At the end of the class you should be able to:

- the model of deterministic planning
- represent a problem using both STRIPs and the feature-based representation of actions.
Planning is deciding what to do based on an agent’s ability, its goals, and the state of the world.

Planning is finding a sequence of actions to solve a goal.

Initial assumptions:

- The world is deterministic.
- There are no exogenous events outside of the control of the robot that change the state of the world.
- The agent knows what state it is in.
- Time progresses discretely from one state to the next.
- Goals are predicates of states that need to be achieved or maintained.
A deterministic action is a partial function from states to states.

The preconditions of an action specify when the action can be carried out.

The effect of an action specifies the resulting state.
Features:
- $RLoc$ – Rob’s location
- $RHC$ – Rob has coffee
- $SWC$ – Sam wants coffee
- $MW$ – Mail is waiting
- $RHM$ – Rob has mail

Actions:
- $mc$ – move clockwise
- $mcc$ – move counterclockwise
- $puc$ – pickup coffee
- $dc$ – deliver coffee
- $pum$ – pickup mail
- $dm$ – deliver mail
### Explicit State-space Representation

<table>
<thead>
<tr>
<th>State</th>
<th>Action</th>
<th>Resulting State</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\langle \text{lab, rhc, swc, mw, rhm} \rangle)</td>
<td>(mc)</td>
<td>(\langle \text{mr, rhc, swc, mw, rhm} \rangle)</td>
</tr>
<tr>
<td>(\langle \text{lab, rhc, swc, mw, rhm} \rangle)</td>
<td>(mcc)</td>
<td>(\langle \text{off, rhc, swc, mw, rhm} \rangle)</td>
</tr>
<tr>
<td>(\langle \text{off, rhc, swc, mw, rhm} \rangle)</td>
<td>(dm)</td>
<td>(\langle \text{off, rhc, swc, mw, rhm} \rangle)</td>
</tr>
<tr>
<td>(\langle \text{off, rhc, swc, mw, rhm} \rangle)</td>
<td>(mcc)</td>
<td>(\langle \text{cs, rhc, swc, mw, rhm} \rangle)</td>
</tr>
<tr>
<td>(\langle \text{off, rhc, swc, mw, rhm} \rangle)</td>
<td>(mc)</td>
<td>(\langle \text{lab, rhc, swc, mw, rhm} \rangle)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
For each action:
- **precondition** is a proposition that specifies when the action can be carried out.

For each feature:
- **causal rules** that specify when the feature gets a new value and
- **frame rules** that specify when the feature keeps its value.
Precondition of pick-up coffee (\textit{puc}): 
\[ RLoc = cs \land \overline{rhc} \]

Rules for location is \textit{cs}:
\[ RLoc' = cs \leftarrow RLoc = \text{off} \land Act = \text{mcc} \]
\[ RLoc' = cs \leftarrow RLoc = \text{mr} \land Act = \text{mc} \]
\[ RLoc' = cs \leftarrow RLoc = cs \land Act \neq \text{mcc} \land Act \neq \text{mc} \]

Rules for “robot has coffee”
\[ rhc' \leftarrow rhc \land Act \neq \text{dc} \]
\[ rhc' \leftarrow Act = \text{puc} \]
Divide the features into:

- primitive features
- derived features. There are rules specifying how derived can be derived from primitive features.

For each action:

- **precondition** that specifies when the action can be carried out.
- **effect** a set of assignments of values to primitive features that are made true by this action.

STRIPS assumption: every primitive feature not mentioned in the effects is unaffected by the action.
Example STRIPS representation

Pick-up coffee (puc):
- **precondition**: [cs, rhc]
- **effect**: [rhc]

Deliver coffee (dc):
- **precondition**: [off, rhc]
- **effect**: [rhc, swc]