An Over-Simplified Introduction to Artificial Intelligence (AI) For Cognitive Science Students

Qiang Huo

Department of Computer Science & Information Systems
The University of Hong Kong, Pokfulam Road, Hong Kong

E-mail: qhuo@csis.hku.hk

OUTLINE

- What is Artificial Intelligence?
- Knowledge Representation: Semantic Networks
- Knowledge Representation: Frames
- Knowledge Representation: Predicate Logic
- Knowledge Representation: Rule-Based Systems
- Summary

CONTENTS

AI Is ......

- **Artificial Intelligence (AI)** was inaugurated as a formal discipline in 1956 at a Dartmouth College conference by *John McCarthy*, *Marvin Minsky*, *Allen Newell*, and *Herbert Simon*.
- **AI is founded on the premise** that all cognitive activity can be explained in terms of computation.
- **AI's scientific goal** is to understand the principles and mechanisms that account for intelligent action.
- **AI's engineering goal** is to design intelligent artifacts that can survive and operate in the physical world and solve problems of considerable scientific difficulty at high levels of competence.
- **AI is a broad field,** and means different things to different people.

Some Definitions of AI

- **Systems that think like humans**
  - *Example 1:* "The exciting new effort to make computers think ... *machnies with minds*, in the full and literal sense" (Haugeland, 1985)
  - *Example 2:* "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ..." (Bellman, 1978)
- **Systems that act like humans**
  - *Example 1:* "The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990)
  - *Example 2:* "The study of how to make computers do things at which, at the moment, people are better" (Rich and Knight, 1991)

Some Definitions of AI (*continued*)

- **Systems that think rationally**
  - *Example 1:* "The study of mental faculties through the use of computational models" (Charniak and McDermott, 1985)
  - *Example 2:* "The study of the computations that make it possible to perceive, reason, and act" (Winston, 1992)
- **Systems that act rationally**
  - *Example 1:* "A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" (Schalkoff, 1990)
  - *Example 2:* "The branch of computer science that is concerned with the automation of intelligent behavior" (Luger and Stubblefield, 1993)

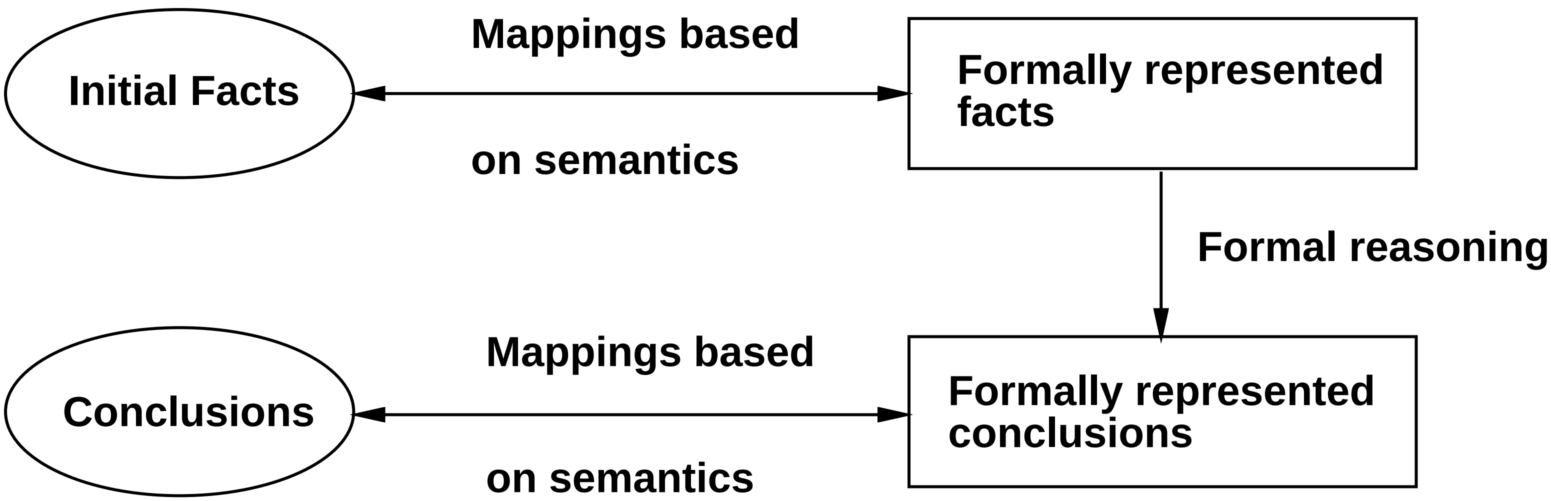
Some Typical AI Problems

- **Mundane Tasks:** tasks we, as humans, do almost automatically, yet require quite complex reasoning.
  - *Perception:* Speech, Vision
  - *Natural Language:* understanding, generation, translation.
  - *Common sense reasoning:*
  - *Robot control:*
- **Expert Tasks:** tasks require specialized skills and training.
  - Medical diagnosis, Equipment repair, Computer configuration, Financial analysis, Engineering design, Manufacturing planning, etc.
- **Formal Tasks:**
  - *Games:* Chess, Go, Checkers, etc.
  - *Mathematics:* Geometry, Logic, Integral calculus, etc.

Some AI Techniques

- **Knowledge Representation**
  - ***Solving problems*** usually requires lots of knowledge.
  - ***To reason with knowledge,*** we first need to be able to represent it in a formal manner.
  - ***Knowledge*** must be represented *efficiently*, and in a meaningful way.
    - impossible (or impractical) to represent explicitly every fact.
    - capture general abstractions which represent general features of sets of objects in the world.
    - infer new facts from existing knowledge when needed.
    - relate facts in a formal representation scheme to facts in the real world.

Some AI Techniques (*continued*)



Representations and Mappings

Some AI Techniques (*continued*)

- **Search:**
  - Often there is no direct way to find a solution to some problem.
  - However, you do know how to generate possibilities.
  - Developing good ways to search through these possibilities for a good solution is therefore vital.
    - *Brute force search:* generate and try out every possible solution.
    - *Heuristic search:* only try the options which, based on current best guess, are most likely to lead to a good solution.
- **Reasoning and Decision Making under Uncertainty**
- **Machine Learning**

General Requirements For Knowledge Representation

Knowledge Representation Languages have been developed to make it easy to represent and reason with complex knowledge about the world.

- **Representation Adequacy:** It should allow you to represent all the knowledge that you need to reason with.
- **Inferential Adequacy:** It should allow new knowledge to be inferred from a basic set of facts.
- **Inferential Efficiency:** Inferences should be made efficiently.
- **Clear Syntax and Semantics:** We should know what the allowable expressions of the language are and what they mean.
- **Naturalness:** The language should be reasonably natural and easy to use.

Main Approaches for Knowledge Representation

No one representation language satisfies all the above-mentioned requirements perfectly. In practice, the choice of language depends on the reasoning task.

- **Semantic Networks**
- **Frames**
- **The use of Logic**
- **IF-THEN or Condition-Action rules**

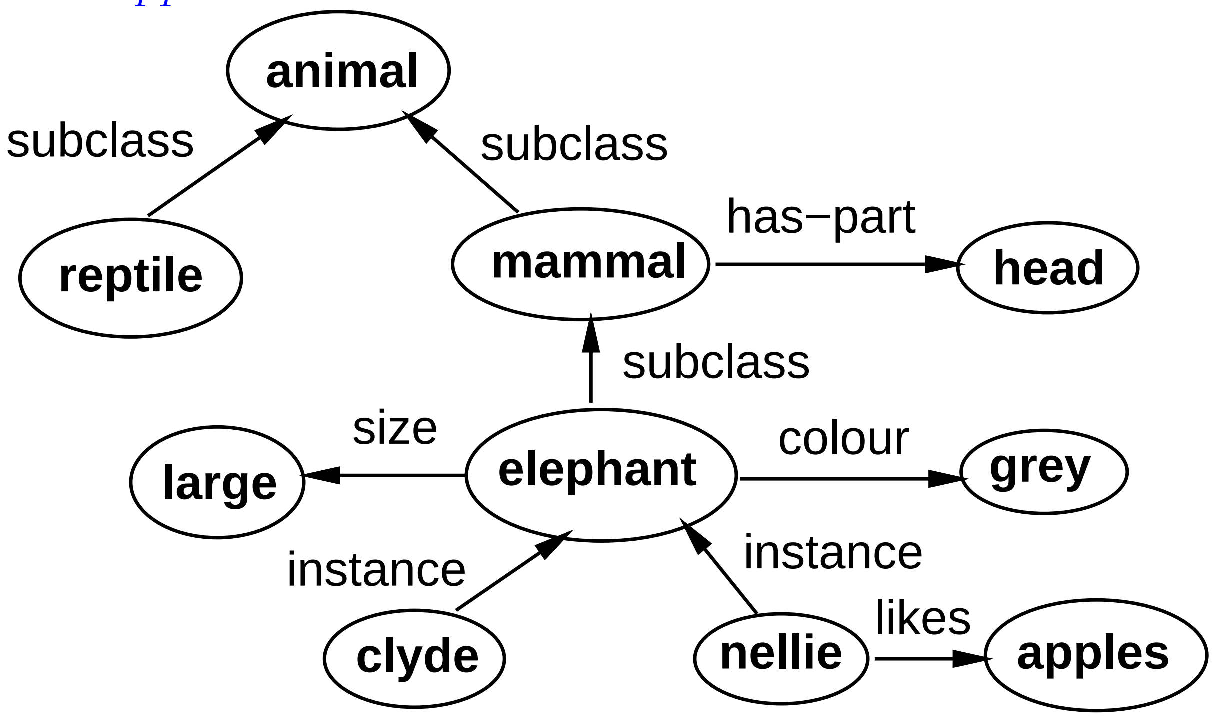
CONTENTS

What Are Semantic Networks?

- **Semantic Networks** were originally developed in the early 1960s to represent the meaning of English words. They have since been used more widely for representing knowledge.
- **In a Semantic Network**, knowledge is represented as a graph, where
  - ***The nodes*** in the graph represent *concepts*, and
  - ***The links (arcs)*** represent *relations* between concepts.
    - *subclass* relations between classes,
    - *instance* relations between particular object instances and their parent class,
    - *any other relations (functions)* to represent properties of objects (and categories of objects)

What Are Semantic Networks? (*continued*)

- **Example:** how to represent the following facts with a semantic network? *Mammals and reptiles are animals, that mammals have heads, an elephant is a large grey mammal, Clyde and Nellie are both elephants, and that Nellie likes apples.*



Inference Mechanism

- The subclass relations define a *class hierarchy*.
- The subclass and instance relations may be used to derive new information which is not explicitly represented.
  - ***Example:*** we should be able to conclude that Clyde and Nellie both have a head, and are large and grey. They *inherit* properties from their parent classes.
  - ***Semantic networks*** normally allow efficient *inheritance-based inferences* using special purpose algorithms.
- **Semantic networks** allow us to represent knowledge about objects and relations between objects in a simple and fairly intuitive way. While the notation may be ill suited where very complex knowledge representation and reasoning is required, it may be a good choice for certain problems.

CONTENTS

What Are Frames?

- **Frames** are a variant of semantic networks. A *frame* has a *name* and a set of *attribute-value* pairs.
  - ***The frame name*** corresponds to a node in a semantic network.
  - ***The attributes*** correspond to the names of arcs (links) associated with this node, and the *values* correspond to nodes at the other ends of these arcs.
    - The attribute-value pairs are usually called *slots*.
    - The attributes are called *slot names*, and the values are called *slot fillers*.

What Are Frames? (*continued*)

- **Example:** three simple frames representing knowledge about elephants.

```
Mammal:
  subclass:     Animal
  has_part:     head

Elephant:
  subclass:     Mammal
  colour:       grey
  size:         large

Nellie:
  instance:     Elephant
  likes:        apples
```

- We can infer, using *inheritance*, that *Nellie* is large, grey and has a head, as well as liking apples.

Defaults and Multiple Inheritance

- In the examples so far, objects (such as Nellie) inherit *all* the properties from their parent class. However,
- It is useful to be able to describe properties that are only *typical* of a class, and then state that a particular instance of that class is an exception to the rule.
  - The value of a property that is only typical of a class is referred to as a *default* value, and can be *overridden* by giving a different value for an instance or subclass.
  - Objects and classes *inherit the default (typical) properties* of their parent classes *unless* they have an individual property value that conflicts with the inherited one.
- Inheritance is simple where each object and class has a single parent class.

Defaults and Multiple Inheritance (*continued*)

- **Example:** elephant frames with defaults.

```
Mammal:
  subclass:     Animal
  warm_blooded: yes
  * furry:      yes

Elephant:
  subclass:     Mammal
  has_trunk:    yes
  * colour:     grey
  * size:       large
  * furry:      no

Clyde:
  instance:     Elephant
  colour:       pink
  owner:        Fred

Nellie:
  instance:     Elephant
  size:         small
```

```
Given this set of frames, we can infer that
- Clyde is warm-blooded, unfurry, has a
  trunk, is pink, large and owned by Fred.
- Nellie is warm-blooded, unfurry, has a
  trunk, grey and small.
```

Defaults and Multiple Inheritance (*continued*)

- **Multiple inheritance** means that more than one parent class is allowed, and an object or class may inherit from all its parents.

```
Elephant:
  subclass:      Mammal
  has_trunk:     yes
  * colour:      grey
  * size:        large
  * habitat:     jungle

Circus_Animal:
  subclass:      Animal
  habitat:       tent
  skills:        balancing-on-ball

Clyde:
  instance:      Circus_Animal Elephant
  colour:        pink
  owner:         Fred
```

- **What is Clyde's habitat?** A frame system must have some mechanism to decide which value to inherit where there are conflicts like this!

Representational Adequacy

- **Semantic networks and frames** provide a fairly simple and clear way of representing properties of objects and categories of objects.
- A basic type of inference is defined, whereby objects may *inherit* properties of parent objects.
- There are many things that *cannot* easily be represented using frames.
  - ***Negation:*** the fact that something is NOT true.
  - ***Disjunction:*** the fact that either one thing OR another is true.
  - ***Quantifications:*** the fact that something is true for ALL or SOME of a set of objects.

If these things are needed, then using a *Logic* may be more appropriate.

CONTENTS

Basic Concepts

- **A logic** is a formal system which may be described in terms of its
  - *Syntax:* what the allowable expressions are;
  - *Semantics:* what they mean; and its
  - *Proof theory:* how can we draw new conclusions given some statements in the logic.
- **The First-Order Predict Logic** is a well-understood formal language, with well-defined syntax, semantics and rules of inference. Consequently, Predict Logic allows us
  - *to represent* fairly complex facts about the world, and
  - *to derive* new facts in a way that guarantees that, if the initial facts were true, then so are the conclusions.

Review of Propositional Logic

- **Syntax**
  - In propositional logic, symbols are used to represent facts about the world. For example,
    - the fact "Alison likes cakes" could be represented by the symbol P (or indeed any other symbol, such as the more meaningful $AlisonLikesCakes$).
    - Simple facts like this are referred to as *atomic propositions*.

Review of Propositional Logic (*continued*)

- We can build up more complex statements (or sentences) by combining atomic propositions with the *logical connectives* $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\rightarrow$ (implication) and $\leftrightarrow$ (equivalence). So, if we had the proposition Q representing the fact "Alison eats cakes", we could have the facts:
  - $P \vee Q$: "Alison likes cakes or Alison eats cakes"
  - $P \wedge Q$: "Alison likes cakes and Alison eats cakes"
  - $\neg Q$: "Alison doesn't eat cakes"
  - $P \rightarrow Q$ "If Alison likes cakes then Alison eats cakes"
  - $P \leftrightarrow Q$: "If Alison likes cakes then Alison eats cakes, and vice versa".

Review of Propositional Logic (*continued*)

- In general, if X and Y are sentences in propositional logic, then so are $X \wedge Y$, $X \vee Y$, $\neg X$, $X \rightarrow Y$, and $X \leftrightarrow Y$. This defines the *syntax* of the logic. The following are all valid sentences in propositional logic:
  $P \vee \neg Q$
  $P \wedge (P \rightarrow Q)$
  $(Q \vee \neg R) \rightarrow P$

Review of Propositional Logic (*continued*)

Semantics

- ***The semantics of propositional logic*** is defined in terms of what is true in the world. For example, if we know whether P, Q, and R are true, the semantics of the logic will tell you whether sentences such as $(P \vee Q) \wedge R$ are true.
- ***We can determine the truth or falsity (or truth value)*** of sentences using *truth tables* which define the truth values of sentences with logical connectives in terms of the truth values of their component sentences.
- ***The truth tables*** provide a simple *semantics* for these logical connectives, i.e., define precisely what the logical connectives mean.

p	$\neg p$
T	F
F	T

Review of Propositional Logic (*continued*)

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Review of Propositional Logic (*continued*)

Proof Theory

- In order to infer new facts in a logic, we need to apply *inference rules*. Only those propositions which are *universally true* can be used as rules of inference.
- One useful inference rule is the so-called *modus ponents*, i.e.,

$$(A \wedge (A \rightarrow B)) \rightarrow B$$

or represented graphically as follows:

$$\frac{A,\ A \rightarrow B}{B}$$

– We could prove that $(A \wedge (A \rightarrow B)) \rightarrow B$ is always true by using truth tables.

– This rule just says that if $A \rightarrow B$ is true, and A is true, then B is necessarily true.

Review of Propositional Logic (*continued*)

- There are many other sound rules of inference. A particularly important one is *resolution*:

$$((A \vee B) \wedge (\neg B \vee C)) \rightarrow (A \vee C)$$

or represented graphically as follows:

$$\frac{A \vee B,\ \neg B \vee C}{A \vee C}$$

Review of Predicate Logic

The trouble with propositional logic is that it is not possible to write general statements in it, such as "Alison eats everything that she likes". Predicate logic makes such general statements possible.

Syntax

- Sentences in predicate logic are built up from *atomic sentences*.
- Rather than dealing with undivisible propositions, predicate logic expresses basic facts in terms of a *predicate name* and some *arguments*. So, for "Alison likes chocolate" we could have
  - a predicate name $likes$
  - arguments $alison$ and $chocolate$

  to give the sentence $likes(alison, chocolate)$.
- In general the arguments in an atomic sentence may be any *term*.

Review of Predicate Logic (*continued*)

- ***Terms*** may be
  - *constant symbols* such as $alison$;
  - *variable symbols* such as X; and
  - *function expressions* such as $father(alison)$.
    - Function expressions consist of a functor followed by a number of arguments, which can be arbitrary terms.
- ***Sentences*** in predicate logic *are constructed* (much as in propositional logic) *by combining atomic sentences with logical connectives*, so the following are all sentences in predicate calculus:
  $friend(alison, richard) \rightarrow likes(alison, richard)$
  $likes(alison, richard) \vee likes(alison, chocolate)$

Review of Predicate Logic (*continued*)

- Sentences can also be formed using *quantifiers* to indicate how any variables in the sentence are to be treated. The two quantifiers in predicate logic are $\forall$ and $\exists$.
  - $\forall$ is read as "for all", and is used to state that something is true for every object, and
  - $\exists$ is read as "there exists", and is used to state that something is true for at least one object.

  Following are some examples:
  - $\forall X(likes(alison, X) \rightarrow eats(alison, X))$, i.e., Alison eats everything that she likes.
  - $\exists X(bird(X) \wedge \neg flies(X))$, i.e., there exists some birds that doesn't fly.
  - $\forall X(person(X) \rightarrow \exists Y loves(X, Y))$, i.e., every person has something that they love.

Review of Predicate Logic (*continued*)

- ***A sentence should have all its variables quantified.*** So, strictly,
  - an expression like $\forall X loves(X, Y)$, although *well-formed formula* of predicate logic, is not a sentence, as the variable Y isn't quantified.
  - Formulae with all their variables quantified are also called *closed formulae*.

Review of Predicate Logic (*continued*)

- **Semantics:**
  The semantics of predicate logic is defined (as in propositional logic) in terms of the truth values of sentences.

- **Proof Theory:**
  Inference rules (and proof procedures) in predicate logic are similar to those for propositional logic. *Modus Ponens* and *Resolution* still apply, but have to be modified to deal with expressions which involve variables and quantifiers. For example

$$\frac{\forall X(man(X) \to mortal(X)),\ man(socrates)}{mortal(socrates)}$$

$man(X)$ can be matched with $man(socrates)$ with $X = socrates$.

Knowledge Representation using Predicate Logic

- **Representing Facts in Logic**
  - *Statements like* "Alison likes chocolate" can be expressed like this:
    - use the verb as the predicate name and the noun as the argument, to get $likes(alison, chocolate)$.
  - *A statement describing a property of an individual,* such as "Mary is tall", can be expressed by using a one-argument predicate, such as $tall(mary)$.

Knowledge Representation using Predicate Logic (*continued*)

- ***Logical connective*** $\wedge$ must be used to express a statement like "Mary is both tall and beautiful" as $tall(mary) \wedge beautiful(mary)$.
  - Logical connectives *cannot* be used within an argument of a predicate, so $likes(alison, chocolate \wedge cream)$ is not correct.
  - A *correct* way to express the statement "Alison likes both chocolate and cream" is $likes(alison, chocolate) \wedge likes(alison, cream)$.
- ***Statements of the form "If X then Y"*** can be translated into $X \rightarrow Y$. So,
  - "If Alison is hungry then she eats chocolate" could be: $hungry(alison) \rightarrow eats(alison, chocolate)$.

Knowledge Representation using Predicate Logic (*continued*)

- ***For statements involving "or"*** then the connective $\vee$ may be used:
  - $eats(alison, chocolate) \vee eats(alison, biscuits)$.
- ***To assert that something is not true,*** $\neg$ is used:
  - $\neg likes(alison, brusselsprouts)$.
- ***For more general rules the quantifier*** $\forall$ ***may be used.*** A common form is $\forall X p(X) \rightarrow q(X)$, where p and q are any predicates.
  For example
  - "All students study" could be expressed as
    $\forall X student(X) \rightarrow study(X)$
    (that is, for all things, if that thing is a student then that thing will study).

Knowledge Representation using Predicate Logic (*continued*)

- ***The quantifier*** $\exists$ is used when something only has to be true of one object. For example,
  - "Someone strange likes brussel sprouts" could be expressed as
    $\exists X strange(X) \wedge likes(X, brusselsprouts)$
    (i.e., there exists some thing that is both strange and likes brussel sprouts).

Knowledge Representation using Predicate Logic (*continued*)

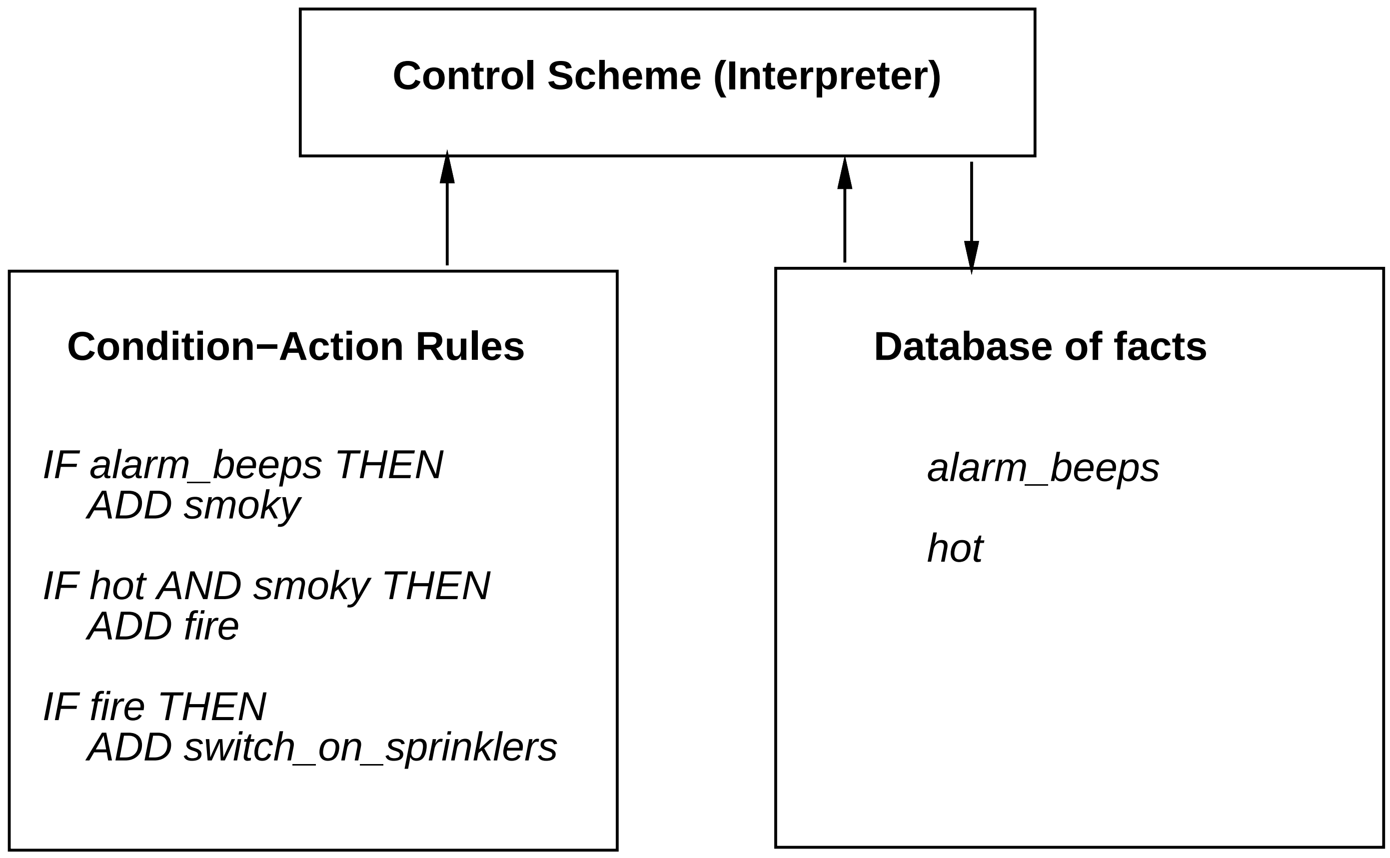
- **Predicate logic** provides a powerful way to represent and reason with knowledge.
  - Some things that cannot be easily represented using frames, such as *negation*, *disjunction* and *quantification*, are easily represented using predicate logic.
  - The available *inference rules* and *proof procedures* mean that a much wider range of inferences are possible than the simple *inheritance-based inference* allowed in a frame system.
- **There are some things that are hard to represent** using *predicate logic*, particularly facts that involve
  - ***Uncertainty*** e.g., "It will probably rain tomorrow",
  - ***Defaults*** e.g., "It normally rains in Glasgow",
  - ***Beliefs*** e.g., "John believes it will rain, but I don't", and
  - ***Time/Change*** e.g., "It will get wetter as you near Glasgow".

CONTENTS

Rule-Based Systems

- Instead of representing knowledge in a relatively declarative, static way (as a set of things that are true), rule-based systems represent knowledge in terms of *a set of rules that tell you what you should do or what you can conclude in different situations.*
- **A rule-based system** consists of
  - ***a set of IF-THEN rules,***
  - ***a set of facts,*** normally representing things that are currently held to be true, and some
  - ***interpreter*** controlling the application of the rules, given the facts.
- A rule-based system treats each rule as an independent chunk of knowledge, to be invoked when needed under the control of the interpreter.

Rule-Based Systems (*continued*)



Rule-Based System Architecture

Rule-Based Systems (*continued*)

- **Two main kinds of interpreter:**
  - *In a forward chaining system* you start with some initial facts, and keep using the rules to draw new conclusions (or take certain actions) given those facts.
    - Data-driven
  - *In a backward chaining system* you start with some hypothesis (or goal) you are trying to prove, and keep looking for rules that would allow you to conclude that hypothesis, perhaps setting new subgoals to prove as you go.
    - Goal-driven

CONTENTS

Summary

- AI is concerned with attempts to produce programs to do tasks which require human intelligence.
- Reasons for doing AI include both the goal of *understanding human intelligence better* and the goal of *developing useful, smarter computer programs*.
- AI tasks involve both *mundane tasks* which people can do very easily (e.g., understanding language) and *expert tasks* which require specialist knowledge (e.g., medical diagnosis).
- AI has been successful in limited tasks, but it is unclear whether a really human-like intelligent robot is *possible* or *desirable*.
- **Knowledge representation languages** provide high-level representation formalisms to represent the knowledge required for AI problem solving.

Summary (*continued*)

- A good language should be natural, clear and precise, allow you to *represent what is required*, and support the sound *inference of new facts*.
- **Frames and semantic networks** represent knowledge as an organized collection of objects with attributes, arranged in a *hierarchy*. If an object is a *subclass* of another it may *inherit* its attributes. They are *limited* in what can be represented and inferred, but provide a natural and efficient representation scheme.
- **A logic, and in particular predicate logic,** may be used as a *precise and formal language* able to represent a fairly wide range of things. A logic may also be used to describe the semantics of other formalisms.
- **Rule-based systems** allow knowledge to be represented as a set of more-or-less independent *IF-THEN* or *condition-action* rules, stating what action to take given different conditions. Reasoning can be controlled using a *forward or backward chaining interpreter.*

The Challenge of AI

- **The main challenge of AI** is to create *models* and *mechanisms* of intelligent action. An intelligent system is characterized as one that can
  - exhibit adaptive goal-oriented behavior,
  - learn from experience,
  - use vast amounts of knowledge,
  - exhibit self-awareness,
  - interact with humans using language and speech,
  - tolerate error and ambiguity in communication, and
  - respond in real time.

The Challenge of AI (*continued*)

- **Creating mechanisms** for sharing of knowledge, know-how, and literacy is the challenge.

  If you give a fish to a man, you will feed him for a day. If you give him a fishing rod, you will feed him for life. **Kong Zi**

  If we can provide him with the knowledge and the know-how for making that fishing rod, we can feed the whole village! **Raj Reddy**

References

- Alison Cawsey, *The Essence of Artificial Intelligence*, Prentice Hall, 1998.
- Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.
- Elaine Rich and Kevin Knight, *Artificial Intelligence*, Second Edition, McGraw-Hill, 1991.
- Nils J. Nilsson, *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann Publishers, 1998.
- Raj Reddy, "The challenge of artificial intelligence," *IEEE Computer Magazine*, October 1996, pp.86-98.