Elementary Petri-Nets

At most one token per place
Workflow Nets

- Unique source place
- Unique sink place
- Connected
- Unique initial marking, unique terminal marking
- Well-formed – every transition is reachable, every marking is reachable, every marking terminates
Types Annotations for activities

- **Automatic Activity**
  - Computer can execute it fully
    - (when enabled, it is executed automatically such as by an algorithm, script task etc.)

- **User Activity**
  - A human being executes it
    - (though enable, it is done manually)

- **Message Activity**
  - An external message triggers the task instance
    - (though enabled, it requires a message to trigger it)

- **Time Triggered Activity**
  - Task needs to be trigged at a particular time, or after a certain period of timeout
    - (though enabled, time has to trigger it)
Types of Activities
An Example

P1=Ready to register
P2=not approved
P3=permitted to Attend classes
P4=state of opinion
P5=approved
P6=Process Terminated
P7=facad is Ready To approve

Submit registration
Get it approved
Attend classes
Form Filled
Fill up form
Evaluate Registration form
Form Evaluated
Submit adjustment
Evaluate Registration form
Submit adjustment

Timeout triggered activity
Classical Petri Nets

- A place can contain 0 or more tokens (unlimited number of them)
  - Infinite capacity net
  - A state in state space then needs to mention the count of tokens held in places
    - e.g. {1 p1, 2 p3, 4 p4} or in another notation (1,0,2,4) enumerating the numbers in each place
      - Note, this state is different from {2 P1, 2 P3, 4 p4}
- Arcs between places and transitions can have weights
  - Place to transition: requires those many tokens in the place for transition to fire
  - Transitions to place: produces those many tokens if the transition fires

- Source transition: no input place- is unconditionally enabled all the time
- Sink transition: no output place – consumes token.
- Pure petri net: without self loops
- Ordinary petri nets: arc weight is 1 (default)
- Finite capacity net: places are marked with capacities
Showing Arcs with weights

Default arc weight is 1
Similarly default capacity is inf
But if the net is elementary, capacity is 1, and arc weight is 1

2 tokens required to enable the transition

2 tokens are produced after firing the transition
Example
what would be the marking after firing the transition?
After this, no more progress
Places with multiple tokens

Build state space
Initial marking: \{P1\}
Exercise: Doctors and Patients in a day using Classical PN

Slot Availability

Patient Arrival

Patient Departure

Doctor Available

Meet

Treatment

Treatment Completed

Return to room

Doctor Left

Refine, Annotate with arc weights, Place capacities, Identify initial marking
An Improved Model

Modules in the System:
1. Meeting Room
2. Doctor
3. Patient
4. Slot System

Yet it is a single net
Producers and Consumers

Producer

Consumer

Buffer

Produce

CONSUME

N Tokens available inside the place

Place can accept at most N tokens
Producers and Consumers with MUTEX For the buffer

**Diagram Description**

- **Producer**: Produces tokens and places them in the buffer.
- **Buffer**: Holds tokens with a capacity of N. When full, a token is blocked and waits for consumption. When empty, a token cannot be produced.
- **Consumer**: Consumes tokens from the buffer.

**Facts**

- **N Tokens available inside the place**: Represents the tokens currently available in the buffer.
- **Place can accept at most N tokens**: Indicates the maximum capacity of the buffer.

**Keywords**

- PRO DUC E
- C O N S U M E R
- M U T E X
- N