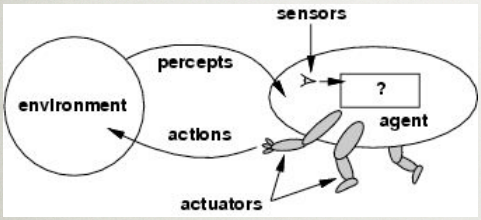
# Agents



# Agents and environments




The *agent function* maps percept sequence to actions

$$f : P^* \rightarrow A$$

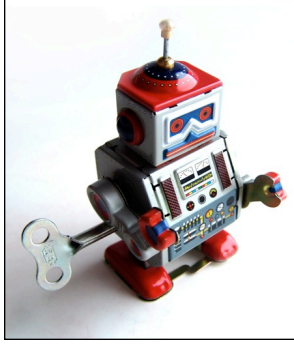
An agent can perceive its own actions, but not always it effects.



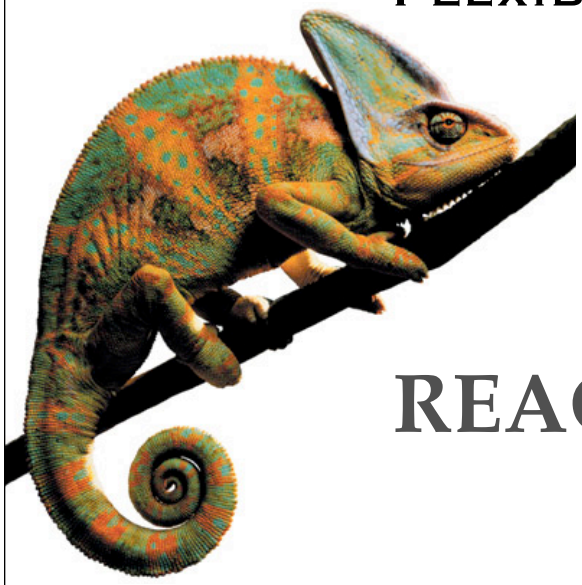

## What is a Software Agent?

- An *"intelligent agent"* is a *persistent* computer system capable of ***flexible*** *autonomous* action in some environment.

*[Wooldridge & Jennings]*



"Flexible"



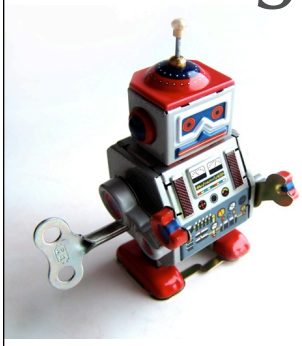
# REACTIVE

5


6

## "Flexible"

SOCIAL

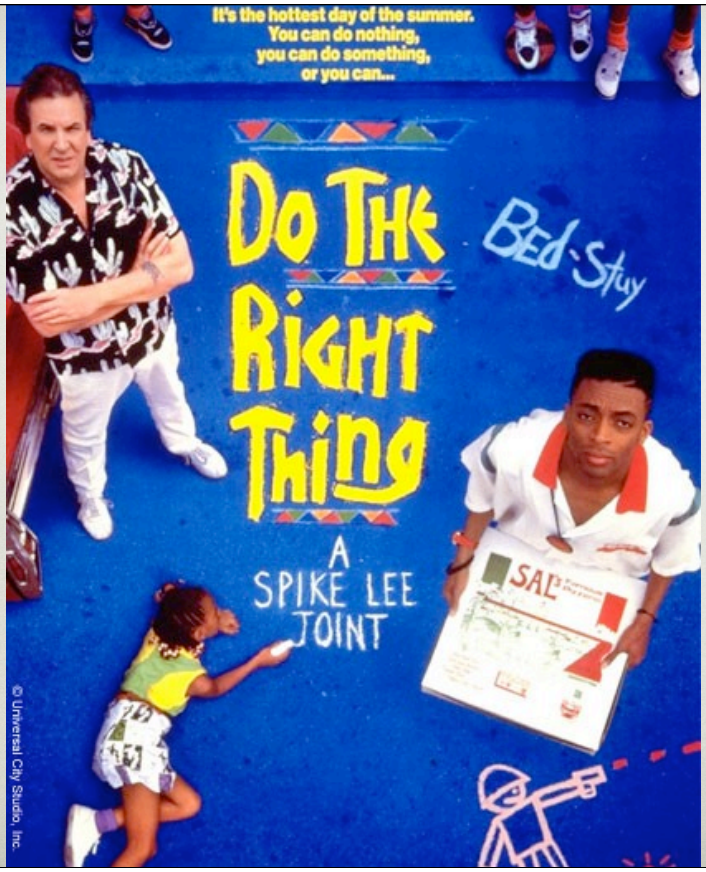


7

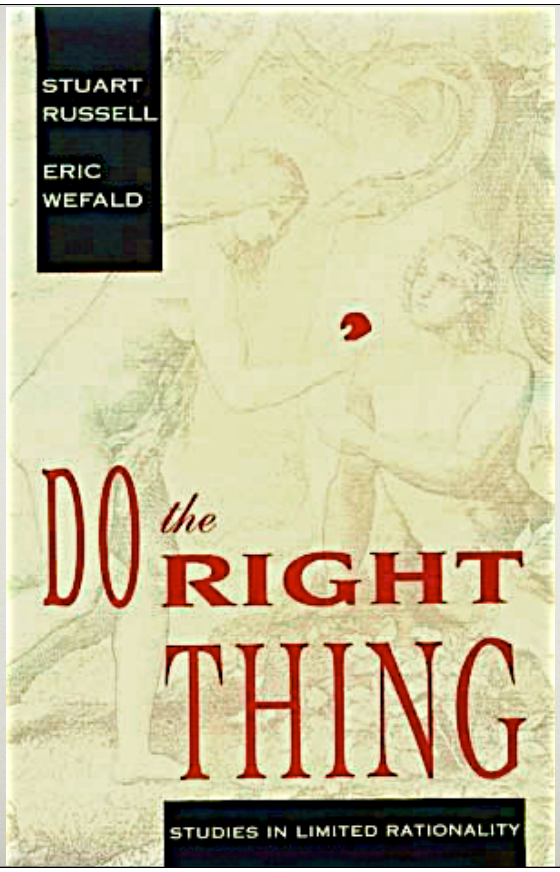
# Rationality

8

8


It's the hottest day of the summer.
You can do nothing,
you can do something,
or you can...
DO THE RIGHT Thing
Bed-Stuy
A SPIKE LEE JOINT
SAL
© Universal City Studio, Inc.

9


STUART RUSSELL
ERIC WEFALD
DO the RIGHT THING
STUDIES IN LIMITED RATIONALITY

10

# Rationality ≠ omniscience



*A **rational agent** chooses whichever **actions** maximize the expected value of the **performance measure** given the **sensed percept sequence** to date and prior **environment knowledge**.*

# P.E.A.S.



## Task Environments

- Performance metric
- Environment
- Actuators (actions)
- Sensors (percepts)

15

16

17

## Task Environments

- *Performance metric*
- **Environment**
- *Actuators (actions)*
- *Sensors (percepts)*



18

# Environment Characteristics

19

# Observability

Full vs. Partial

20

# Deterministic

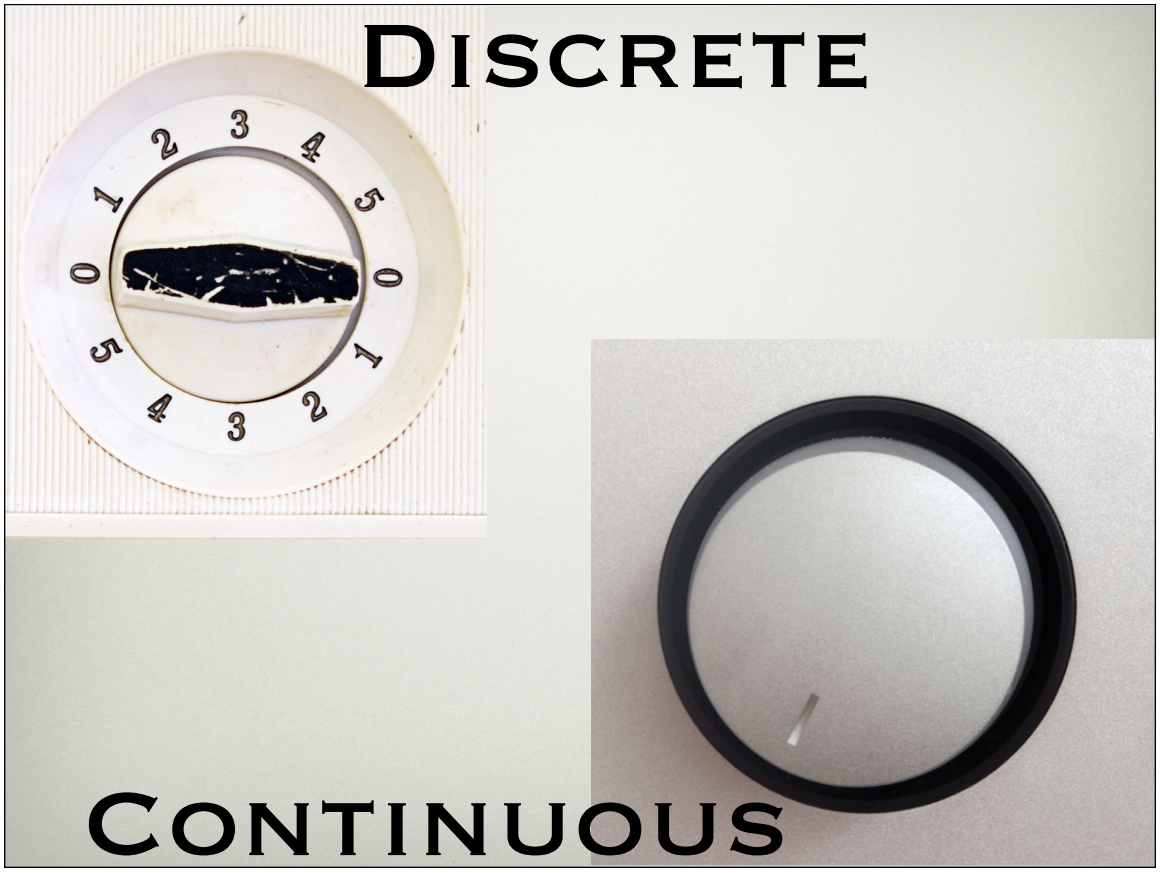
# Stochastic

21

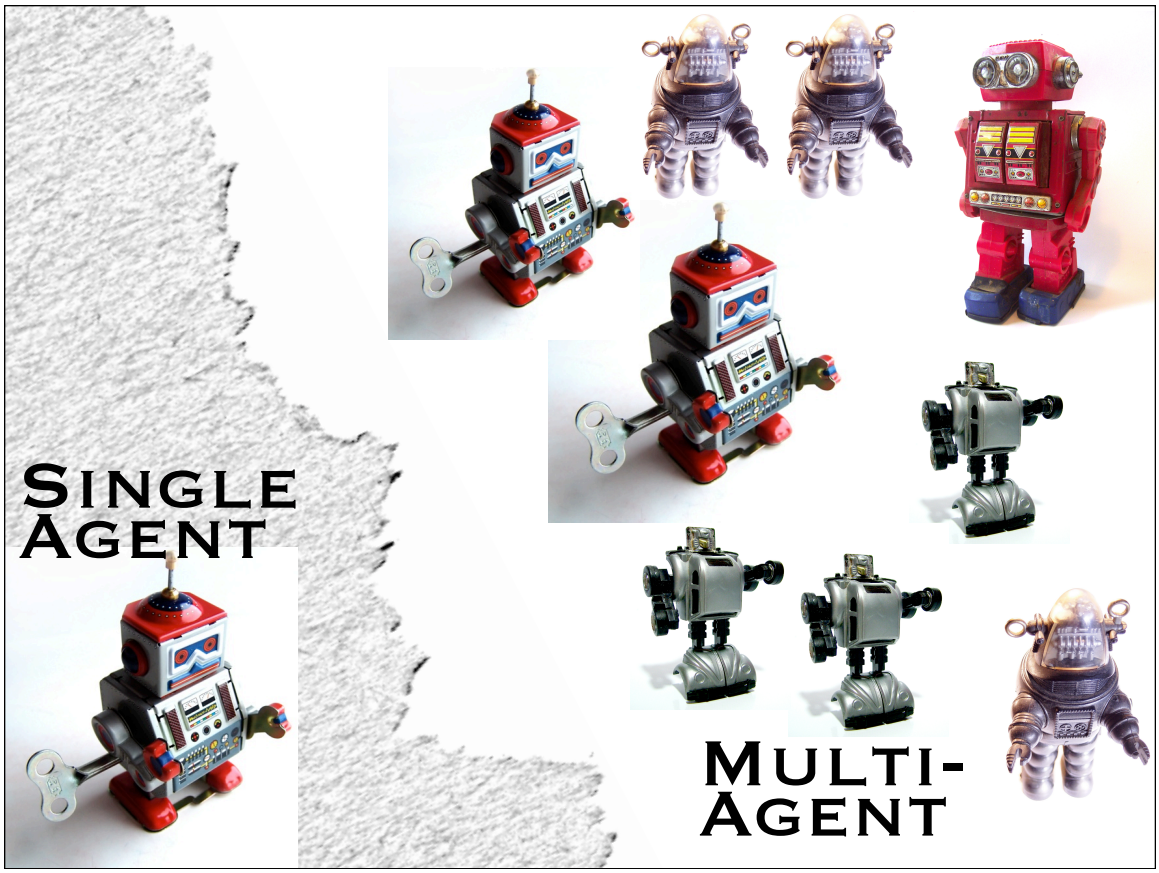
# Episodic vs. Sequential


22


23



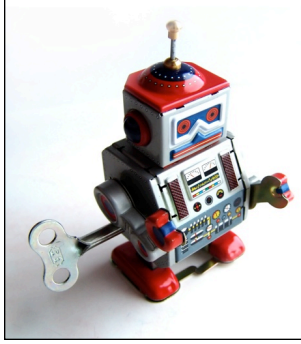

24






# Ideal Rational Agent

*For each possible percept sequence*
*Do whatever Action is expected to maximize performance*
*(Given information in the percept sequence and any built-in knowledge)*



# Agent Programs

27

Agent = Architecture + Program



28

# Use a Table??

**Function** TABLE-DRIVEN_AGENT(*percept*) **returns** an action

**static**: *percepts*, a sequence initially empty

*table*, a table of actions, indexed by percept sequence

append *percept* to the end of *percepts*

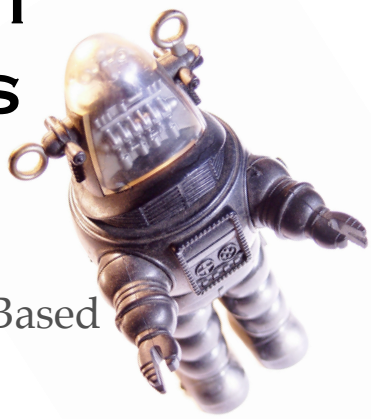
*action* ← LOOKUP(*percepts*, *table*)

**return** *action*



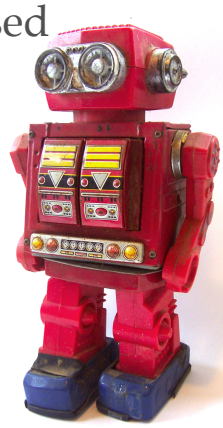


# Agent Types

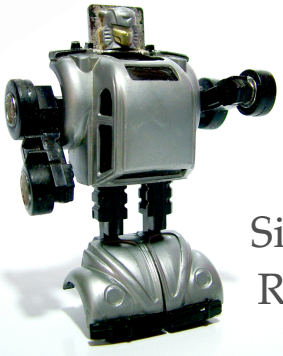


Utility-Based

Model-Based Reflex

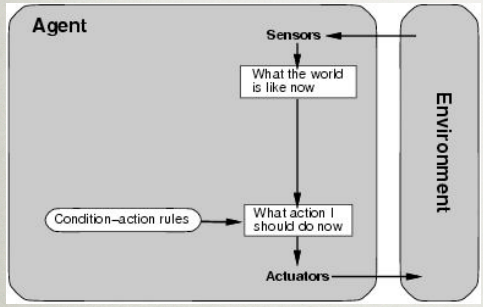


Goal-Based



Simple Reflex

# SIMPLE REFLEX




Select action on the basis of *only the current* percept.

- E.g. the vacuum-agent

Large reduction in possible percept/action situations (next page).

Implemented through *condition-action rules*

- If dirty then suck





# SIMPLE REFLEX

**function** SIMPLE-REFLEX-AGENT(*percept*) **returns** an action

**static**: *rules*, a set of condition-action rules

*state* ← INTERPRET-INPUT(*percept*)

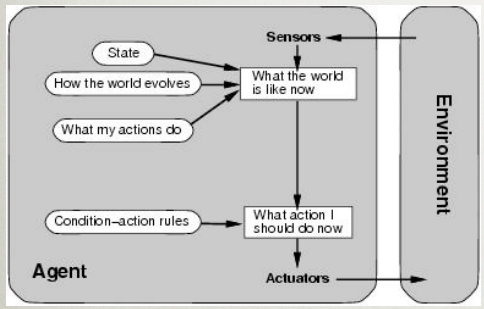
*rule* ← RULE-MATCH(*state*, *rule*)

*action* ← RULE-ACTION[*rule*]

return *action*

Will only work if the environment is *fully observable* otherwise infinite loops may occur.

# Model-Based Reflex




To tackle *partially observable* environments.

- Maintain internal state

Over time update state using world knowledge

- How does the world change.
- How do actions affect world.

⇒ *Model of World*





# Model-Based Reflex

**function** REFLEX-AGENT-WITH-STATE(*percept*) **returns** an action

**static**: *rules*, a set of condition-action rules

*state*, a description of the current world state

*action*, the most recent action.

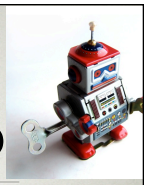
*state* ← UPDATE-STATE(*state*, *action*, *percept*)

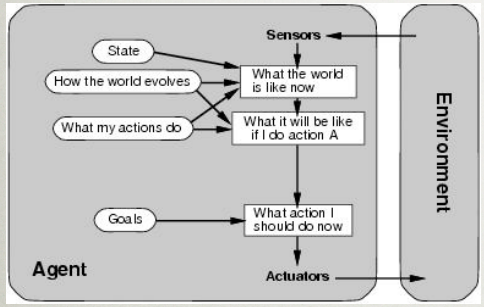
*rule* ← RULE-MATCH(*state*, *rule*)

*action* ← RULE-ACTION[*rule*]

return *action*

# Agent types; goal-based






The agent needs a goal to know which situations are *desirable*.

- Things become difficult when long sequences of actions are required to find the goal.

Typically investigated in **search** and **planning** research.

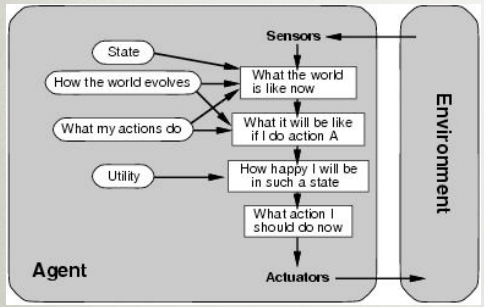
Major difference: future is taken into account

Is more flexible since knowledge is represented explicitly and can be manipulated.





# Agent types; utility-based




Certain goals can be reached in different ways.

- Some are better, have a higher utility.

Utility function maps a (sequence of) state(s) onto a real number.

Improves on goals:

- Selecting between conflicting goals
- Select appropriately between several goals based on likelihood of success.

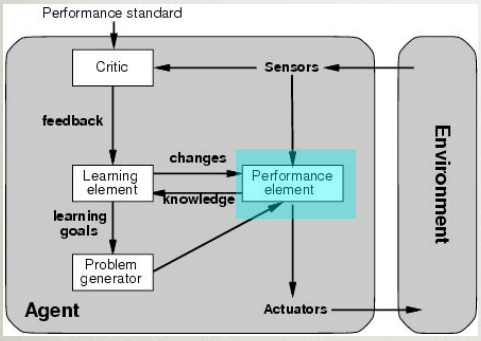
# LEARNING

- Previous agent-programs describe methods for selecting **actions**.
  - *Where do these programs come from?*





# AGENT TYPES; LEARNING




*Performance element*: selecting actions based on percepts.

- Corresponds to the previous agent programs

*Learning element*: introduce improvements in performance element.

- Critic provides feedback on agents performance based on fixed performance standard.

*Problem generator*: suggests actions that will lead to new and informative experiences.

- Exploration vs. exploitation