CSC 481: First Order Logic

1- Introduction

Rodrigo Canaan
Assistant Professor
Computer Science Department
Cal Poly, San Luis Obispo
rcanaan@calpoly.edu

Motivation

Example 1

- Every day, I either drive to work or bike to work
- Today, I did not bike to work
- Therefore, today, I drove to work

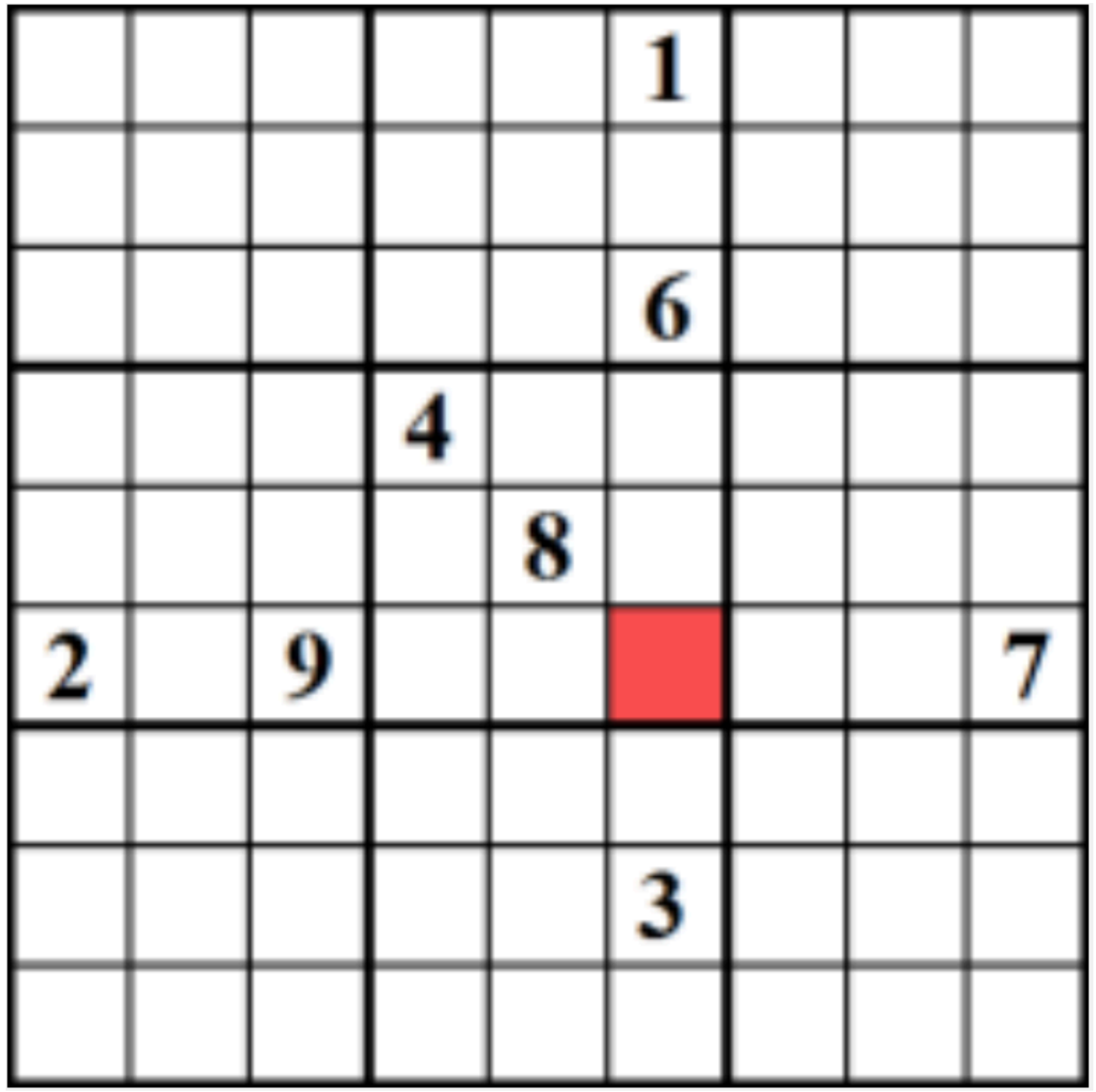
Example 2

- I live in San Luis Obispo
- San Luis Obispo is in California
- California is in the United States
- Therefore, I live in the United States

What's the difference between these two examples?

Example 3

- What number should go in the red cell?



Source: https://www.kristanix.com/sudokuepic/sudoku-solving-techniques.php (access Jan 6 2022)

Motivation

- We want to express universally true rules of logical inference such as “if X or Y is true, but Y is false, then X is true” (example 1)
- We may need to express “common sense” facts about the world such as “if X is in Y and Y is in Z, then X is in Z” and “if A lives in X, and X is in Z, then A lives in Z” (example 2)
- We may need to express domain-specific knowledge such as the rule of sudoku (example 3)

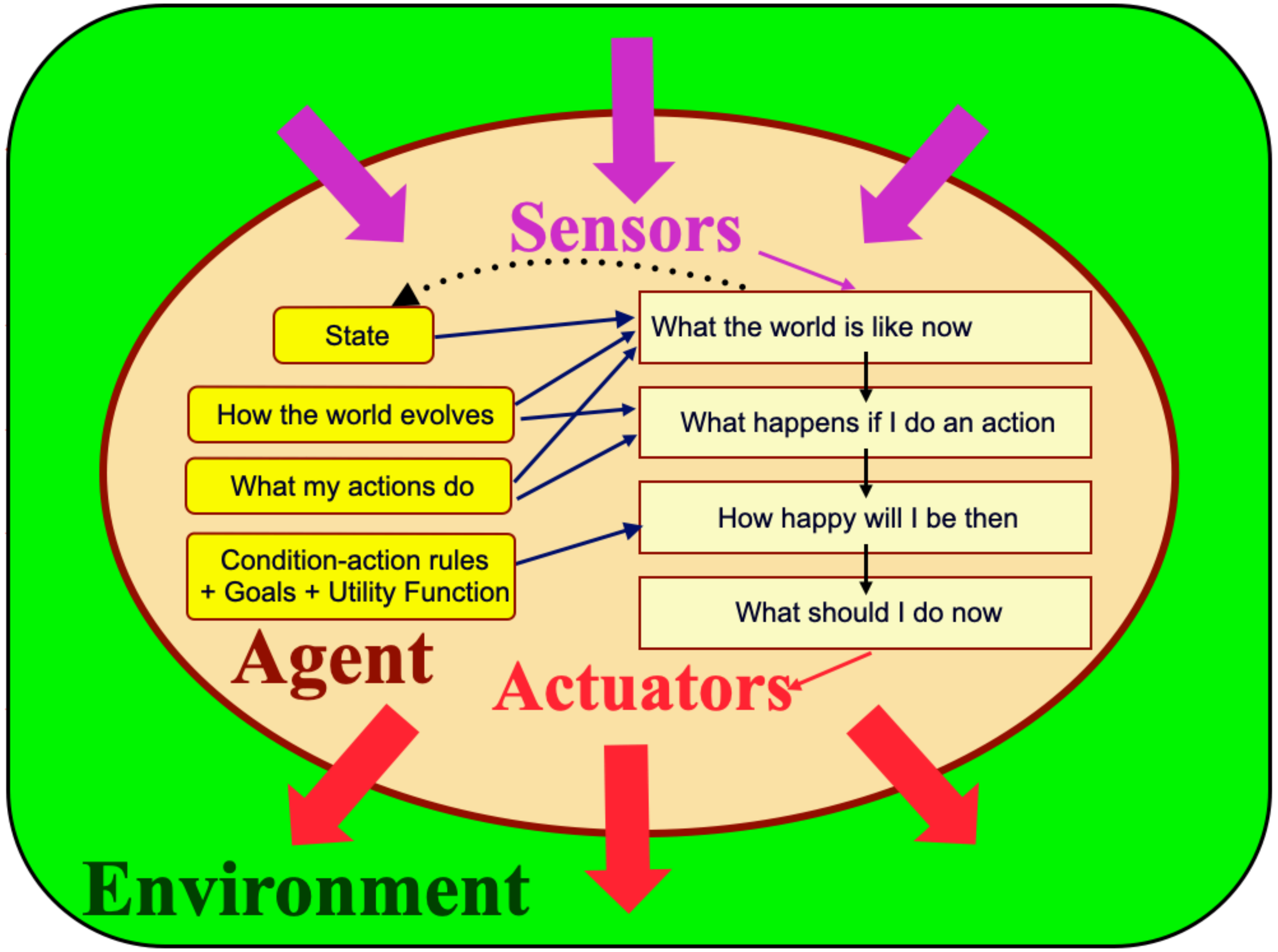
Motivation

- In general, we may want to talk about objects in the world, their relationships, and properties that hold to some (or all) of them
- We need a language
- **First Order Logic** is one (but not the only) such language

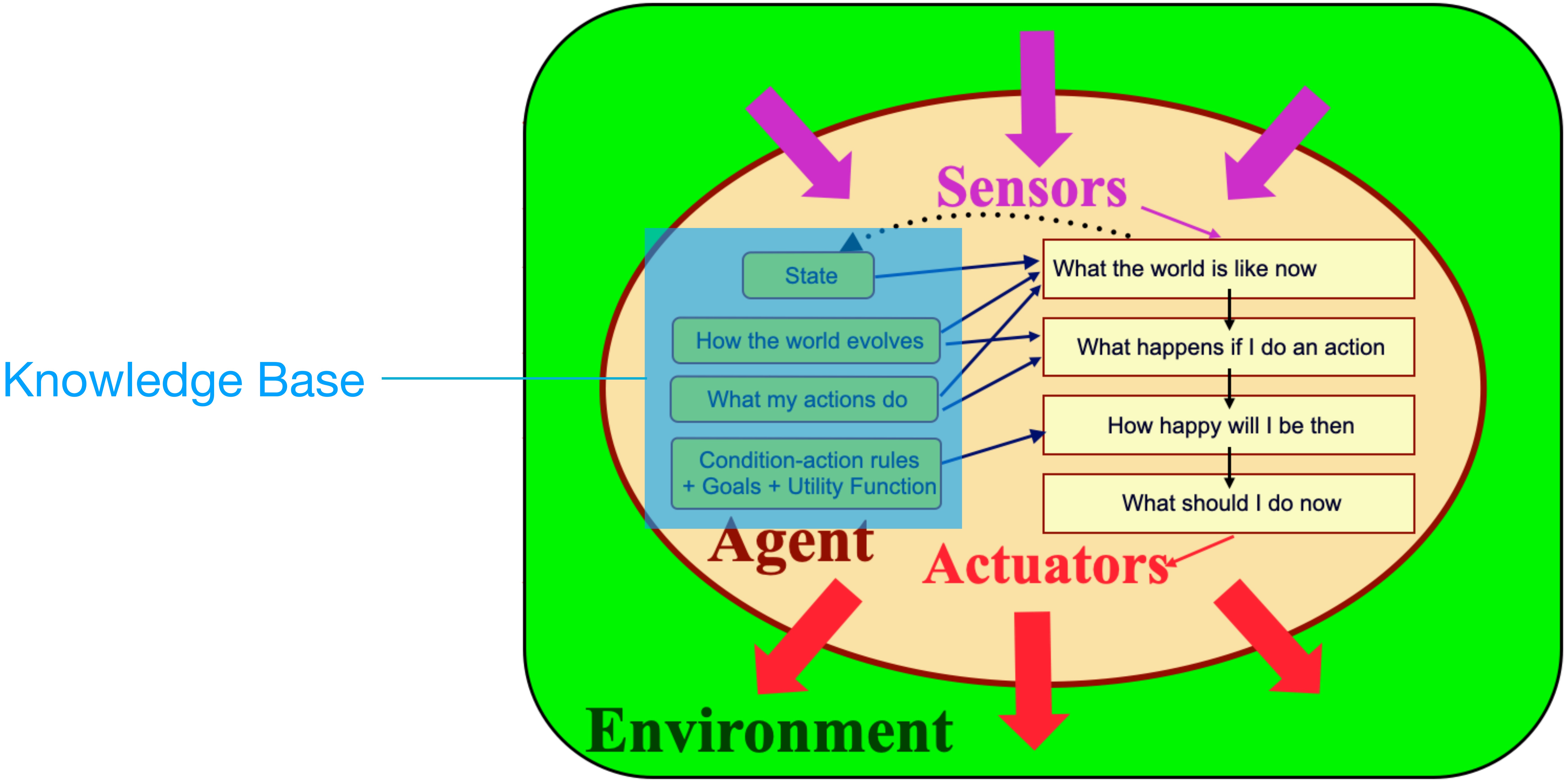
A note on agents

- Strictly speaking, logic tells us what knowledge can be inferred from a KB (that is, producing new knowledge from existing knowledge)
- However, we are often interested in *what is the right thing to do*.
- An agent is an entity (machine or human) that interacts with an environment trying to achieve a goal
- To bridge the gap between knowledge and action, we many want to check whether the KB entails propositions of the form “The correct thing to do is X”
- An agent that acts according to this principle is often called a knowledge-based agent

Example of Agent Diagram



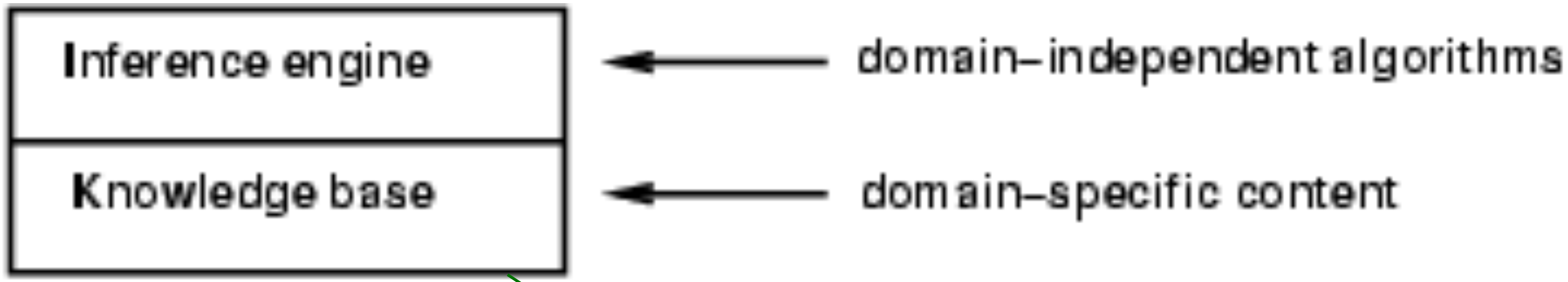
Example of Agent Diagram



Knowledge-Based agents

Inference rules such as:

- $A \vee B, \neg A \vdash B$



http://aima.eecs.berkeley.edu/slides-ppt/

"Facts" such as
- Fido is a dog
- There is a 2 in the top-left square

"Rules" such as
- Every dog is a mammal
- If two squares share a column, they must have different numbers

Knowledge-Based agents pseudocode

```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action
```

http://aima.eecs.berkeley.edu/slides-ppt/

Propositional Logic

Syntax and Semantics

First Order Logic

Note

A language is composed of:

- **Syntax:** what symbols may be used and what combinations of symbols are well-formed
  - "I drove to work today" is a well-formed English sentence
  - "I work to today drove" is not
- **Semantics:** what each sentence means
  - Sentences are used to convey that the world is one way and not another
- **Pragmatics**: what the language is used for
  - The sentence "fire!" suggests a different response in a crowded theater or in a shooting range

First Order Logic - Syntax

Logical symbols (like reserved words in a programming language)

- Punctuation: (,), .
- Connectives: $\neg, \wedge, \vee, \exists, \forall, =$
- Variables: x, y, z, x_1, x_2, x_3 etc

These have **fixed meaning** , independent on the domain (“world”)

Syntax

First Order Logic - Syntax

Non-Logical Symbols

- **Functional symbols** (returns an element in the domain)
  - Constants: a, b, c, “John”, “San Luis Obispo”…
  - Functions: bestFriend(“John”), firstChild(“John”,”Mary”)
- **Predicate symbols** (returns true or false)
  - dog(“Fido”)
  - olderThan(“Alice”, “Bob”)

Results of non-logical symbols depend on the domain (“world”) - see first slide of Semantics

First Order Logic - Arity

Arity is the number of “arguments” a function or predicate has

- Functions with zero arguments are called constants (e.g. “Bob”)
- Predicates with zero arguments are called propositions (e.g. P = “it will rain tomorrow”)
  - True (or T) and false (or F) are also propositions
- **Remember:** functions (including constants) evaluate to a member of the domain. Predicates (including propositions) evaluate to true or false

First Order Logic - Legal expressions

- terms refer to an object in the world and evaluate to an object
  - every constant or variable is a term
  - if t1, …, tn are terms, and f is a function of parity n, then f(t1,…tn) is a term
    - Special case: if f has arity zero, then the term is a constant
- formulas express propositions and evaluate to true or false
  - if t1, …, tn are terms, and P is a predicate with arity n, then P(t1,…tn) is a formula
  - if t1 and t2 are terms, then t1=t2 is a formula
  - if α and β are formulas and x is a variable, then $\neg \alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$,$\forall x. \beta$ and $\exists x. \beta$ are formulas (with x standing for one or more terms in β.
- formulas of the first two types are atomic formulas or atoms.

First Order Logic - Abbreviations

- Implication: $\alpha \implies \beta$ is the same as $\neg\alpha \vee \beta$
  - Sometimes written as $\alpha \rightarrow \beta$ or $\alpha \supset \beta$ (book)
- Biconditional or equivalence: $\alpha \equiv \beta$ is the same as $(\alpha \implies \beta) \wedge (\beta \implies \alpha)$
  - Sometimes written as $\alpha \iff \beta$

First Order Logic - Propositional Subset

- The propositional subset of First Order Logic is FOL with no terms, no quantifiers, only propositional symbols (predicates of arity zero)
- The only connectives used are $\neg$, $\wedge$, $\vee$, $\rightarrow$
  - Ex: $P \rightarrow [(\neg Q \vee R) \wedge S]$
- Also called Propositional Logic
- Simpler, but has limitations when dealing with many (or infinite) objects and complicated relationships

First Order Logic - Scope of Quantifiers

- If a variable appears within the scope of a quantifier (∃ or ∀), it is called *bound*. Otherwise it is called *free*.

$$P(x) \wedge \exists x[P(x) \vee Q(x)]$$

free bound occurrences of variables

- Formulas with no free values are called *sentences*.
  - We will mostly focus on sentences

Semantics

Semantics - Interpretation

Intuition

- Without further information, a sentence like dog("Fido") or bestPlaceToLive("California") = "San Luis Obispo" or In("Beverly Hills", "California") is neither true nor false
- We need to decide for ourselves what "Fido" means and whether it is a dog or not, which city in California is the best to live, and whether we're talking about Beverly Hills, California, or Beverly Hills, Texas
- Semantics is always relative to an interpretation, which defines:
  - What objects exist in the domain
  - Which objects have which properties
    - That is, which predicates evaluate to true or false given every possible combination of objects as arguments
  - What objects are returned by each function
- An interpretation is sometimes called a "possible world"