innovate achieve lead

BITS Pilani
Pilani Campus

CS/IS F214
Logic in Computer Science

MODULE: **TEMPORAL LOGICS**

State Machines for Modelling

State Machines as Models

- We will study the use of *state machines* for modeling:
  - in particular, we will use *paths in a state machine as models*
    - for providing the meaning of formulas written in a ***temporal logic***.

- This logic is referred to as ***Linear Temporal Logic*** where :
  - *formulas are interpreted in the context of paths*
    - along which the truth (or falsity) of atomic propositions could change

State Machines and Graphs

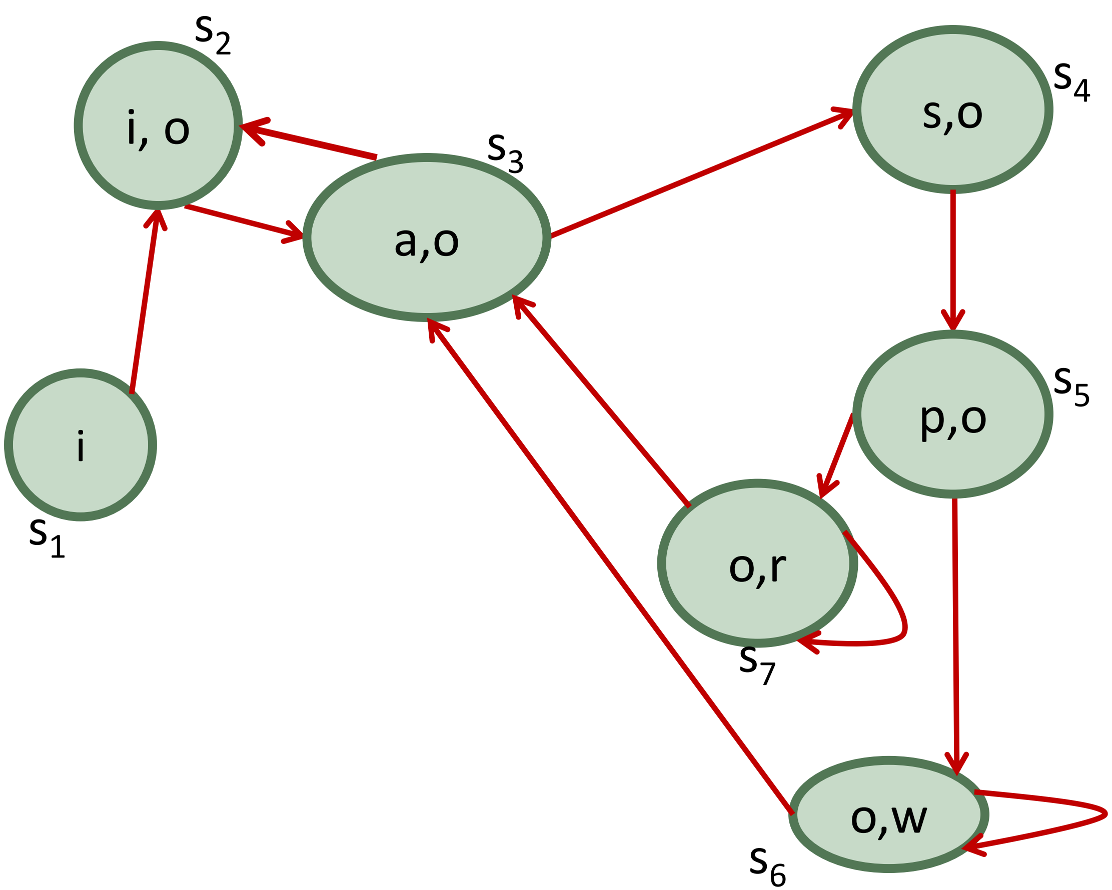
- A state machine can be represented as *a vertex-labeled, directed graph*:
  - <**S**, **→**, **L**> where
    - **S** is a set of states
    - **→** is the *transition* relation (i.e. a *total*, *ordered*, *binary relation* on S)
    - **L** is a *labeling function*:
      - i.e. **L : S --> Ᵽ(Atoms)**
        - where **Ᵽ(Atoms)** is the *power-set* of **Atoms**, which is a *set of atomic propositions*

State Machine – Example: I/O in a hard disk

- Operation of a hard disk (assumed to have a single platter i.e. a *single head*):
  - Disk must be *activated to rotate*
  - *Rotation must be steady* to accept requests
  - Request queue may only contain *requests of the same type*
    - e.g. all *read* requests with the same *cylinder* address – i.e. *track* address
  - *Seeking is required* to position the head over the track
  - *Request will be pending until* the requested (*sector*) address rotates under the head

State Machine - Example

- Often, we use a pictorial representation of a state machine.
  - e.g. state machine M depicting *operation of a hard disk*:



Atoms = { **i**dle, **ro**tating, **a**ccepting, **s**eeking, **p**ending, **r**eading, **w**riting }

S = { s_1, s_2, s_3, s_4, s_5, s_6, s_7 }

$L(s_1)$ = { i }
$L(s_2)$ = { i,o }
$L(s_3)$ = { a,o }
$L(s_4)$ = { s,o }
$L(s_5)$ = { p,o }
$L(s_6)$ = { o,w }
$L(s_7)$ = { o,r }

State Machines and Semantics

- Interpretation using state machine, **<S, $\rightarrow$,L>**:
  - Each **state** in **S** is a point in time
    - i.e. a point in future (or time δ >= 0 from now)
  - **L** is an assignment of truth values to all propositional atoms
    - i.e. in a given state:
      - the valuation of all propositional atoms can be stated as the subset of those atoms that are TRUE
  - **$\rightarrow$ is total**
    - i.e. from every state, there is an outgoing transition
      - therefore we will consider paths that are infinite

State Machines - Paths

- Define **paths** as <u>infinite sequences of states</u>
  - A ***path*** in a model $\boldsymbol{M} = (\mathbf{S}, \rightarrow, \mathbf{L})$ is an infinite sequence of states $\mathbf{s_1, s_2, \ldots}$ in **S** such that:
    - $\forall \mathbf{i}\ \ \mathbf{i>=1}\ \ \mathbf{-->}\ \ \mathbf{s_i \rightarrow s_{i+1}}$
- We will denote paths as:
  - $\mathbf{s_1 \rightarrow s_2 \rightarrow \ldots}$

- A **path** $\boldsymbol{\pi} = \mathbf{s_1 \rightarrow s_2 \rightarrow \ldots}$ is a <u>*possible future*</u> in our system:
  - it is first in state s_1, then it is in state s_2, and so on.
- We will denote the <u>suffix of a path</u> $\boldsymbol{\pi}$ starting at state $\mathbf{s_i}$ as $\boldsymbol{\pi}^{\mathbf{i}}$
  - e.g. π^4 is $s_4 \rightarrow s_5 \rightarrow \ldots$