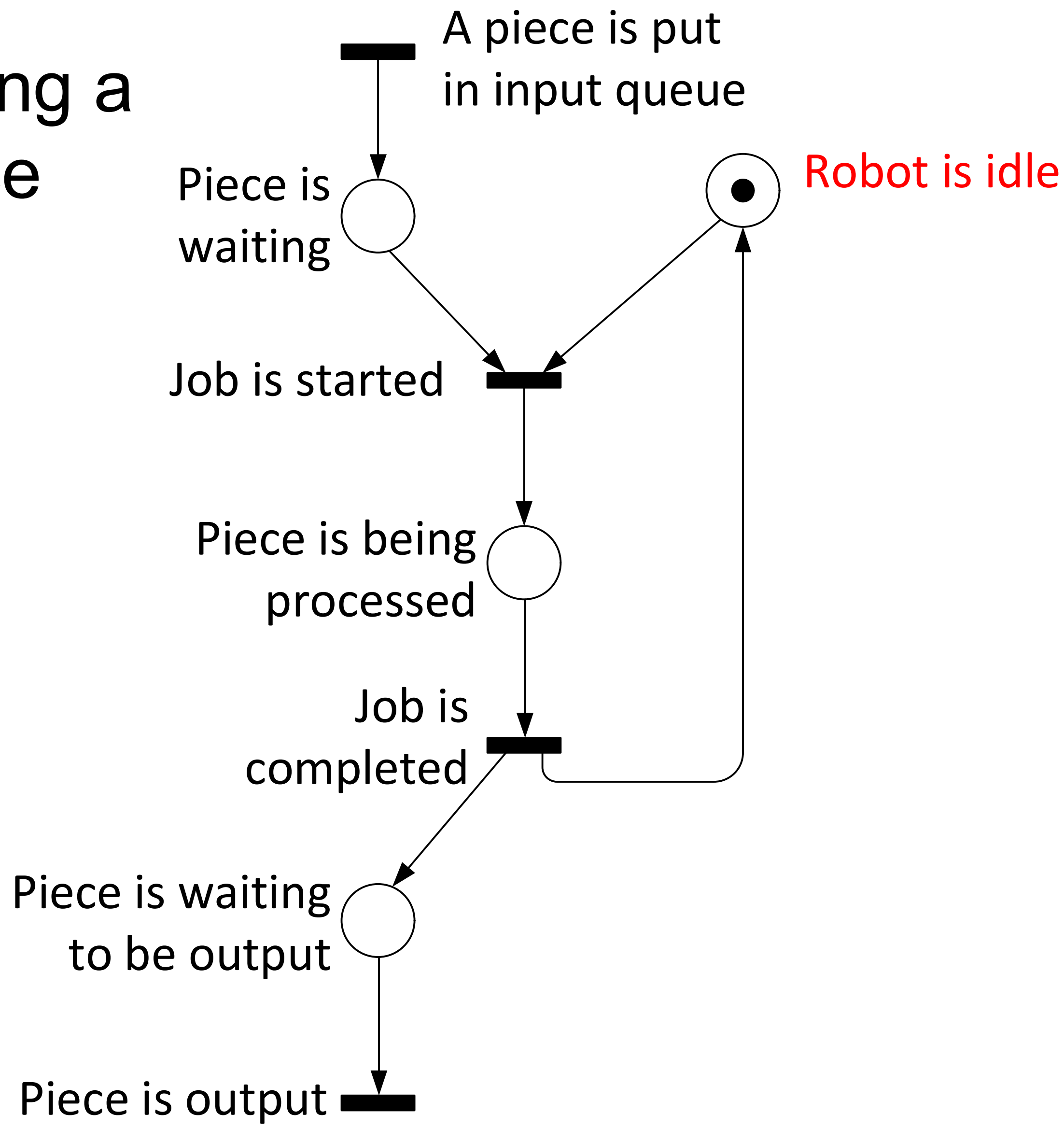
Example

- Processing a workpiece



Example

- Processing a workpiece



Formal description

- source transition has no input place $\rightarrow t_1$
- sink transition has no output place $\rightarrow t_4$
- source transitions: unconditionally enabled
- sink transitions: consumes tokens

- Petri net is ordinary: all arc weights are 1's

- self-loop: *p* is input <u>and</u> output place of t
- Petri net is pure: no self-loop in it

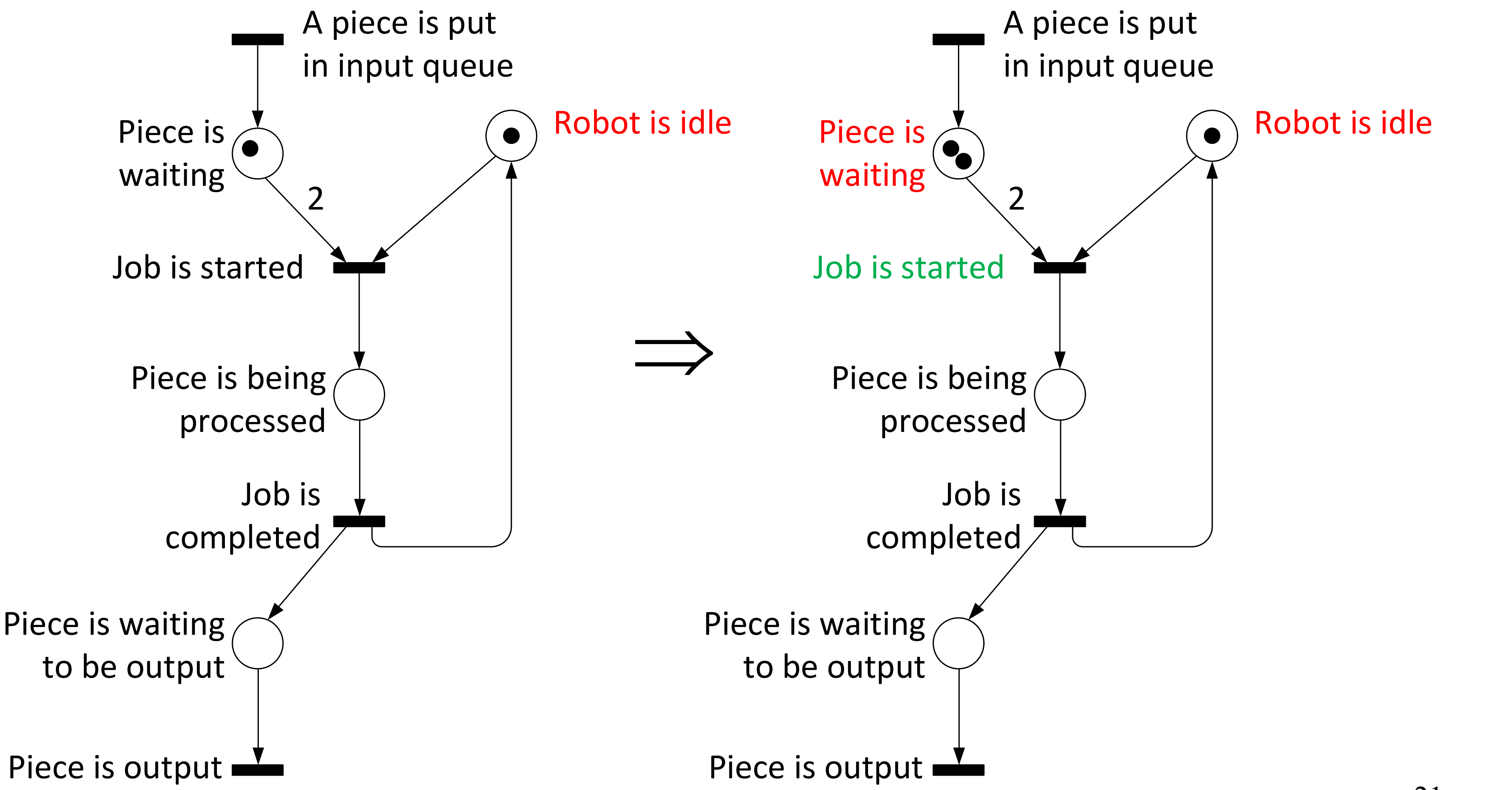
Formal description

- Capacity of places: the maximum number of tokens that they can hold any time
  - infinite capacity – no limit to number of tokens
  - finite capacity – the maximum token number is defined: $K(p)$
  - transition can fire depending on the capacity of their output places!

  - number of waiting pieces : $K(p_1) = \infty$
  - one robot is in the system: $K(p_2) = 1$
  - one piece is being processed: $K(p_3) = 1$
  - number of pieces to be output $K(p_4) = 1(!)$

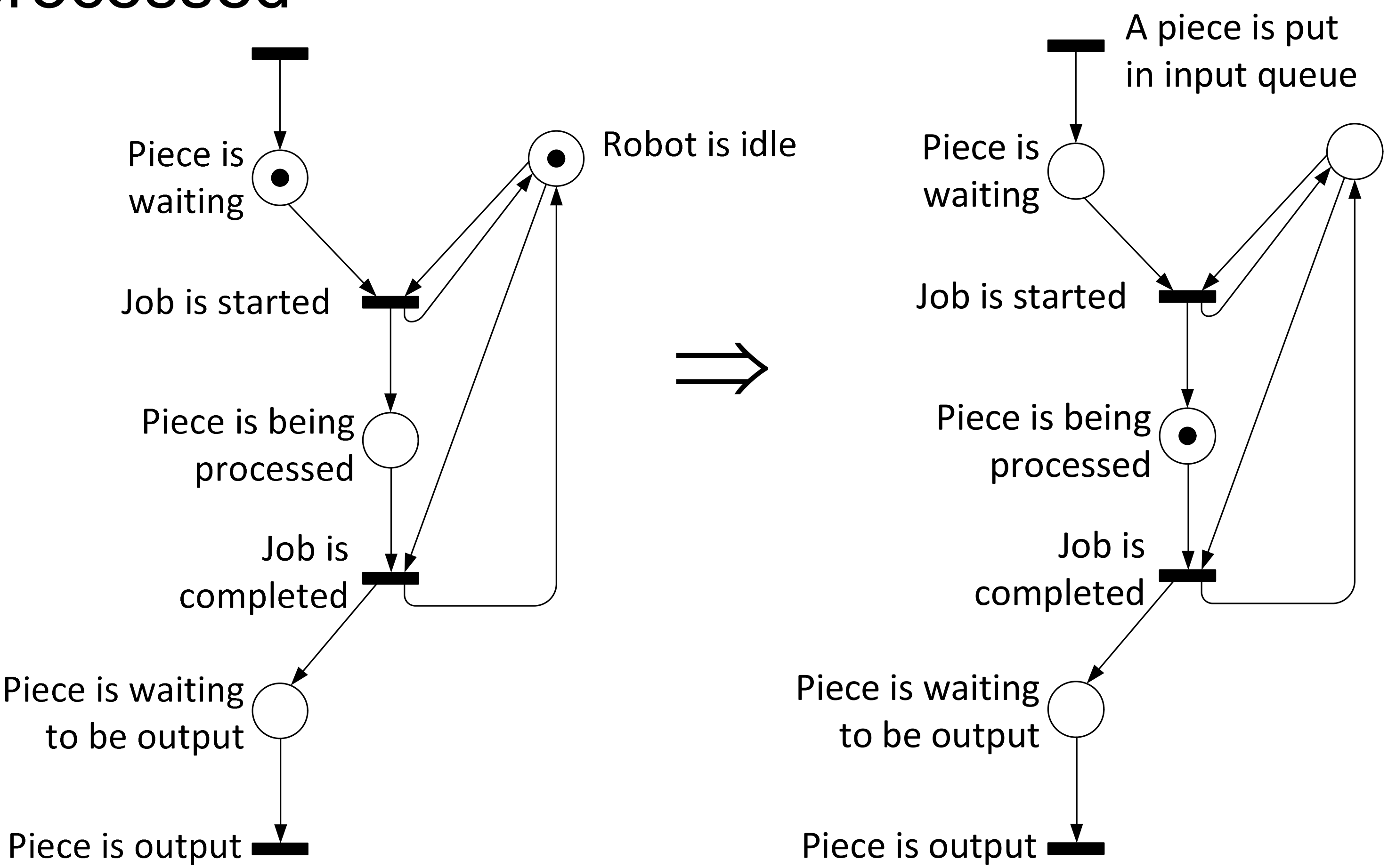
Example

- two pieces are needed for the process: $w(p_1, t_2) = 2$



Example

- Self-loop: robot is idle until the piece is being processed



Formal description

- Formal description of net state changes after firing transition t_j:

$$M_{k+1}(p_i) = M_k(p_i) - w(t_j, p_i) + w(p_i, t_j)$$

where

$M_k(p_i)$ is the number of token on place p_i

i = 1, ..., m, m is the number of places in the net

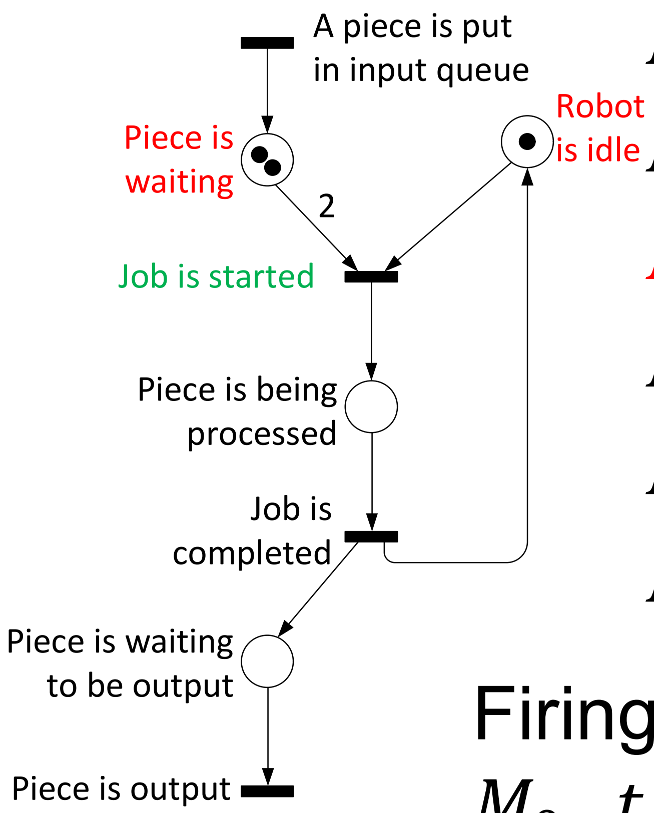
$w(t_j, p_i)$ = 0 and $w(p_i, t_j)$ = 0 no connection between t_j and p_i

- Firing or occurrence sequence:

$$M_0 \ \ t_{j_1} \ M_1 \ \ t_{j_2} \dots M_{k-1} \ \ t_{j_k} \ \ M_k$$

Example

- Markings after firing of transitions (assume 2 pieces are to be processed):



$M_0 = (0, 1, 0, 0)$ – initial marking

$M_1 = (1, 1, 0, 0)$ – after firing t_1

$M_2 = (2, 1, 0, 0)$ – after firing t_1 again

$M_3 = (0, 0, 1, 0)$ – after firing t_2

$M_4 = (0, 1, 0, 1)$ – after firing t_3

$M_5 = (0, 1, 0, 0)$ – after firing t_4

Firing sequence:

$M_0 \quad t_1 \quad M_1 \quad t_1 \quad M_2 \quad t_2 \quad M_3 \quad t_3 \quad M_4 \quad t_4 \quad M_5$

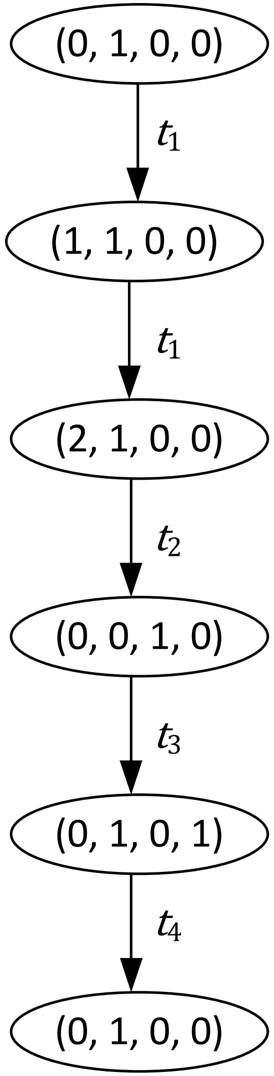
Occurrence graph

- Occurrence graph: a graph containing
  - all reachable markings from a given initial marking and
  - all possible firings at each marking

- Definition
  - $M_0 \in R(P, T, F, W, M_0)$
  - if $M' \in R(P, T, F, W, M_0)$ and $\exists t_j$ is enabled in M' and after its firing M'' is generated, then $M'' \in R(P, T, F, W, M_0)$

Example

- Occurrence graph of the example

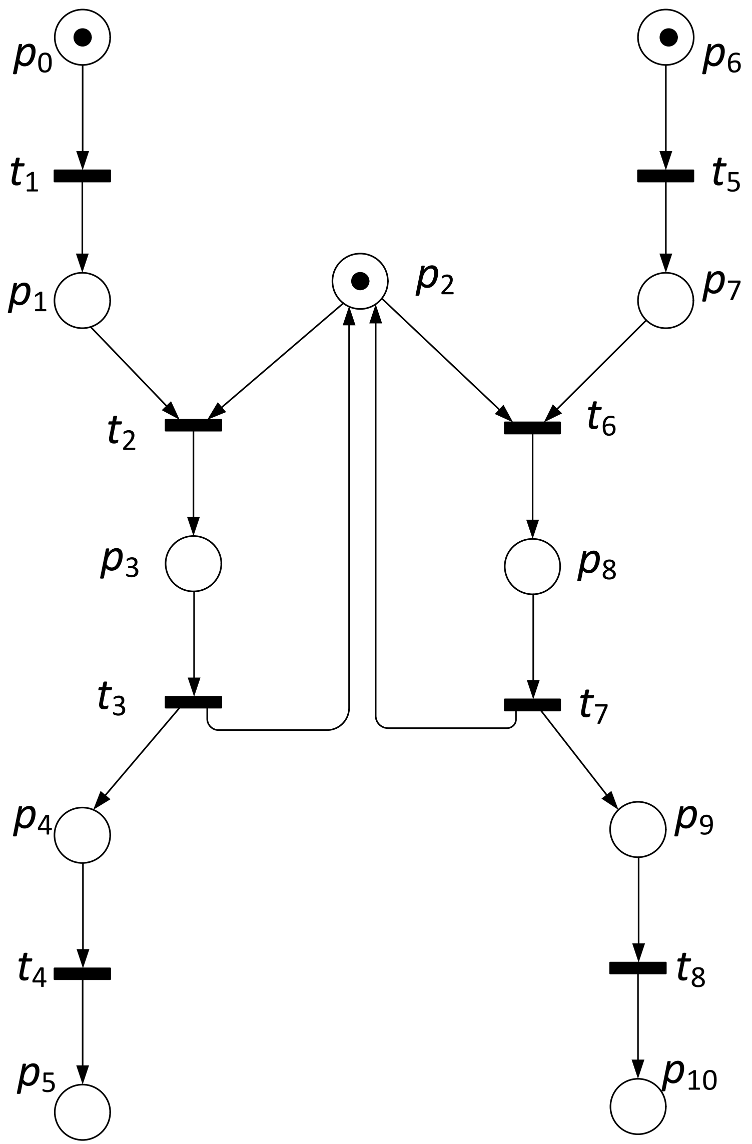


Parallel activities

- Firing two or more transitions at the same time:
  - concurrent situation: the transitions can fire independently of each other
    the places have exactly one incoming and one outcoming arc $\rightarrow$ marked graph
  - conflict situation: after firing of one transition the other will not be enabled
  - confusion: if concurrent and conflict situations present at the same time (symmetric and asymmetric)

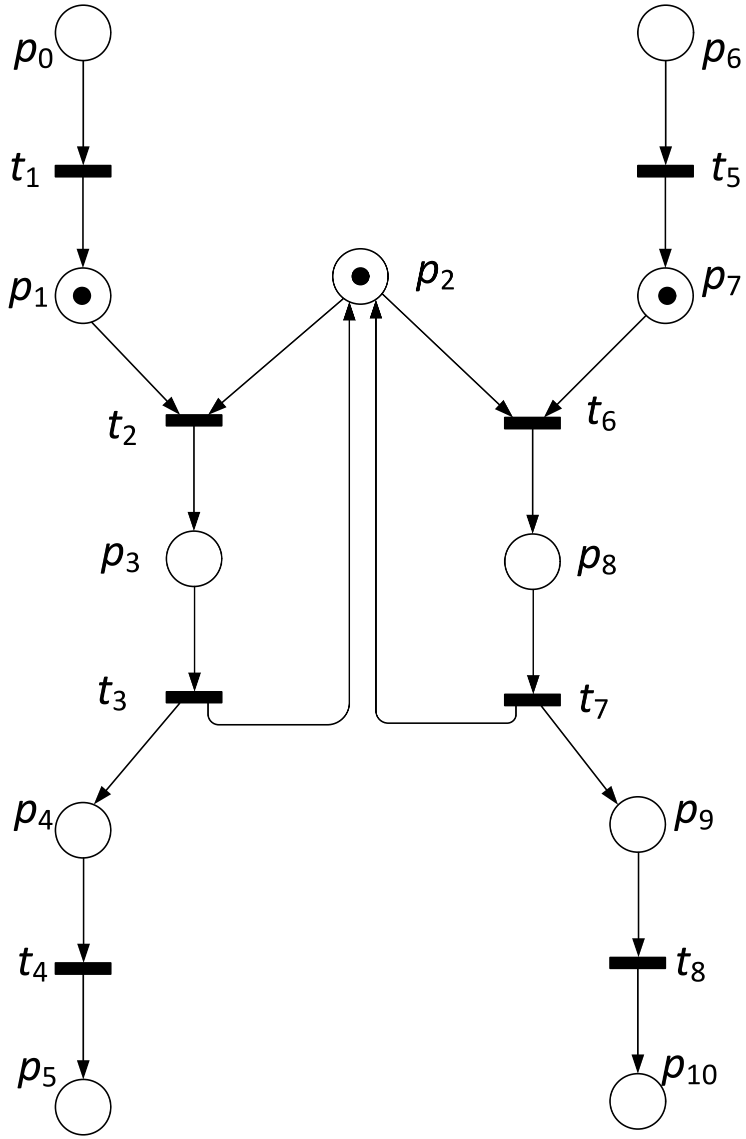
Example

- Firing of transitions: robot serves two manufacturing lines
- Transitions t_1 and t_5 **<u>can</u>** fire at the same time
- Concurrent situation



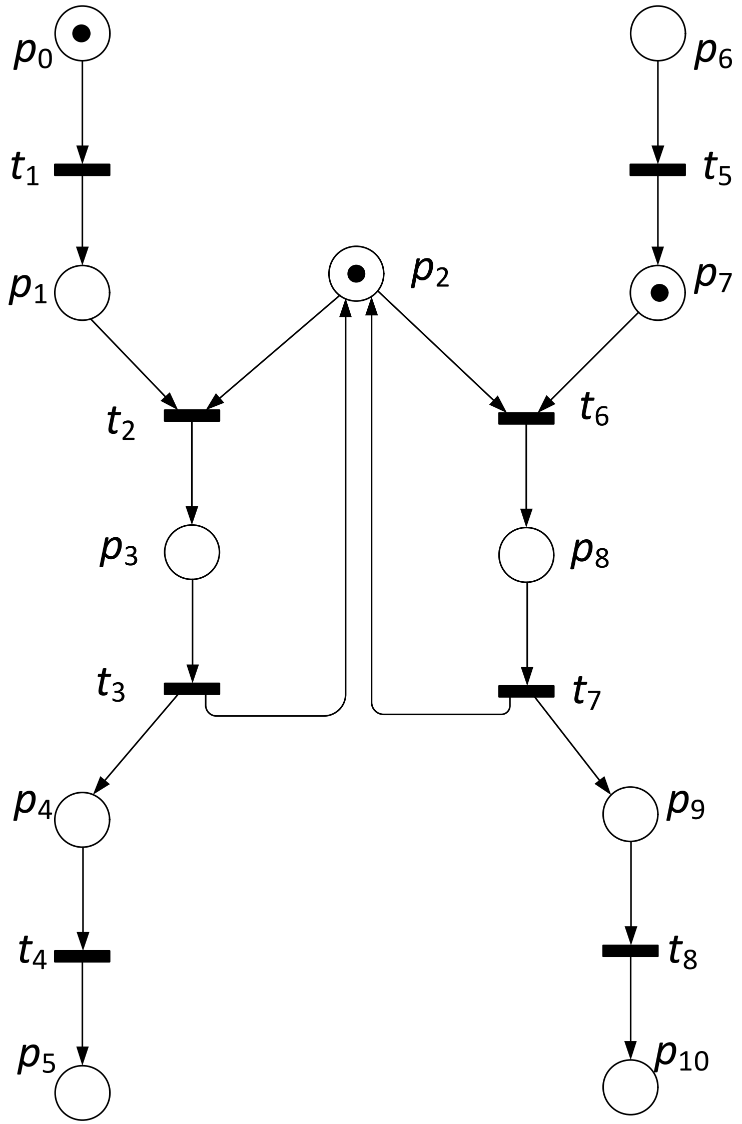
Example

- Firing of transitions: robot serves two manufacturing lines
- Transitions t_2 and t_6 **<u>can not</u>** fire at the same time
- Conflict situation



Example

- Firing of transitions: robot serves two manufacturing lines
- If transition t_6 fires first, there is no conflict
- If transition t_1 fires first, there is conflict between transitions t_2 and t_6
- Confusion



Parallel activities

- concurrent situation: arbitrary order for the firing of transitions
- conflict situation: the firing of transitions mutually exclusive
- confusion: conflict depends on the order of the firing of transitions

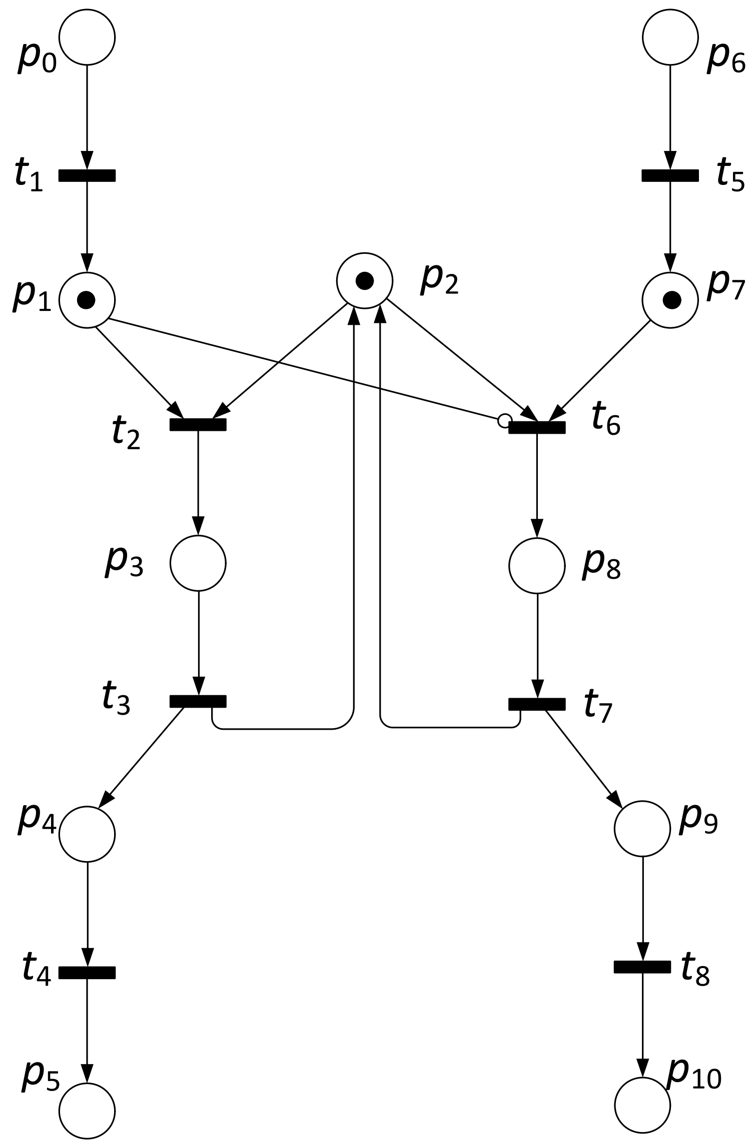
- branches on reachability tree: refer to either concurrent or conflict situation

Parallel activities

- Solutions for conflict situation:
  - inhibitor arc
    transition is enabled iff the place does not contain any token
  - priority function: transition having higher priority fires
  - extended Petri net models

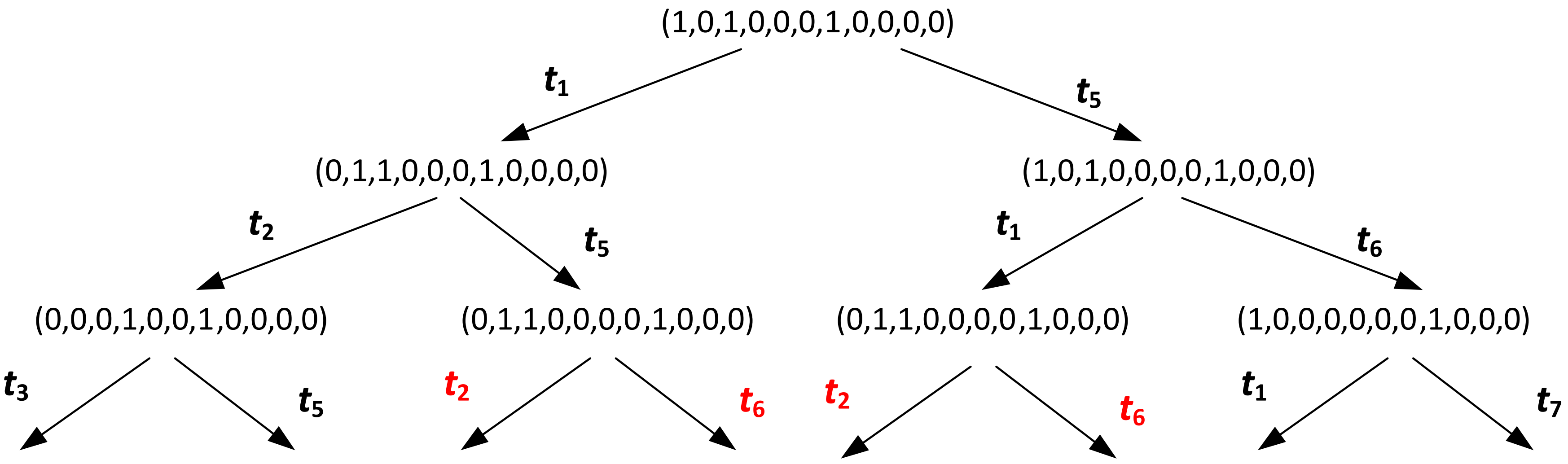
Example

- only t_2 is enabled



Reachability graph of extended example

- Reachability graph of two manufacturing line system (part)



- transitions denoted by red are in conflict

Occurrence graph

- properties of occurrence graph
  - even if the net is simple the graph can be infinite
  - solution:
    - delete duplicate nodes from the graph
    - introduction of symbol ω, where ω represents arbitrarily large number, representing the accumulation of tokens on a given place

Behavioral properties

- Analysis of Petri nets:
  - reachability
  - boundedness
  - liveness
  - reversibility
  - coverability
  - persistence
  - fairness

Behavioral properties

- Reachability
  - A marking M_n is reachable from marking M_0, if there exists a firing sequence from M_0 to M_n
  - $R(M_0)$ set of all possible markings reachable from M_0
  - reachability problem: $M_n \in R(M_0)$?
  - submarking reachability

Behavioral properties

- Boundedness
  - A Petri net is k-bounded if the number of tokens in each place does not exceed a finite number k.
  - $M(p_j)$ denotes number of tokens on a place p_j
  - boundedness problem: $M(p_j) \leq k$ for $\forall p_j$ and $\forall M_n \in R(M_0)$.
  - safe net $\Rightarrow k = 1$

Behavioral properties

- Liveness
  - deadlock-free operation
  - A Petri net is live if it is possible to fire any transition by progressing through some further firing sequence
  - liveness of all net is ideal property
  - too costly to verify
  - liveness of a given transition

Behavioral properties

- Liveness (cont.)
  - different level of liveness for a transition t:
    - $L0$-live or dead – t can never be fired in any firing sequence
    - $L1$-live or potentially fireable – if t can be fired at least once in some firing sequence
    - $L2$-live – if t can be fired at least k-times in some firing sequence
    - $L3$-live – if t can be fired infinitely often in some firing sequence
    - $L4$-live – if t is $L1$-live for every marking

Behavioral properties

- Reversibility
  - A Petri net is reversible if the initial marking $M0$ is reachable from every marking $M_n \in R(M_0)$
  - home state: a marking M’ is home state if it is reachable from every marking $M_n \in R(M_0)$

Behavioral properties

- Coverability
  - a marking M is coverable if $\exists M' \in R(M_0)$ such that $M'(p) \geq M(p)$ for $\forall p$ in the net
  - coverability $\Leftrightarrow$ $L1$-liveness:
  - let M be the minimum marking needed to enable transition t, then

    t is dead iff M is not coverable

    t is $L1$-live iff M is coverable

Behavioral properties

- Persistence
  - a Petri net is persistent if, any two enabled transitions, the firing of one transition will not disable the other
  - a transition in a persistent net stays enabled until it fires

Behavioral properties

- Fairness
  - different definitions in the literature
  - bounded-fairness: two transitions is in bounded-fair relation if the maximum number of times that either can fire while the other is not firing is bounded
  - unconditionally fairness: a firing sequence is unconditionally fair if it is finite or every transition in the net appears infinitely often in it

Behavioral properties

- Analysis of behavioral properties
  - constructing the occurrence graph for given initial markings
  - searching on the occurrence graph
    - desired or undesired markings
    - number of tokens on given place
    - firing sequences based on arc labels
    - checking the terminal nodes
  - may be NP-hard
  - cyclic behavior, symbol ω

Structural properties

- aim is to characterize the Petri net independently from the initial marking
- matrix equations governing the dynamic behavior of concurrent systems modeled by Petri nets
- solvability of these equations is limited
  - nondeterministic nature inherent in Petri net model
  - solutions must be found as non-negative integer
- assume: Petri net is pure (no self loop in it) or can be made pure

Structural properties

- incidence matrix
- let the number of transition n and the number of places m in a Petri net
- the incidence matrix $A = [a_{ij}]$

$$a_{ij} = a^{+}_{ij} - a^{-}_{ij}$$

where $a^{+}_{ij} = w(i,j)$ is the weight of the arc from t_i to p_i and $a^{-}_{ij} = w(j,i)$ is the weight of the arc from p_i to t_i

Structural properties

- a^{-}_{ij} is the number of tokens to be removed
- a^{+}_{ij} is the number of tokens to be added
- a_{ij} is the number of tokens changed in a place

- transition t_i is enabled iff

$$a^{-}_{ij} < M(j), \quad j = 1, 2, \ldots, m$$

Structural properties

- State equations:
  - let M_k is $m \times 1$ column vector and $M_k(j)$ denotes the number tokens in place j after kth firing in some firing sequence
  - let control or firing vector u_k is $n \times 1$ unit column vector, the value 1 in ith position indicates that transition i fires at kth firing
  - state equation of Petri net:

$$M_k = M_{k-1} + A^T u_k \qquad k = 1, 2, \ldots$$

Structural properties

- Necessary reachability condition:
  - let M_d is reachable from M_0 trough a firing sequence $\{u_1, u_2, ..., u_d\}$
  - expressing with state equation:

$$M_d = M_0 + A^T \sum_{k=1}^{d} u_k$$

or

$$A^T x = \Delta M$$

where $\Delta M = M_d - M_0$; $x = \Sigma u_k$

x firing count vector

Structural properties

- equation $A^T x = \Delta M$ has a solution x iff ΔM is orthogonal to every solution y of its homogeneous system:

$$Ay = 0$$

- T-invariant: An integer solution of the homogeneous equation:

$$A^T x = 0$$

- P-invariant: An integer solution of the homogeneous equation:

$$Ay = 0$$

Structural properties

- T-invariants:
  - if x is a T-invariants then there exists a marking M_0 and firing sequence starting from M_0 back to M_0, that its firing count vector is equal to x

- P-invariants:
  - if y is a P-invariants then $M^T y = M_0{}^T y$ for any fixed initial marking M_0 and any M in $R(M_0)$

Automata, formal languages and Petri nets

- Automata and Petri nets:
  - both suitable for representing DES
  - explicit representation of state transitions
  - automata: the definition contains the possible states and the possible transitions between them in explicit way
  - Petri nets: the state description is defined in distributed way, it is encoded into the state of places

Automata, formal languages and Petri nets

- Petri-net languages
  - labelled Petri net generating the context-sensitive language $L(M_0) = \{a^n b^n c^n \mid n \geq 0\}$