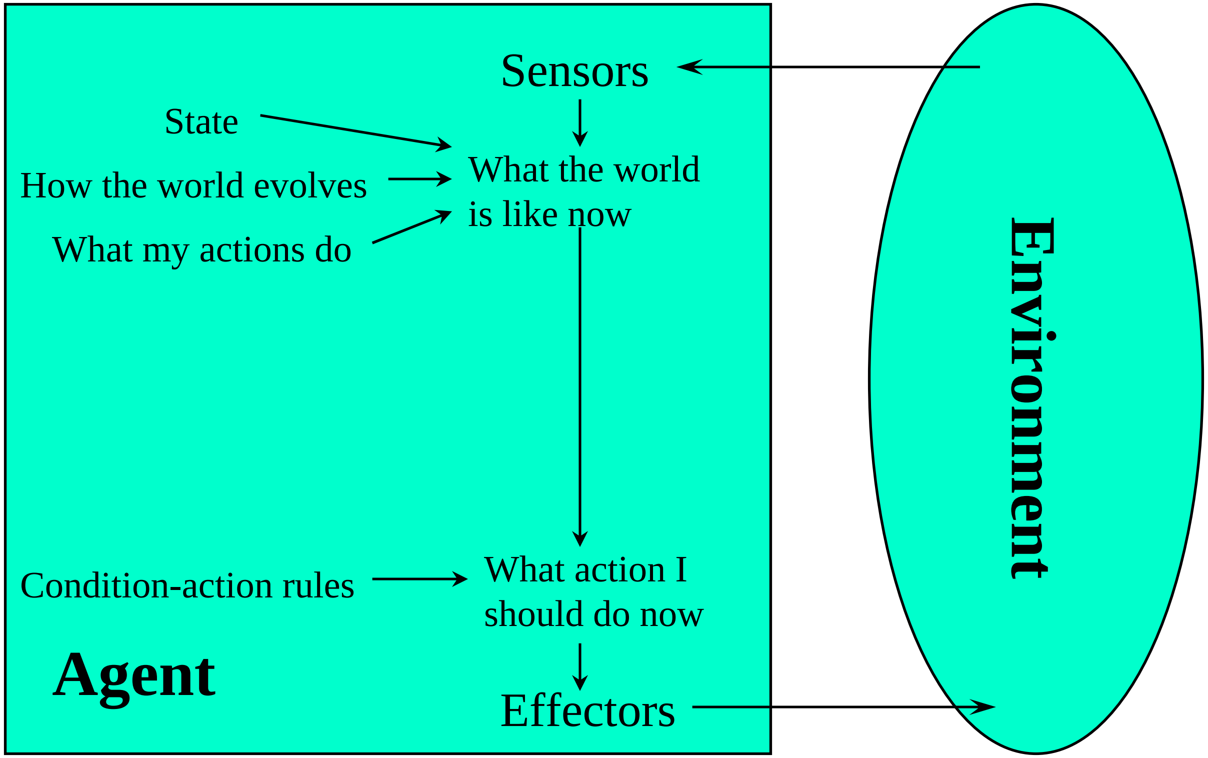
Agent Architectures (cont.)

Logic-Based Agents

Agent Architectures (recap)

- We want to build agents, that enjoy the properties of **autonomy, reactiveness, pro-activeness, and social ability** that we talked about earlier
- This is the area of ***agent architectures***
- Patty Maes from MIT defines an agent architecture as: ‘[A] particular methodology for building [agents]. It specifies how… the agent can be decomposed into the construction of a set of component modules and how these modules should be made to interact. The total set of modules and their interactions has to provide an answer to the question of **how the sensor data and the current internal state of the agent determine the actions… and future internal state of the agent**. An architecture encompasses techniques and algorithms that support this methodology.’

A Reactive Agent with State

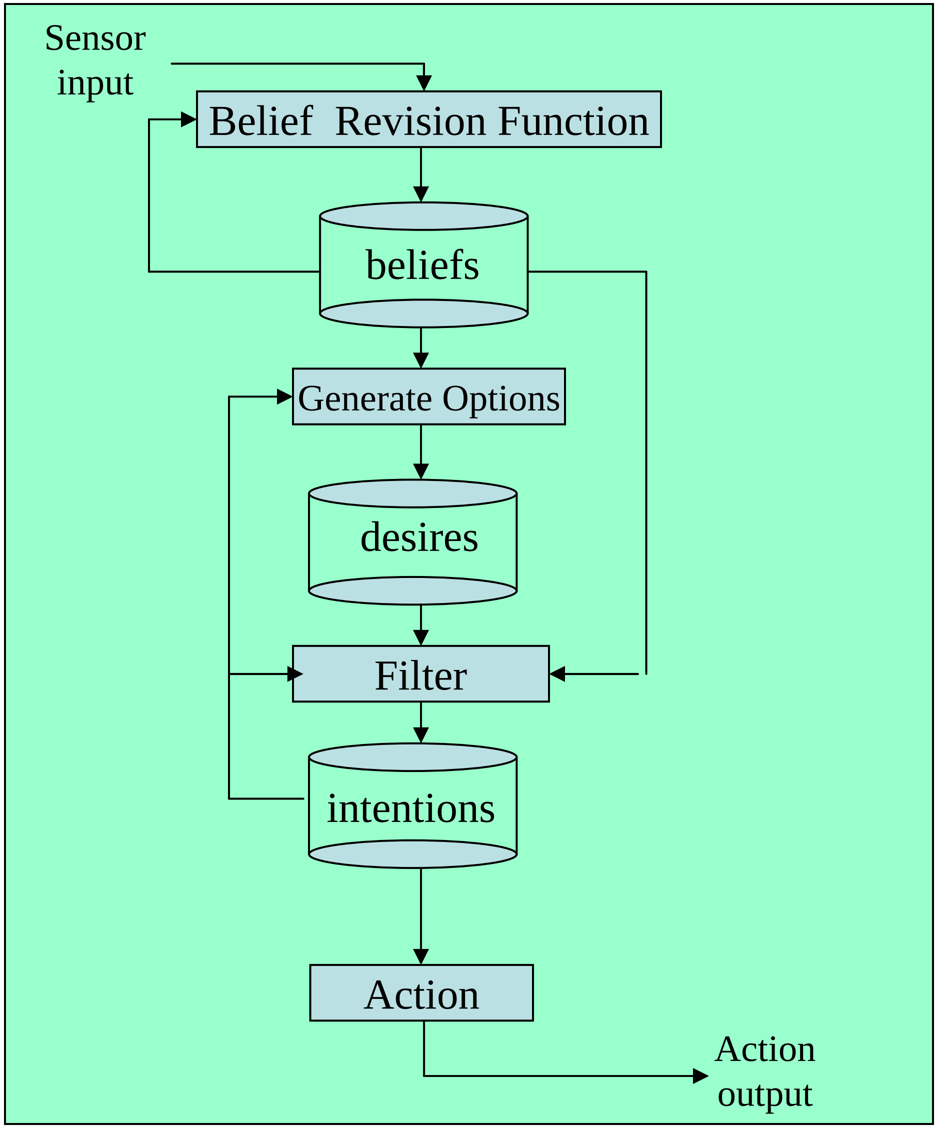


Agent Architectures

- Kaelbling considers an agent architecture to be:

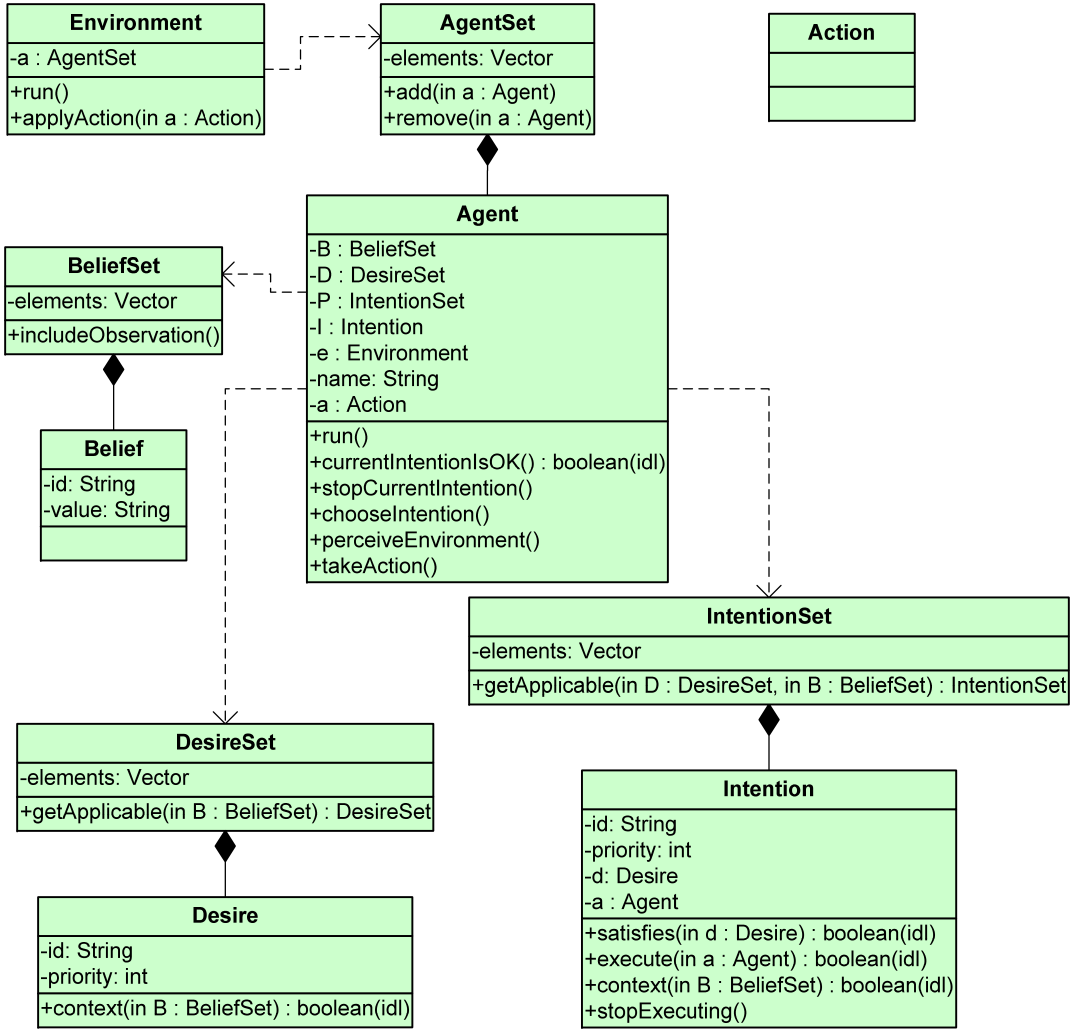
  ‘[A] specific collection of software (or hardware) modules, typically designated by boxes with arrows indicating the data and control flow among the modules. A more abstract view of an architecture is as a general methodology for designing particular modular decompositions for particular tasks.’

Generic BDI Architecture



Generating options and filtering options are together called *deliberation*

Architecture of BDI-Based Agent



Execution Cycle:

1. New information arrives that updates beliefs and goals
2. Actions are triggered by new beliefs or goals
3. A triggered action is intended
4. An intended action is selected
5. The selected intention is activated
6. An action is performed
7. New beliefs or goals are stored
8. Intentions are updated

Symbolic Reasoning Agents

- The **classical approach** to building agents is to view them as a particular type of ***knowledge-based system***, and bring all the associated (discredited?!) methodologies of such systems to bear
- This paradigm is known as *symbolic AI*
- We define **a deliberative agent or agent architecture** to be one that:
  - contains an explicitly represented, *symbolic model of the world*
  - makes decisions (for example about what actions to perform) *via symbolic reasoning*

Agent Architectures

- Originally (1956-1985), pretty much all agents designed within AI were *symbolic reasoning* agents
- Its purest expression proposes that agents use ***explicit logical reasoning*** in order to decide what to do
- Problems with symbolic reasoning led to a reaction against this — the so-called *reactive agents* movement, 1985–present
- From 1990-present, a number of alternatives proposed: *hybrid* architectures, which attempt to combine the best of reasoning and reactive architectures

Logic-Based Agents

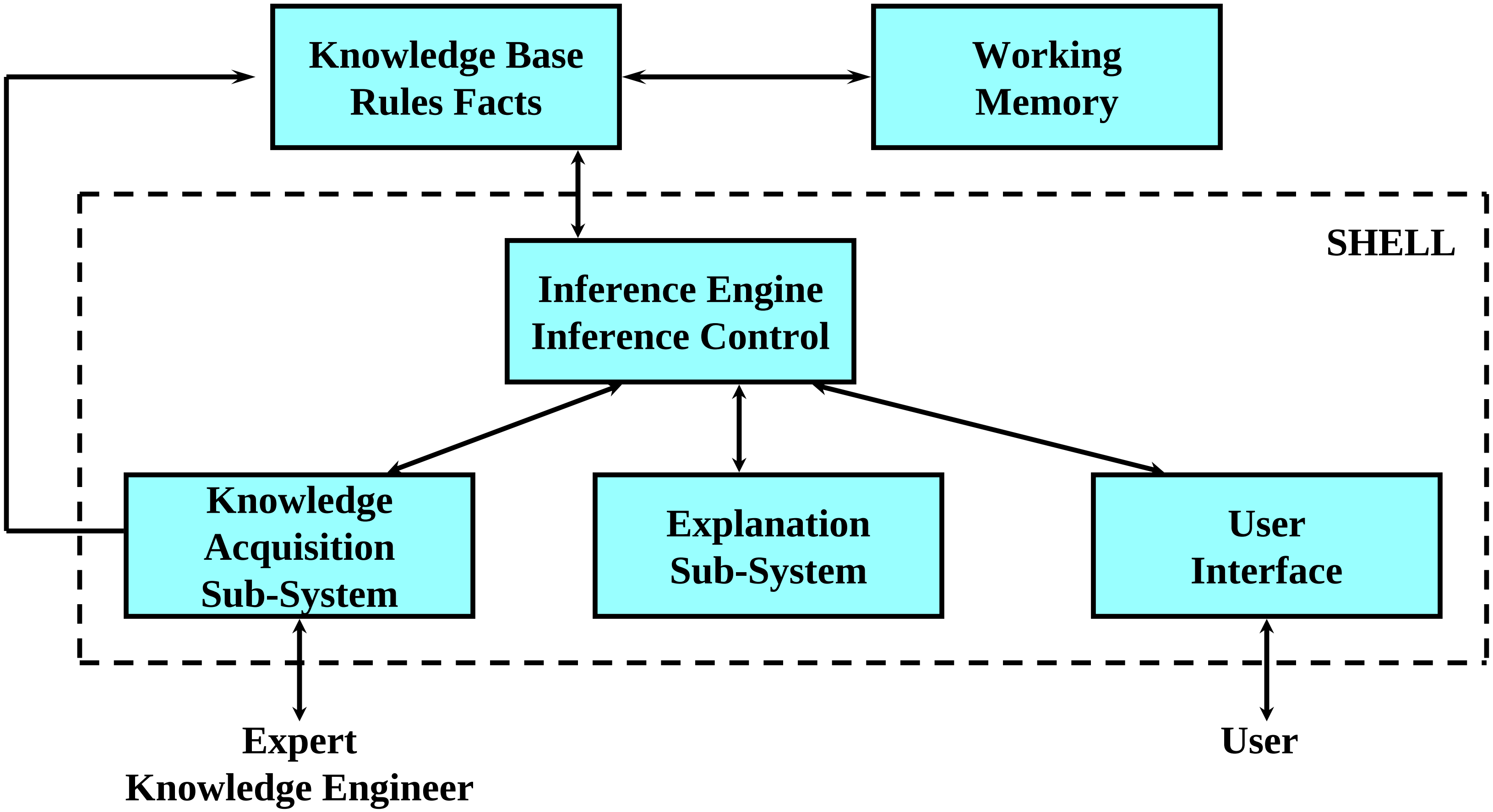
Decision making is realized through logical deduction

Agent viewed as a kind of **knowledge-based system**

- Contains an explicitly represented symbolic model of the world
- Takes decisions via symbolic reasoning

Problems:

- Translating the real world into an accurate and adequate symbolic description, in real-time
- How to symbolically represent information about complex real-world entities



Knowledge Base
Rules Facts

Working
Memory

SHELL

Inference Engine
Inference Control

Knowledge
Acquisition
Sub-System

Explanation
Sub-System

User
Interface

Expert
Knowledge Engineer

User

Architecture of a Knowledge-Based Expert System

Agent architectural types (recap)

An agent architecture describes how an agent can be built

- Reactive
- Belief-Desire-Intention
- **Logic-based**
- Hybrid

Logic-based architecture

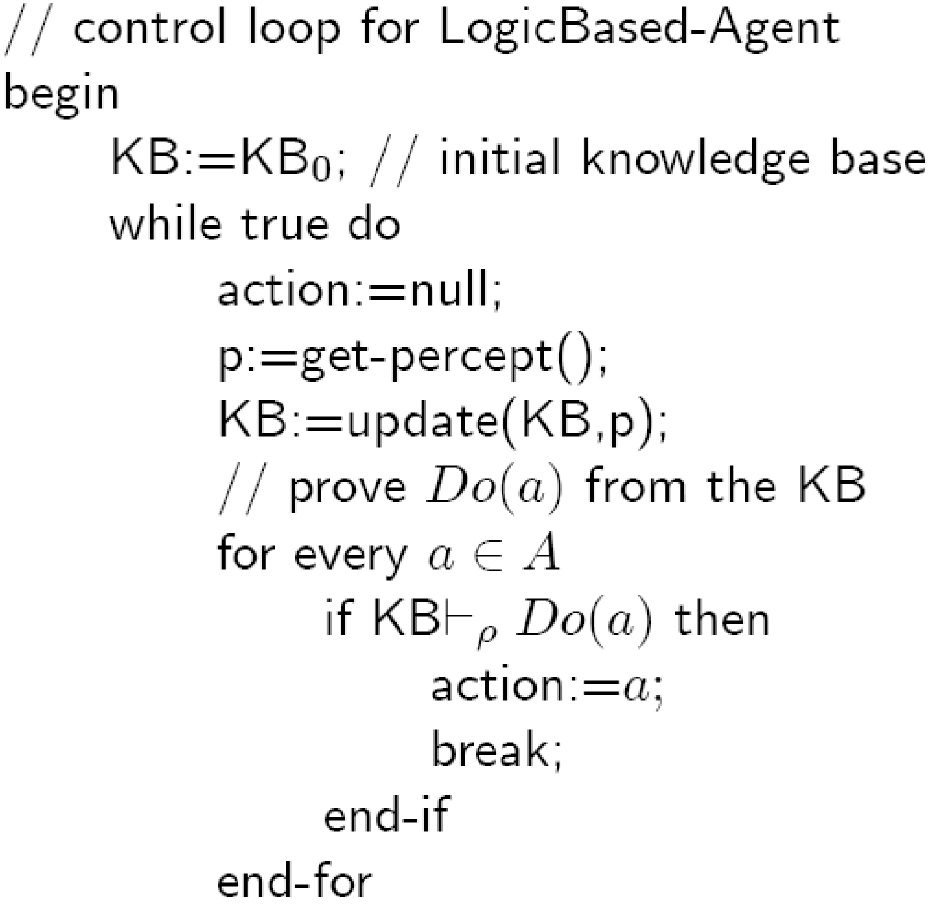
- The ‘traditional’ symbolic artificial intelligence approach
- The agent possesses a symbolic representation of its environment (logical formulas) and rules on how it should behave and what actions it can take
- The behaviour of the system is generated by syntactic manipulation of the symbolic representations (logical deduction)
- Agent execution as theorem proving: If there is a theory ϕ that explains how the agent behaves, how goals are generated and how the agent can take action to satisfy them, then this specification can be directly executed to produce behaviour

- The environment is described by sentences in **L, $KB= P(L)$**
- At every moment in time t an agent’s internal state is $KB_t \in KB$
- Environment states $S=\{s_1, s_2, \ldots\}$
- Perception $see:S \rightarrow P$
- The agent’s internal state is updated by percepts:

  $next:KB \times P \rightarrow KB$
- An agent can choose an action from a set

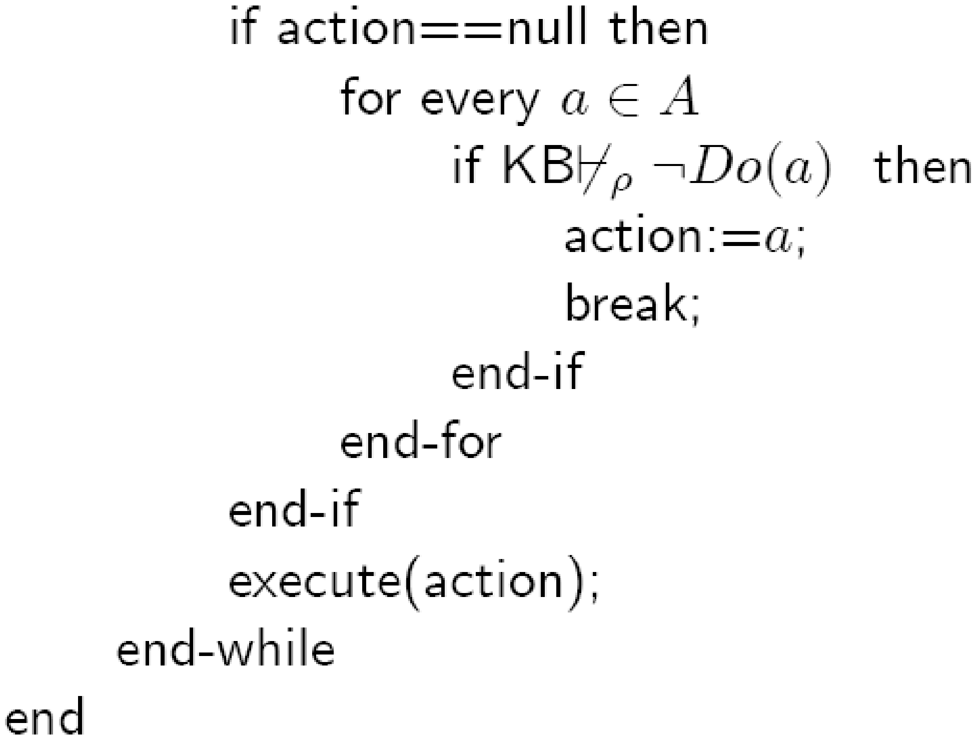
- The agent’s decision-making process is modelled through the rules of inference ρ
- $KB \Rightarrow_{\rho} \phi$: ϕ can be proven from the inference rules ρ
- **The agent programmer has to encode the inference rules ρ in a way that enables the agent to decide what to do**

The control loop for a logic-based agent take



```
// control loop for LogicBased-Agent
begin
    KB:=KB$_0$; // initial knowledge base
    while true do
        action:=null;
        p:=get-percept();
        KB:=update(KB,p);
        // prove $Do(a)$ from the KB
        for every $a \in A$
            if KB$\vdash_\rho Do(a)$ then
                action:=$a$;
                break;
            end-if
        end-for
```

If *Do(a)* cannot be proven, find an action that is consistent with *KB*



```
            if action==null then
                for every $a \in A$
                    if KB$\not\vdash_\rho \neg Do(a)$ then
                        action:=$a$;
                        break;
                    end-if
                end-for
            end-if
            execute(action);
        end-while
    end
```

Symbolic Reasoning Agents

- If we aim to build an agent in this way, there are two key problems to be solved:

1. ***The transduction problem*****:** that of translating the real world into an accurate, adequate symbolic description, in time for that description to be useful…vision, speech understanding, learning
2. ***The representation/reasoning problem*****:** that of how to symbolically represent information about complex real-world entities and processes, and how to get agents to reason with this information in time for the results to be useful…knowledge representation, automated reasoning, automatic planning

Symbolic Reasoning Agents

- Most researchers accept that neither problem is anywhere near solved
- Underlying problem lies with the complexity of symbol manipulation algorithms in general: many (most) search-based symbol manipulation algorithms of interest are *highly intractable*
- Because of these problems, some researchers have looked to alternative techniques for building agents – we discussed them at last lecture.
- **CONTINUE VISITING Prof. Fleming’s**

Deductive Reasoning Agents

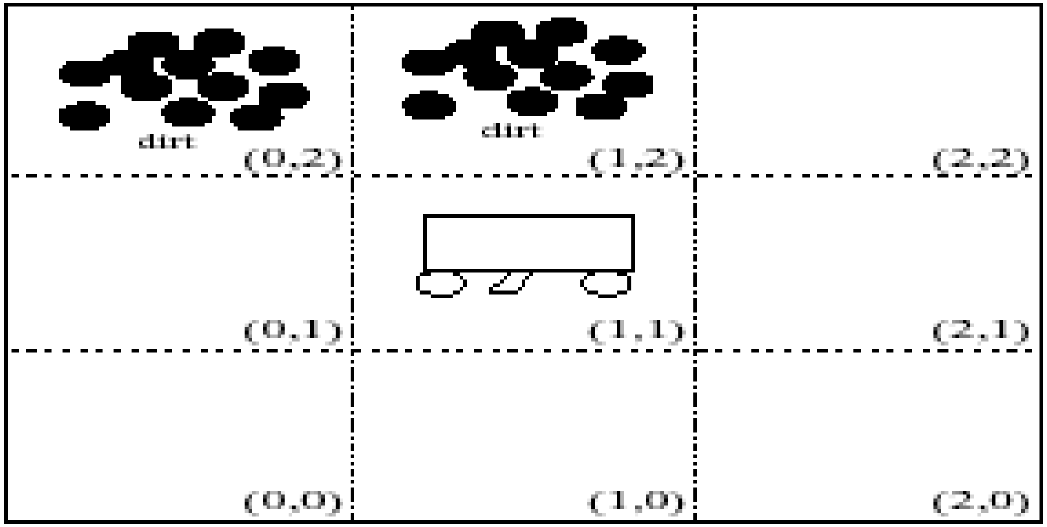
- How can an agent decide what to do using theorem proving?
- Basic idea is to use logic to encode a theory stating the *best* action to perform in any given situation
- Let:
  - ρ be this theory (typically a set of rules)
  - Δ be a logical database that describes the current state of the world
  - Ac be the set of actions the agent can perform
  - Δ ✿$_{\rho}\phi$ mean that ϕ can be proved from Δ using ρ

Deductive Reasoning Agents

```
/* try to find an action explicitly prescribed */
for each $a \in Ac$ do
    if $\Delta$ ✿$_{\rho}$ $Do(a)$ then
        return $a$
    end-if
end-for
/* try to find an action not excluded */
for each $a \in Ac$ do
    if $\Delta$ $\not$✿$_{\rho}$ $\neg Do(a)$ then
        return $a$
    end-if
end-for
return $null$ /* no action found */
```

Deductive Reasoning Agents

- An example: The Vacuum World
- Goal is for the robot to clear up all dirt



Deductive Reasoning Agents

- Use 3 *domain predicates* to solve problem:

  $In(x, y)$ agent is at (x, y)

  $Dirt(x, y)$ there is dirt at (x, y)

  $Facing(d)$ the agent is facing direction d

- Possible actions:

  $Ac = \{turn, forward, suck\}$

P.S. *turn* means “turn right”

Deductive Reasoning Agents

- Rules ρ for determining what to do:

$$
\begin{aligned}
In(0,0) \wedge Facing(north) \wedge \neg Dirt(0,0) &\longrightarrow Do(forward)\\
In(0,1) \wedge Facing(north) \wedge \neg Dirt(0,1) &\longrightarrow Do(forward)\\
In(0,2) \wedge Facing(north) \wedge \neg Dirt(0,2) &\longrightarrow Do(turn)\\
In(0,2) \wedge Facing(east) &\longrightarrow Do(forward)
\end{aligned}
$$

- …and so on!
- Using these rules (+ other obvious ones), starting at $(0, 0)$ the robot will clear up dirt

Deductive Reasoning Agents

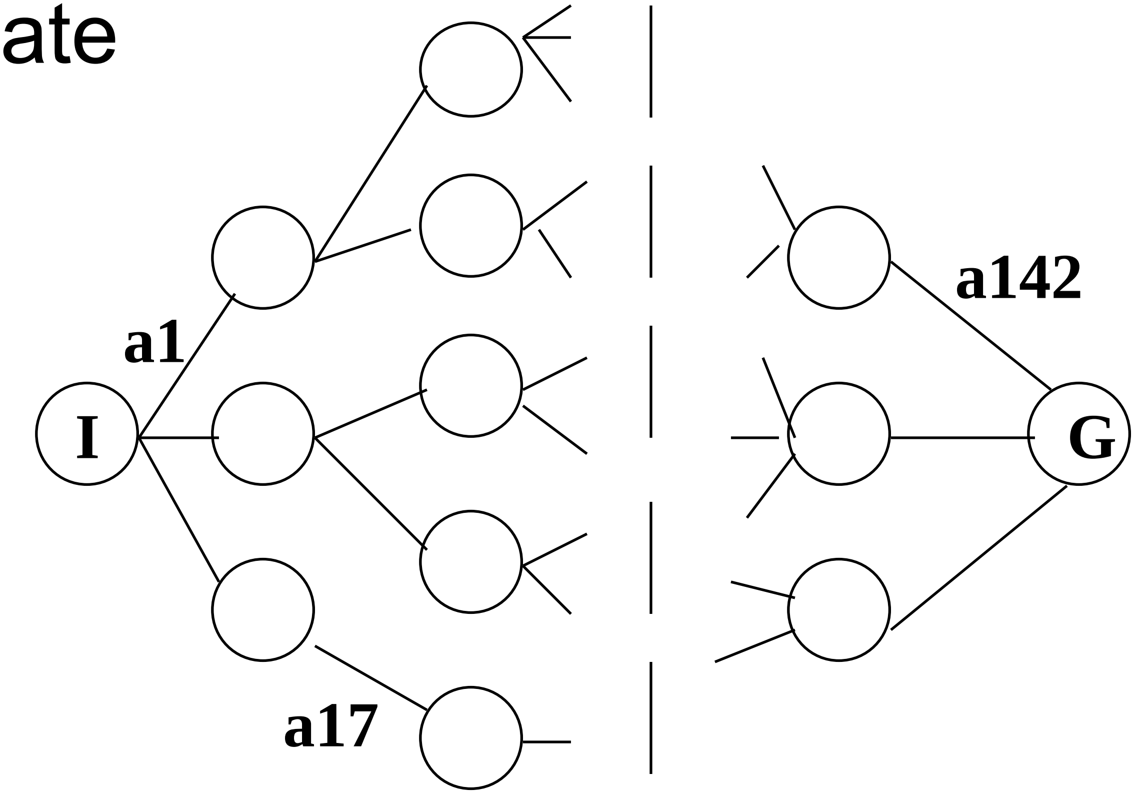
- **Problems**:
  - How to convert video camera input to *Dirt*(0, 1)?
  - decision making assumes a *static* environment: *calculative* rationality
  - decision making using first-order logic is *undecidable*!
- Even where we use *propositional* logic, decision making in the worst case means solving co-NP-complete problems
  (PS: co-NP-complete = bad news!)
- Typical solutions:
  - weaken the logic
  - use symbolic, non-logical representations
  - shift the emphasis of reasoning from *run time* to *design time*
- **We will look at some examples of these approaches in this class**

More Problems…

- The “logical approach” that was presented implies adding and removing things from a database
- That’s not pure logic
- Early attempts at creating a “planning agent” tried to use true logical deduction to the solve the problem

Planning Systems (in general)

- Planning systems find a sequence of actions that transforms an initial state into a goal state



Rationality

Rationality depends on...

- The performance measure for success, usually taken as **utility**
- What the agent has perceived so far
- What the agent knows about the environment
- The actions the agent can perform

An **ideal rational agent:** *for each possible percept sequence, it acts to maximize its expected utility, on the basis of its knowledge and the evidence from the percept sequence*

Rationality

Definition: An ideal rational agent performs actions that are expected to maximize its performance measure

What is rational at any given time depends on:

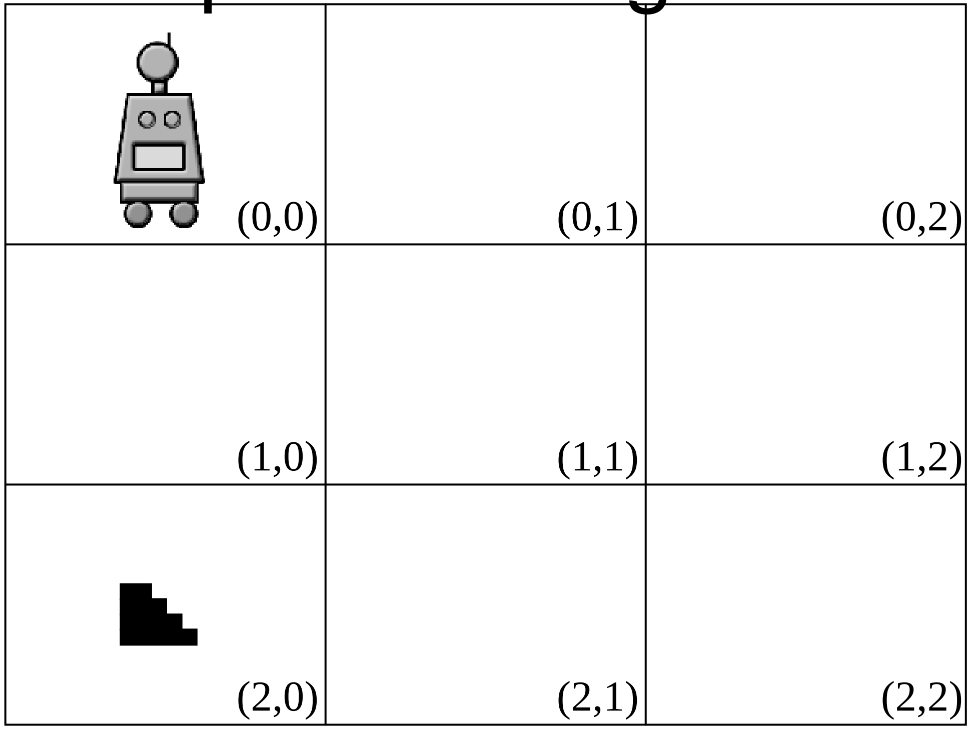
- the performance measure that determines the degree of success
- everything that the agent has perceived so far
- what the agent expects to perceive and happen in the future
- what the agent knows about the environment
- the actions that the agent can perform

Bounded rationality

- Making a decision requires computational power, memory and computation takes time
- Agents are resource-bounded and this has an impact on their decision-making process: optimal decision making may not be possible
- Ideal rationality may be difficult to achieve
- Bounded rationality
  - restrictions on the types of options may be imposed
  - the time/computation for option consideration may be limited
  - the search space may be pruned
  - the option selected will be strategically inferior to the optimal one

The Maze World

The agent’s objective is to discover the gold, pick it up and then get it to the exit (2,2)



Starting position (0,0) facing *East*

The state of the world is described by the following predicates

- *In*(*x*,*y*) the agent is in square with coordinates (*x*,*y*)
- *Gold*(*x*,*y*) there is gold in square (*x*,*y*)
- *Facing*(d) the agent faces *d* ∈{*North*, *South*, *East*, *West*}

Perception:

- The agent can perceive the world by detecting whether or not there is gold in a square, *gold* or *null* respectively
- It can also perceive its position on the grid and its direction

Possible actions *A*={*pick-up*, *forward*, *turn*}

When the agent turns, it turns 90 degrees clockwise

- The rules of inference ρ determine the agent's behaviour
- Rule for picking up the gold when detected:

$In(x,y) \wedge Gold(x,y) \rightarrow Do(pick\text{-}up)$

- Rules to enable the agent to move around:

$In(0,0) \wedge Facing(East) \wedge \neg Gold(0,0) \rightarrow Do(forward)$

$In(0,1) \wedge Facing(East) \wedge \neg Gold\ (0,1) \rightarrow Do(forward)$

$In(0,2) \wedge Facing(East) \wedge \neg Gold(0,2) \rightarrow Do(turn)$

$In(0,2) \wedge Facing(South) \wedge \neg Gold(0,0) \rightarrow Do(forward)$

…

Advantages of logic-based approach

- If there is a theory ϕ which describes the agent’s behaviour, all we have to do is execute this specification
- Elegant, intuitive, clear semantics

Other issues

- How to transform percepts into declarative statements that describe the environment precisely enough
- Writing down all the rules that would allow agents to operate in complex environments is unrealistic
- Assumes calculative rationality: the world does not change in a significant way while the agent is deliberating – not realistic
- Computational complexity of theorem proving is a problem. Propositional logic is decidable, but first-order logic is only semi-decidable: even if there is a proof, the theorem prover may fail to terminate
- Representing temporal information and changes is difficult

Agents vs. expert systems

- Expert systems which started appearing in the 1970s were heralded as the first intelligent pieces of software

Are they agents?

- Although intelligent, they have no control over their actions
- They only provide domain-specific information and usually do not adapt
- No true social ability: limited interaction with humans or other (expert) systems
- They are decoupled from their environment

ASSIGNMENT 4

(1) Contrast BDI and rational agents with regards to **rationality**. Do both involve rationality? If yes how?

(2) Does an agent have control of its actions?

Write your thoughts about points (1) and (2) – half page max for each. Due on October 2 at class.

For Sept 30 at class

- Prepare two .ppt slides outlining your intention regarding the project that you would like to accomplish for this class
- Please include: the topic of the project clearly specifying the problem that you want to address and the area of AI which you want to apply (*deductive reasoning*, *intentional* agents etc)
- Be prepared to make a 5-10 mins presentation in the beginning of the class arguing why you chose that particular area for solving the chosen problem and how do you see the possible solution.