

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

- explain the model of deterministic planning
- represent a problem using the STRIPs representation of actions.

State-space Search

- **deterministic** or stochastic dynamics
- **fully observable** or partially observable
- **explicit states** or features or individuals and relations
- static or finite stage or **indefinite stage** or infinite stage
- **goals** or complex preferences
- **perfect rationality** or bounded rationality
- **flat** or modular or hierarchical
- **single agent** or multiple agents
- **knowledge is given** or knowledge is learned
- **reason offline** or reason while interacting with environment

Classical Planning

- **deterministic** or stochastic dynamics
- **fully observable** or partially observable
- explicit states or **features** or individuals and relations
- static or finite stage or **indefinite stage** or infinite stage
- **goals** or complex preferences
- **perfect rationality** or bounded rationality
- **flat** or modular or hierarchical
- **single agent** or multiple agents
- **knowledge is given** or knowledge is learned
- **reason offline** or reason while interacting with environment

- Planning is deciding what to do based on an agent's ability, its goals, and the state of the world.

- Planning is deciding what to do based on an agent's ability, its goals, and the state of the world.
- 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.

- Planning is deciding what to do based on an agent's ability, its goals, and the state of the world.
- 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. (Fully observable)

- Planning is deciding what to do based on an agent's ability, its goals, and the state of the world.
- 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. (Fully observable)
 - ▶ Time progresses discretely from one state to the next.

- Planning is deciding what to do based on an agent's ability, its goals, and the state of the world.
- 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. (Fully observable)
 - ▶ Time progresses discretely from one state to the next.
 - ▶ Goals are predicates of states that need to be achieved.

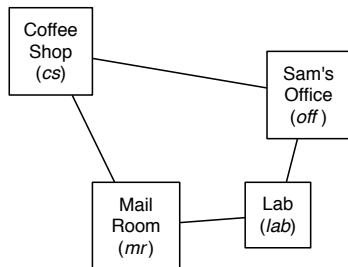
- Planning is deciding what to do based on an agent's ability, its goals, and the state of the world.
- Initial assumptions:
 - ▶ The world is deterministic. There are no exogenous events outside of the control of the robot that change the state of the world.
 - ▶ The agent knows what state it is in. (Fully observable)
 - ▶ Time progresses discretely from one state to the next.
 - ▶ Goals are predicates of states that need to be achieved.
- Planning is finding a sequence of actions to solve a goal.

- A deterministic **action** is a partial function from states to states.

- A deterministic **action** is a partial function from states to states.
- The **preconditions** of an action specify when the action can be carried out.

- 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.

Delivery Robot Example



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

State	Action	Resulting State
$\langle lab, \neg rhc, swc, \neg mw, rhm \rangle$	<i>mc</i>	$\langle mr, \neg rhc, swc, \neg mw, rhm \rangle$
$\langle lab, \neg rhc, swc, \neg mw, rhm \rangle$	<i>mcc</i>	$\langle off, \neg rhc, swc, \neg mw, rhm \rangle$
$\langle off, \neg rhc, swc, \neg mw, rhm \rangle$	<i>dm</i>	$\langle off, \neg rhc, swc, \neg mw, \neg rhm \rangle$
$\langle off, \neg rhc, swc, \neg mw, rhm \rangle$	<i>mcc</i>	$\langle cs, \neg rhc, swc, \neg mw, rhm \rangle$
$\langle off, \neg rhc, swc, \neg mw, rhm \rangle$	<i>mc</i>	$\langle lab, \neg rhc, swc, \neg mw, rhm \rangle$
...

Explicit State-space Representation

State	Action	Resulting State
$\langle lab, \neg rhc, swc, \neg mw, rhm \rangle$	<i>mc</i>	$\langle mr, \neg rhc, swc, \neg mw, rhm \rangle$
$\langle lab, \neg rhc, swc, \neg mw, rhm \rangle$	<i>mcc</i>	$\langle off, \neg rhc, swc, \neg mw, rhm \rangle$
$\langle off, \neg rhc, swc, \neg mw, rhm \rangle$	<i>dm</i>	$\langle off, \neg rhc, swc, \neg mw, \neg rhm \rangle$
$\langle off, \neg rhc, swc, \neg mw, rhm \rangle$	<i>mcc</i>	$\langle cs, \neg rhc, swc, \neg mw, rhm \rangle$
$\langle off, \neg rhc, swc, \neg mw, rhm \rangle$	<i>mc</i>	$\langle lab, \neg rhc, swc, \neg mw, rhm \rangle$
...

What happens when we also want to model its battery charge?

Want “elaboration tolerance”.

STRIPS Representation

The state is a function from features into values. So it can be represented as a set of feature-value pairs.

For each action:

- **precondition** a set of feature-value pairs that must be true for the action to be carried out.

The state is a function from features into values. So it can be represented as a set of feature-value pairs.

For each action:

- **precondition** a set of feature-value pairs that must be true for the action to be carried out.
- **effect** a set of a feature-value pairs that are made true by this action.

STRIPS Representation

The state is a function from features into values. So it can be represented as a set of feature-value pairs.

For each action:

- **precondition** a set of feature-value pairs that must be true for the action to be carried out.
- **effect** a set of a feature-value pairs that are made true by this action.

STRIPS assumption: every feature not mentioned in the effect is unaffected by the action.

Example STRIPS representation

Pick-up coffee (*puc*):

- **precondition:** $\{RLoc = cs, RHC = False\}$. “The robot needs to be at the coffee shop and not holding coffee in order to pick up coffee.”

Example STRIPS representation

Pick-up coffee (*puc*):

- **precondition:** $\{RLoc = cs, RHC = False\}$. “The robot needs to be at the coffee shop and not holding coffee in order to pick up coffee.”
- **effect:** $\{RHC = True\}$. “After the robot picks up coffee, it is holding coffee. Nothing else has changed.”

Example STRIPS representation

Pick-up coffee (*puc*):

- **precondition:** $\{RLoc = cs, RHC = False\}$. “The robot needs to be at the coffee shop and not holding coffee in order to pick up coffee.”
- **effect:** $\{RHC = True\}$. “After the robot picks up coffee, it is holding coffee. Nothing else has changed.”

Deliver coffee (*dc*):

- **precondition:** $\{Rloc = off, RHC = True\}$ “The robot needs to be at the office and holding coffee in order to deliver coffee.”

Example STRIPS representation

Pick-up coffee (*puc*):

- **precondition:** $\{RLoc = cs, RHC = False\}$. “The robot needs to be at the coffee shop and not holding coffee in order to pick up coffee.”
- **effect:** $\{RHC = True\}$. “After the robot picks up coffee, it is holding coffee. Nothing else has changed.”

Deliver coffee (*dc*):

- **precondition:** $\{Rloc = off, RHC = True\}$ “The robot needs to be at the office and holding coffee in order to deliver coffee.”
- **effect:** $\{RHC = False, SWC = False\}$ “After the robot delivers coffee, it is not holding coffee, and Sam no longer wants coffee. Nothing else has changed.”

Given:

- A description of the effects and preconditions of the actions
- A description of the initial state
- A goal to achieve

find a sequence of actions that is possible and will result in a state satisfying the goal.

