Petri Nets

- A Petri net is a graphical and mathematical modeling tool.

- A Petri net consists of places, transitions, and arcs that connect them. Input arcs connect places with transitions, while output arcs start at a transition and end at a place.

- Places can contain tokens; the current state of the modeled system (the marking) is given by the number (and type if the tokens are distinguishable) of tokens in each place.

- The concept of Petri nets has its origin in Carl Adam Petri’s dissertation “Kommunikation mit Automaten”, submitted in 1962 to the faculty of Mathematics and Physics at the Technische Universität Darmstadt, Germany.
Petri Net Transitions

- Transitions are active components. They model activities which can occur (the transition *fires*), thus changing the state of the system (the marking of the Petri net).

- Transitions are only allowed to fire if they are *enabled*, which means that all the preconditions for the activity must be fulfilled (there are enough tokens available in the input places).

- When the transition fires, it removes tokens from its input places and adds some at all of its output places. The number of tokens removed/added depends on the cardinality of each arc.

- The interactive firing of transitions in subsequent markings is called token game.
A Petri net $C = \langle P, T, I, O \rangle$ consists of:

- A finite set $P$ of places
- A finite set $T$ of transitions
- An input function $I : T \rightarrow \mathcal{N}^p$ (maps to bags of places)
- An output $O : T \rightarrow \mathcal{N}^p$

A marking of $C$ is a mapping $\mu : P \rightarrow \mathcal{N}$

Example:

$P = \{x, y\}$
$T = \{a, b\}$
$I(a) = \{x\}, I(b) = \{x, x\}$
$O(a) = \{x, y\}, O(b) = \{y\}$
$\mu = \{x, x\}$
Firing Transitions

To fire a transition t:
1. There must be enough tokens: $\mu \geq I(t)$
2. Consume inputs and generate output: $\mu' = \mu - I(t) + O(t)$
Modeling Seasons

Spring Start of Spring → Start of Summer → Summer

Start of Spring → Start of Fall → Fall

Winter → Start of Winter
Uses of Petri Nets

Petri nets are good for modeling:
- Concurrency
- Synchronization

Tokens can represent:
- Resource availability
- Jobs to perform
- Flow of control
- Synchronization conditions
Concurrency

Independent inputs permit “concurrent” firing of transition:
Conflict

Overlapping inputs put transitions in conflict:

Only one of a or b may fire.
Mutual Exclusion

The two subnets are forced to synchronize:
Fork and Join

Subsystem A

Subsystem B

Subsystem C

Com S 430
Producers and Consumers
Bounded Buffer

Producer

Consumer

Free slots

Used slots

Com S 430
Reachability and Boundedness

Reachability:
- The reachability set $R(C, \mu)$ of a Petri net $C$ is the set of all markings $\mu'$ reachable from initial marking $\mu$.

Boundedness:
- A Petri net $C$ with an initial marking $\mu$ is safe if its places always hold at most 1 token.
- A marked Petri net is *(k-)*bounded if its places never hold more than $k$ tokens.
- A marked Petri net is conservative if the number of tokens is constant.
Liveness and Deadlock

Liveness:

- A transition is **deadlocked** if it can never fire.
- A transition is live if it can never deadlock.

This Petri net is both safe and conservative.
A transition “a” is deadlocked.
Transitions “b” and “c” are live.
The reachability set is \{y\}, \{z\}.
Other Variants

- **Colored Petri nets:**
  - Tokens are “colored” to represent different *kinds* of resources.

- **Augmented Petri nets:**
  - Transitions additionally depend on external *conditions*.

- **Timed Petri nets:**
  - A duration is associated with each transition.