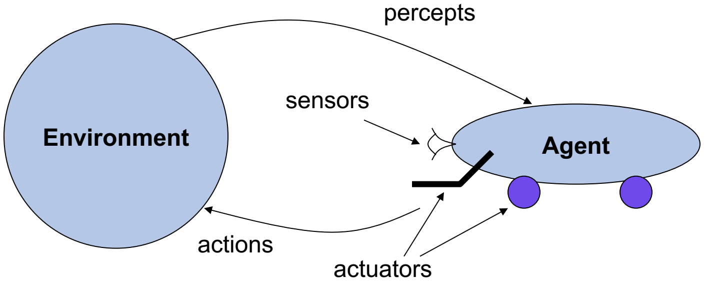
Agents

1

An Agent

- **Def:** An agent is anything that can *perceive* its environment through a set of *sensors* and *act* upon that environment through a set of *actuators*.



2

A Rational Agent

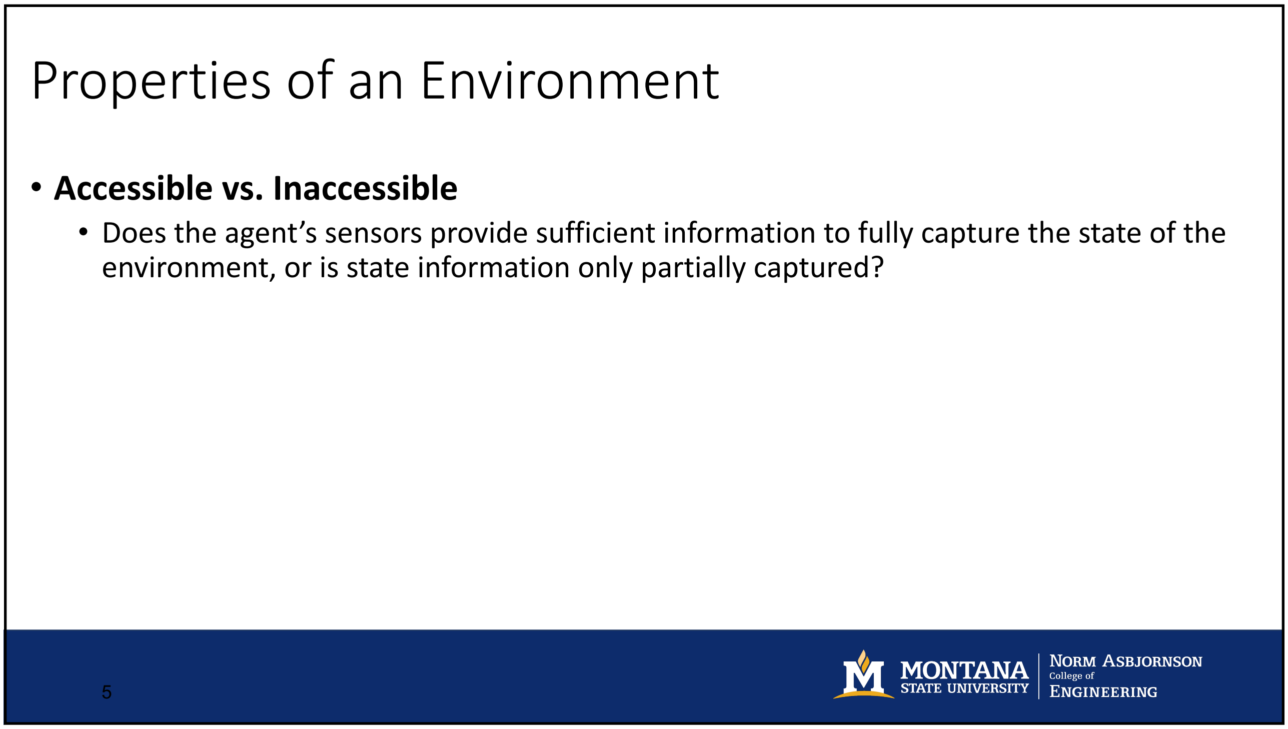
- **Def 1:** An agent is said to be *rational* if it “does the right thing.”
- **Def 2:** For each possible percept sequence, a *rational agent* should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.
- **Key elements:**
  - Performance measure (to be maximized)
  - Environment (and knowledge about the environment)
  - Available actions (determined by actuators)
  - Percept sequence (sensor information)

3

PEAS Model

- **P**erformance Measure
  - Distance traveled, patient outcome, experience points
- **E**nvironment
  - City streets, hospital or clinic, Oribos
- **A**ctuators
  - Velocity, turn angle, diagnostic tests, pyroblast/ice block
- **S**ensors
  - Distance to lead vehicle, biopsy image, proximity alarm

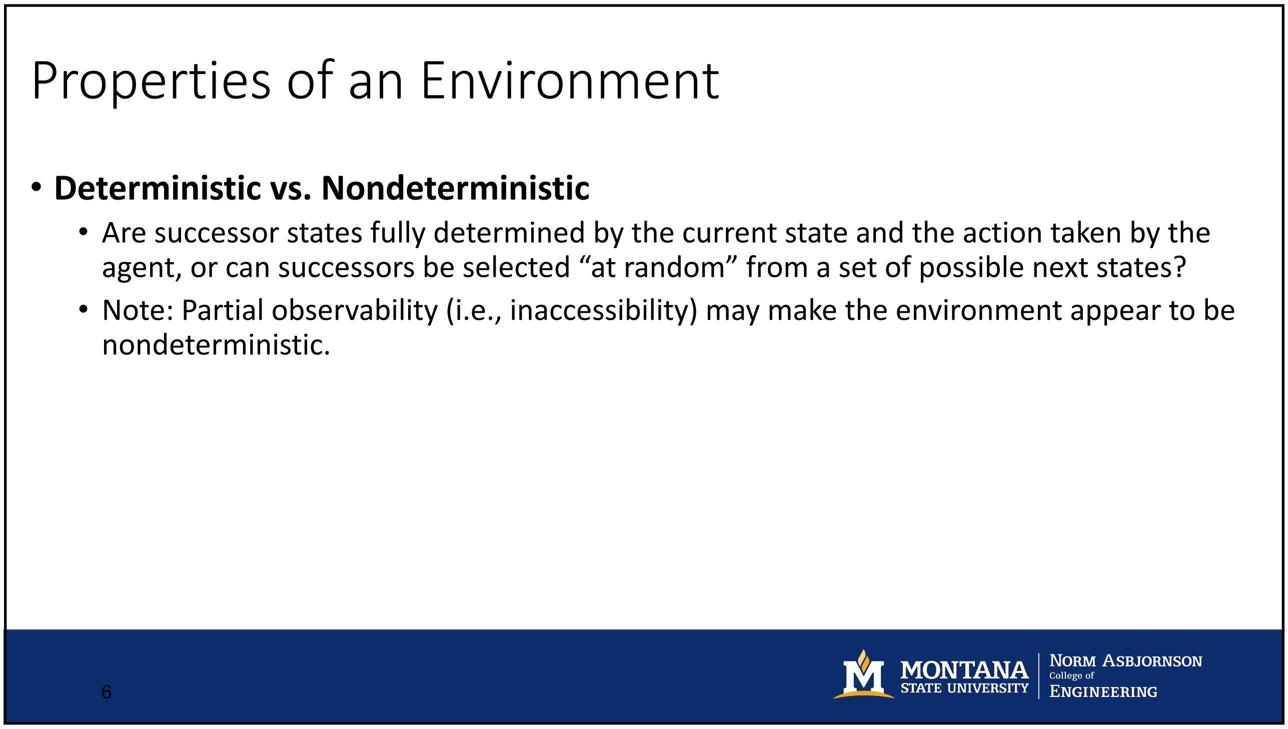
4



Properties of an Environment

- **Accessible vs. Inaccessible**
  - Does the agent's sensors provide sufficient information to fully capture the state of the environment, or is state information only partially captured?

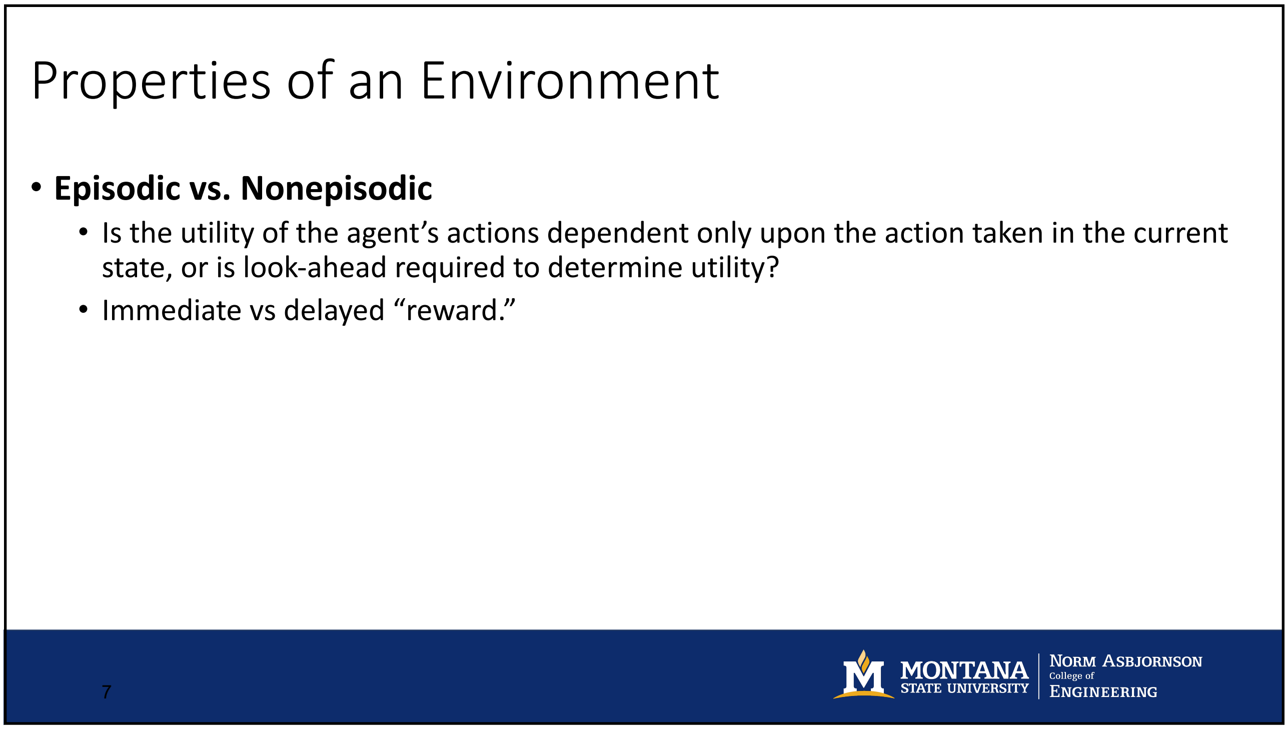
5



Properties of an Environment

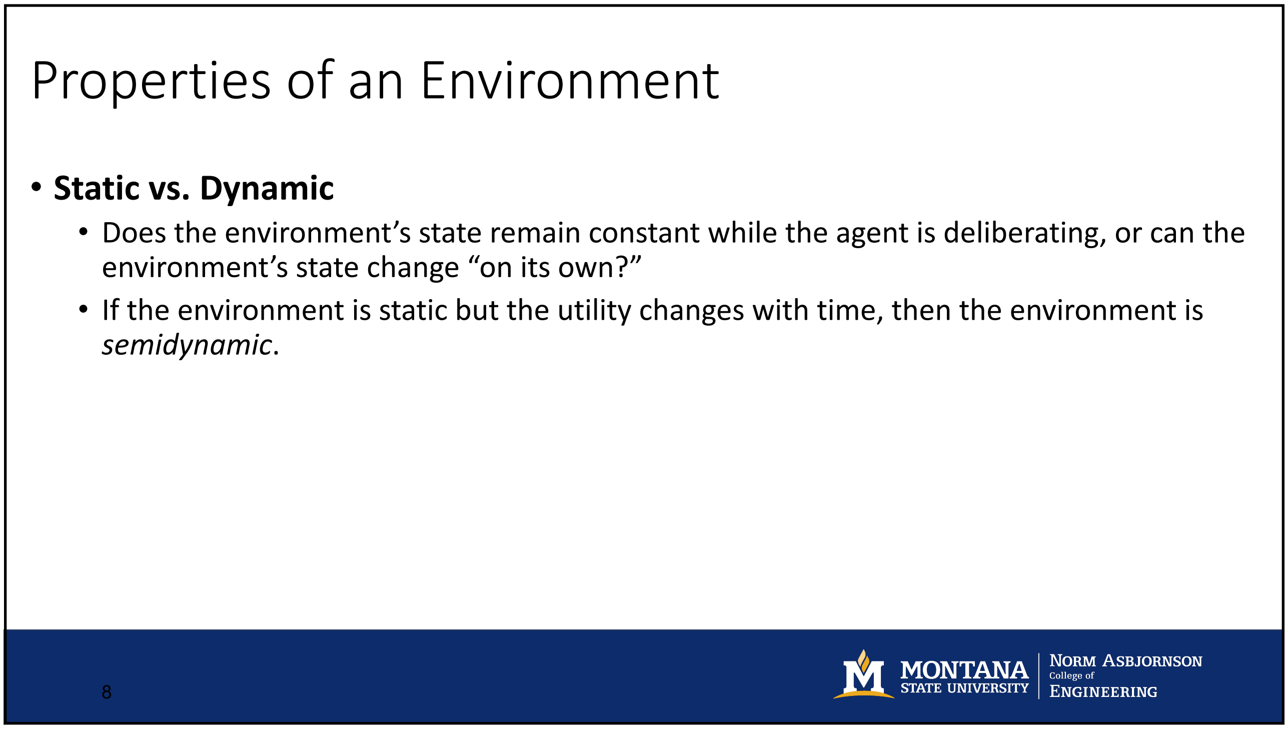
- **Deterministic vs. Nondeterministic**
  - Are successor states fully determined by the current state and the action taken by the agent, or can successors be selected "at random" from a set of possible next states?
  - Note: Partial observability (i.e., inaccessibility) may make the environment appear to be nondeterministic.

6



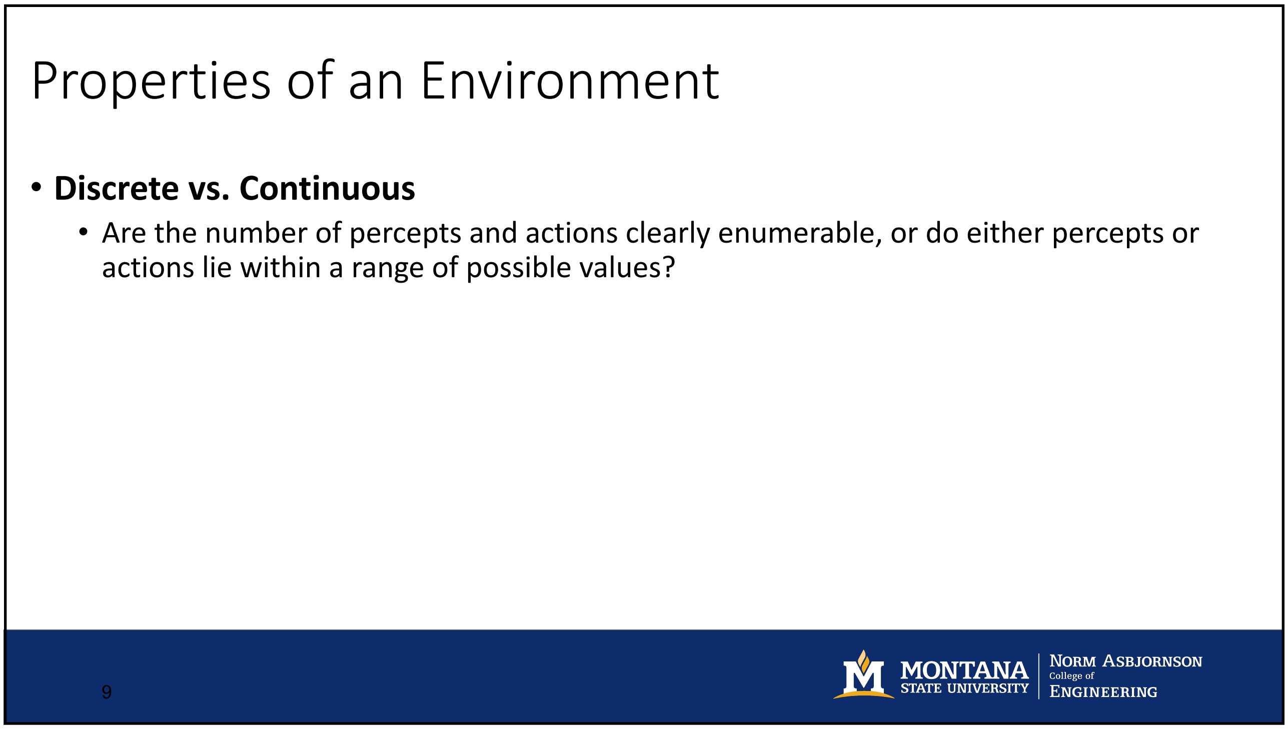
Properties of an Environment

- **Episodic vs. Nonepisodic**
  - Is the utility of the agent’s actions dependent only upon the action taken in the current state, or is look-ahead required to determine utility?
  - Immediate vs delayed “reward.”



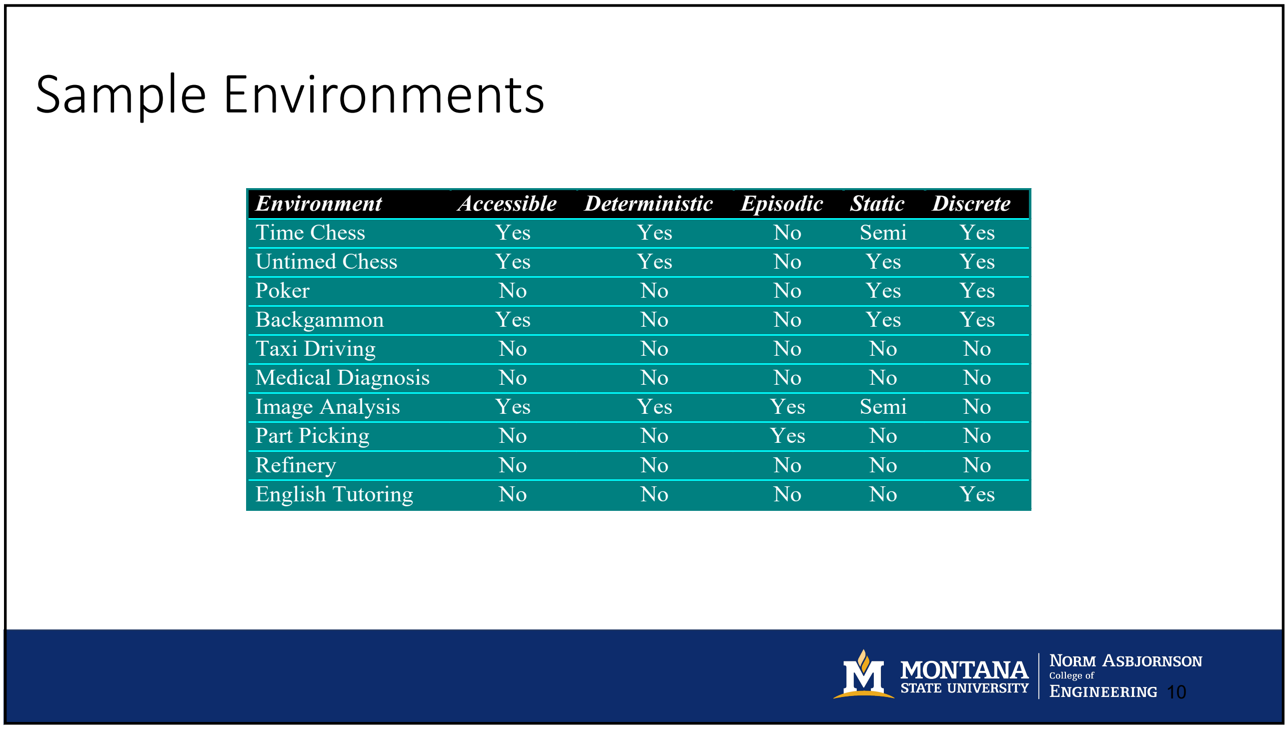
Properties of an Environment

- **Static vs. Dynamic**
  - Does the environment’s state remain constant while the agent is deliberating, or can the environment’s state change “on its own?”
  - If the environment is static but the utility changes with time, then the environment is *semidynamic*.



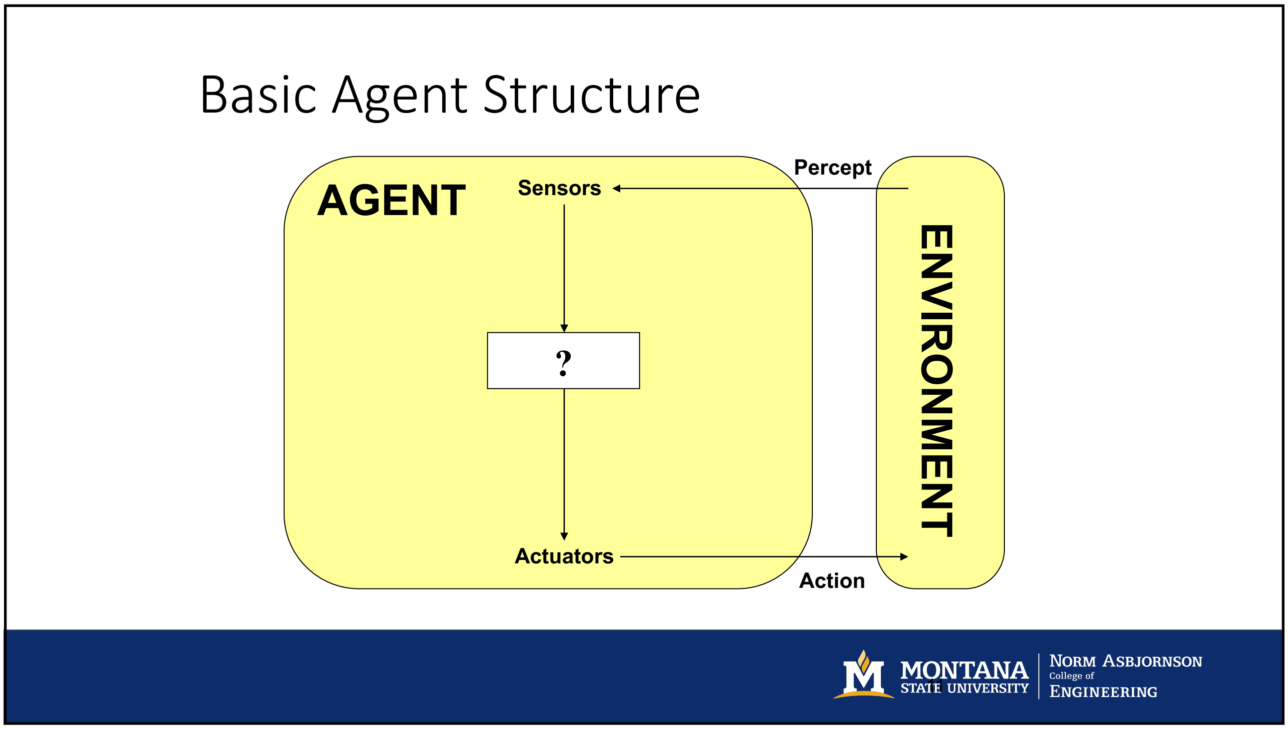
Properties of an Environment

- **Discrete vs. Continuous**
  - Are the number of percepts and actions clearly enumerable, or do either percepts or actions lie within a range of possible values?



Sample Environments

Environment	*Accessible*	*Deterministic*	*Episodic*	*Static*	*Discrete*
Time Chess	Yes	Yes	No	Semi	Yes
Untimed Chess	Yes	Yes	No	Yes	Yes
Poker	No	No	No	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes
Taxi Driving	No	No	No	No	No
Medical Diagnosis	No	No	No	No	No
Image Analysis	Yes	Yes	Yes	Semi	No
Part Picking	No	No	Yes	No	No
Refinery	No	No	No	No	No
English Tutoring	No	No	No	No	Yes



Basic Agent Structure

AGENT

Sensors

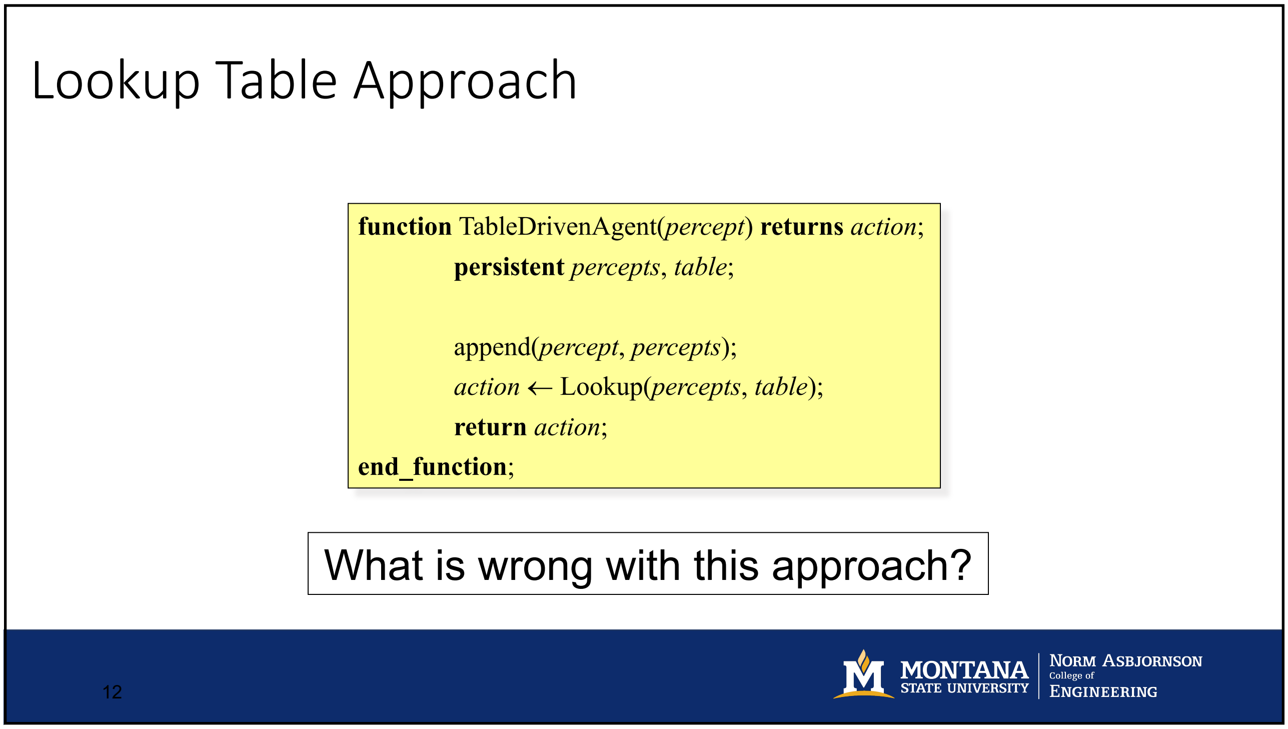
?

Actuators

Percept

Action

ENVIRONMENT

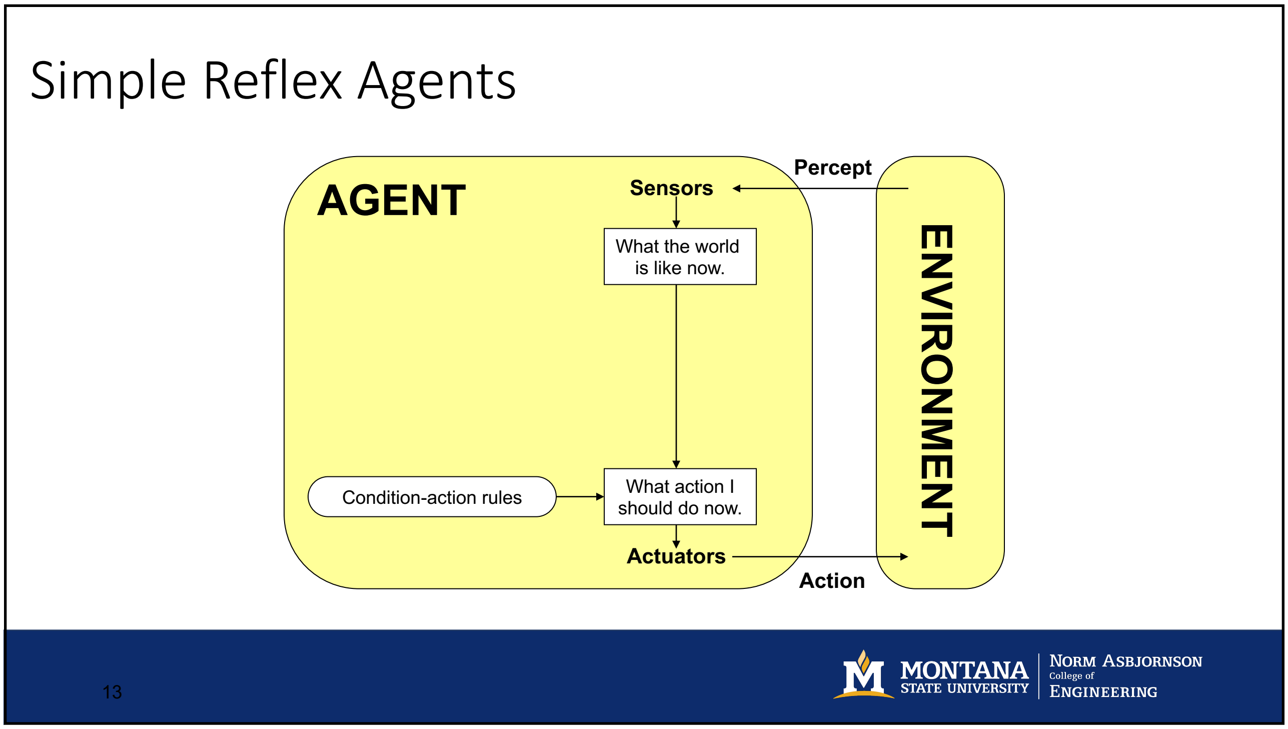


Lookup Table Approach

```
function TableDrivenAgent(percept) returns action;
        persistent percepts, table;

        append(percept, percepts);
        action ← Lookup(percepts, table);
        return action;
end_function;
```

What is wrong with this approach?



Simple Reflex Agents

AGENT

Sensors

What the world is like now.

Condition-action rules

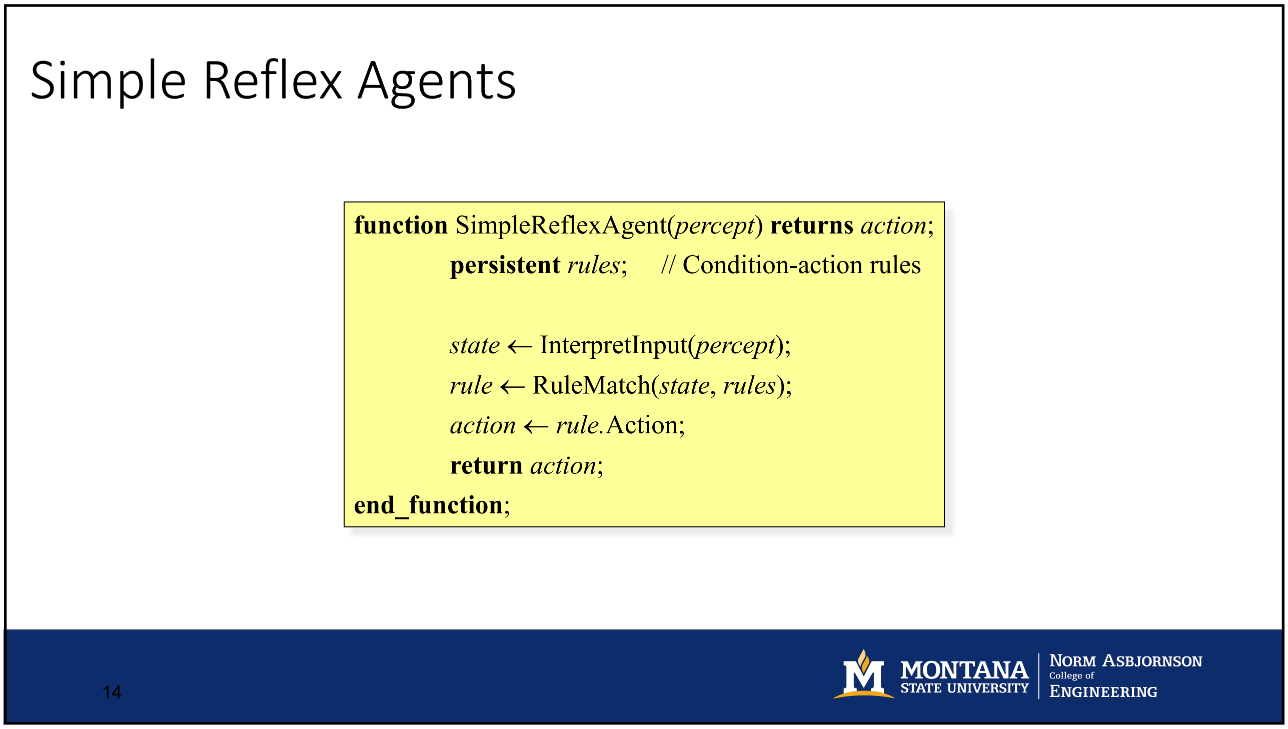
What action I should do now.

Actuators

Percept

Action

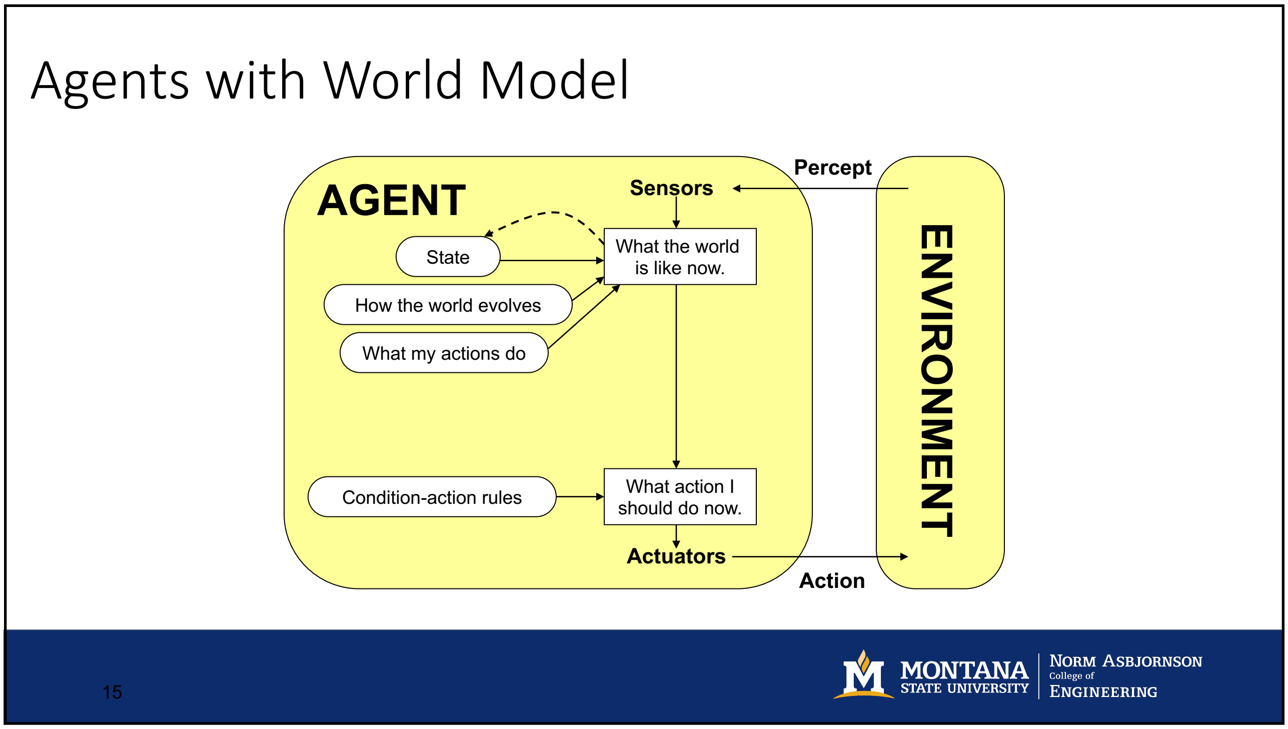
ENVIRONMENT



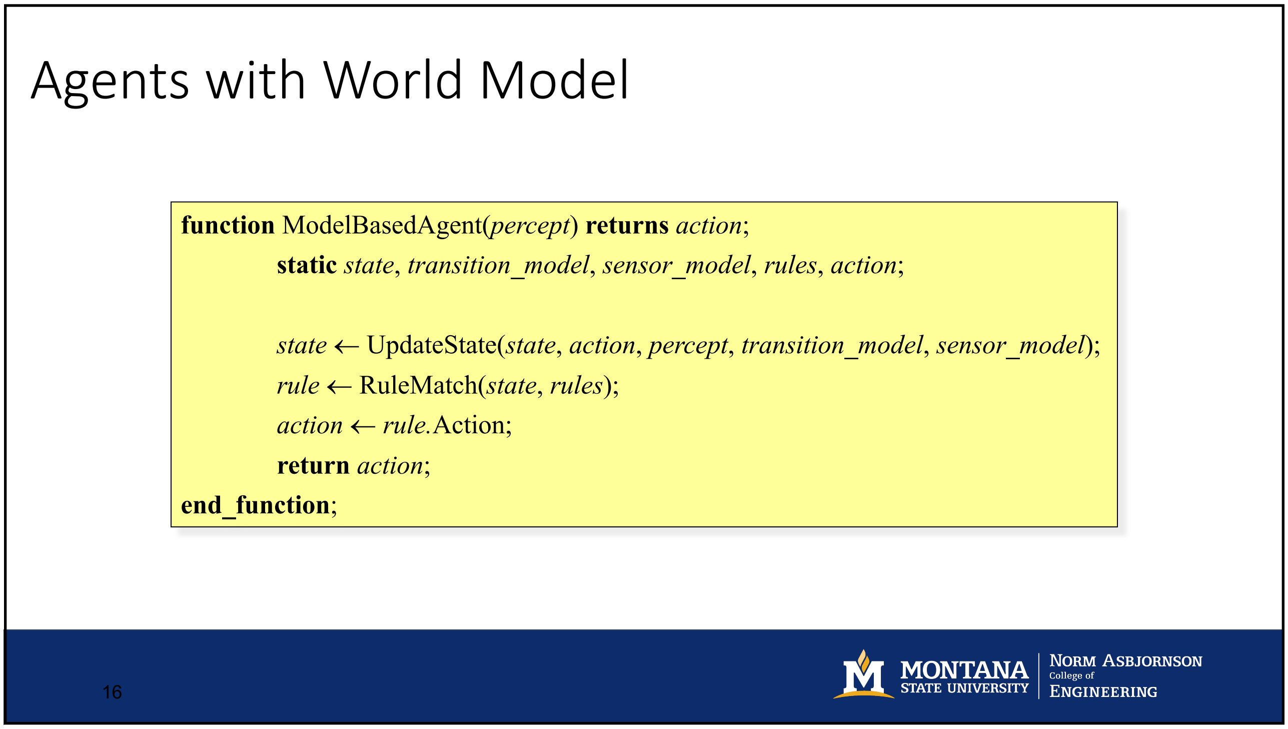
Simple Reflex Agents

```
function SimpleReflexAgent(percept) returns action;
        persistent rules;    // Condition-action rules

        state ← InterpretInput(percept);
        rule ← RuleMatch(state, rules);
        action ← rule.Action;
        return action;
end_function;
```



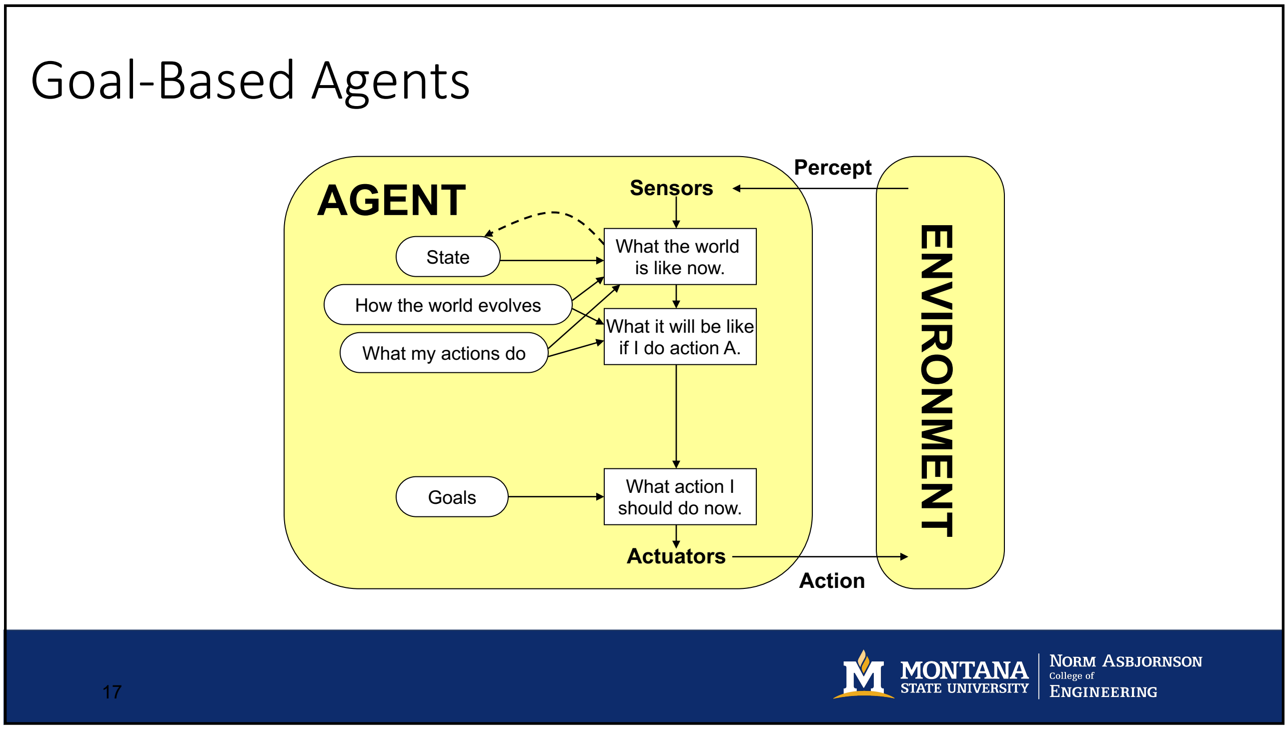
Agents with World Model



Agents with World Model

```
function ModelBasedAgent(percept) returns action;
        static state, transition_model, sensor_model, rules, action;

        state ← UpdateState(state, action, percept, transition_model, sensor_model);
        rule ← RuleMatch(state, rules);
        action ← rule.Action;
        return action;
end_function;
```

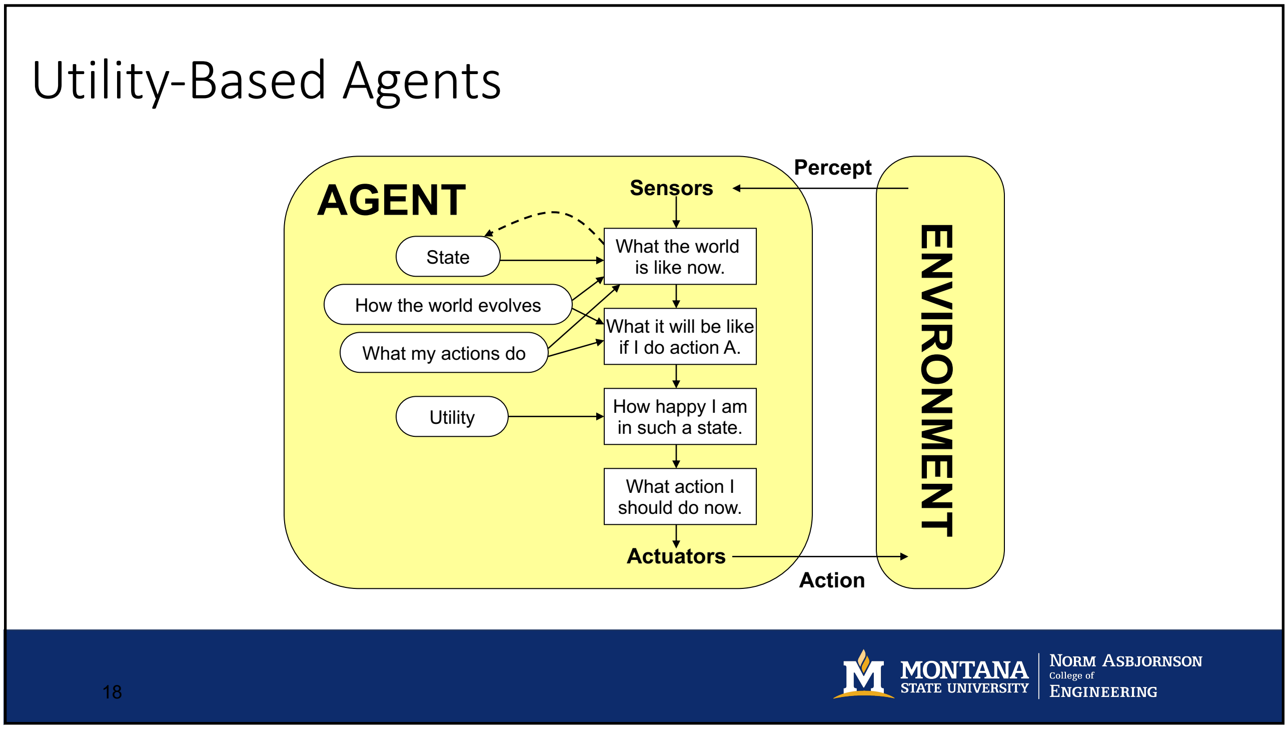


Goal-Based Agents

17

MONTANA STATE UNIVERSITY | Norm Asbjornson College of Engineering

17



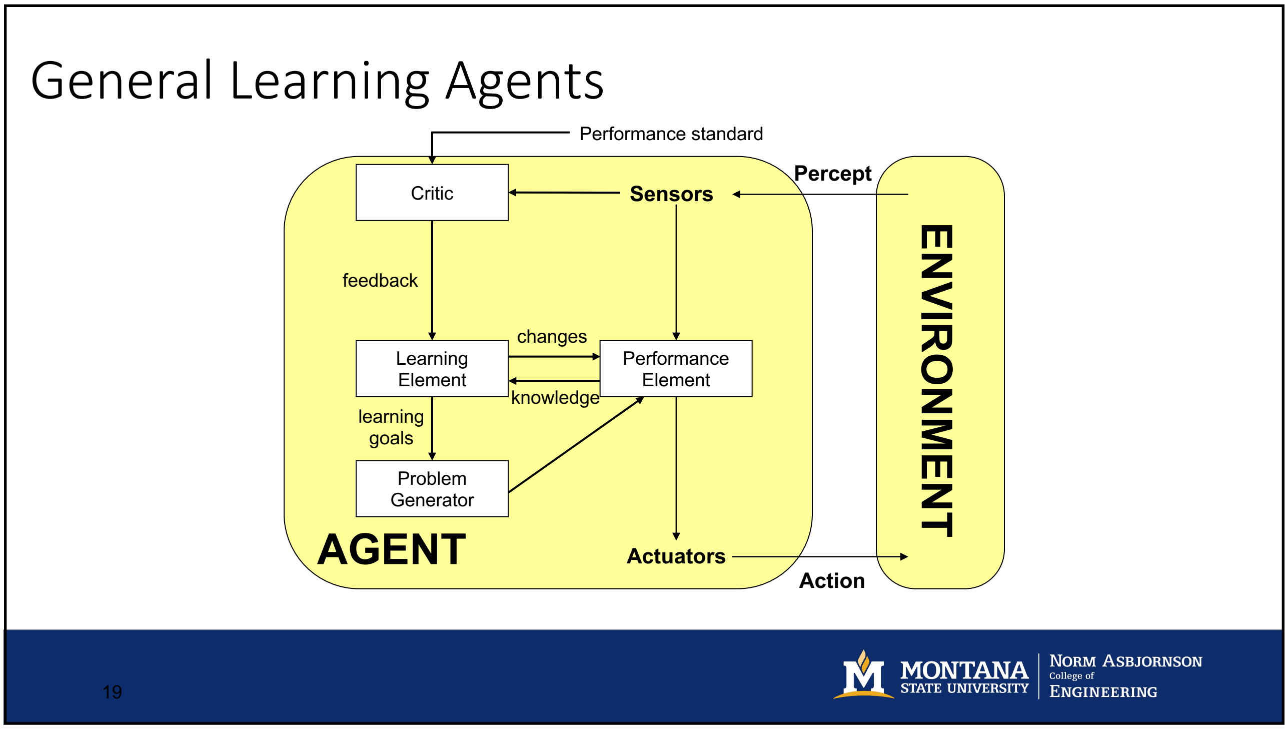
Utility-Based Agents

AGENT
Sensors
Percept
State
What the world is like now.
How the world evolves
What my actions do
What it will be like if I do action A.
Utility
How happy I am in such a state.
What action I should do now.
Actuators
Action
ENVIRONMENT

18

MONTANA STATE UNIVERSITY | Norm Asbjornson College of Engineering

18



General Learning Agents

19