Artificial Intelligence

Unit 1: Introduction to AI, Agents and Logic

Syedur Rahman

syedur.rahman@wolfson.oxon.org

Artificial Intelligence: Lecture Notes

The lecture notes from the introductory lecture and this unit will be available shortly from the following URL:

- http://www.geocities.com/syedatnsu/

Acknowledgements

These lecture notes contain material from the following sources

- *Logical Programming and Artificial Intelligence* by S. Kapetanakis, 2004
- *Artificial Intelligence: A modern approach* by S. Russell and P. Norvig, International Edition, 2nd edition
- *Intelligent Systems* by S.Clark, 2005

Unit 1: Introduction to AI, Agents and Logic

- History and Introduction to Artificial Intelligence
- Definition of Rational Agents
- Environments, PEAS description and types
- Types of Rational Agents
- Logic and Logical Arguments
- Models and Countermodels
- Syntax/Semantics of Propositional Logic
  Sentences, models and truth tables
- Logical Equivalence and Entailment

What is Artificial Intelligence?

Definition:

- Systems that think like humans?
- Systems that think rationally?
- Systems that act like humans?
- Systems that act rationally?

The definition of AI that we will adopt in this course is that of an agent performing rational action.

A Brief History of AI (1)

The gestation of AI: Warren McCulloch and Walter Pitts (1943): a model of artificial neurons able to perform computations

1950: Alan Turing's "Computing Machinery and Intelligence". First Complete Vision of AI

The birth of AI (1956): Dartmouth Workshop bringing together top minds on automata theory, neural nets and the study of intelligence.

Collapse in AI research (1966 - 1973): unrealistic predictions, lacks of scalability, limitations on techniques/representation

A Brief History of AI (2)

AI revival through knowledge-based systems (1969-1979): DENDRAL, MYCIN (diagnoses blood infections)

AI becomes an industry (1980-): R1 at DEC, 5th generation in Japan, return of neural networks

AI becomes a science (1987-): speech recognition, neural networks, bayesian networks

Emergence of intelligent agents (1995-): agents act/behave embedded in real environments with continuous sensory inputs

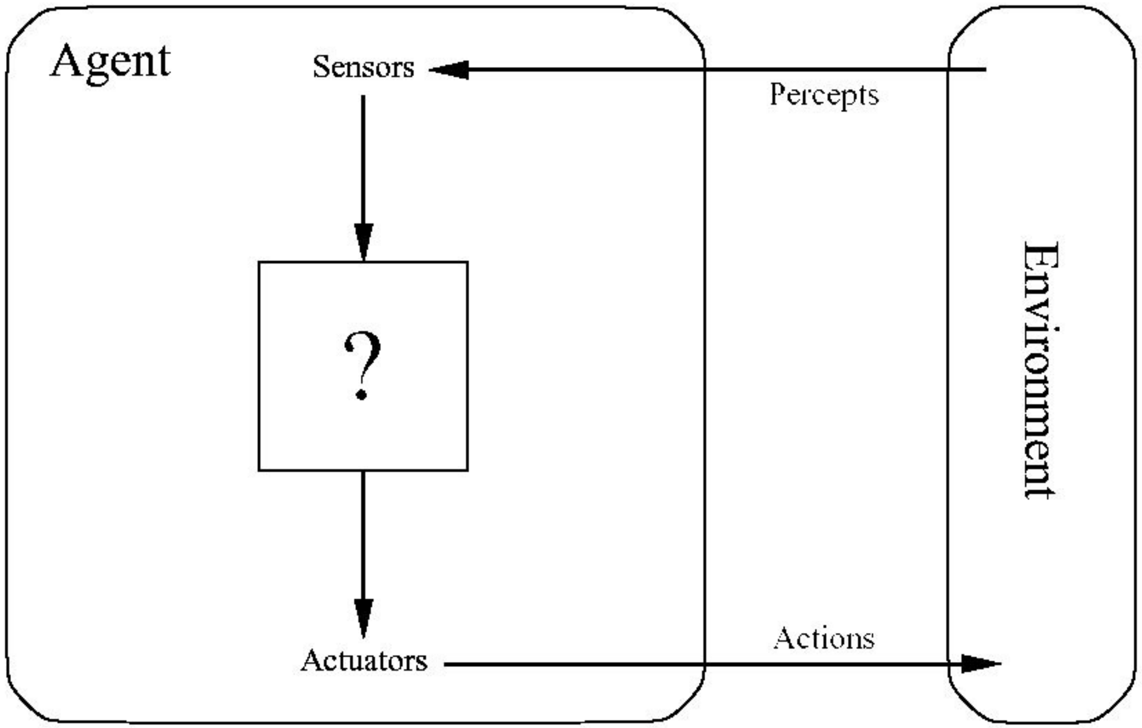
Back to the definition of AI

If one defines Artificial Intelligence as an agent performing rational action, some new questions come to mind.

- What is an agent?
- Rational behaviour: doing the right thing, but what is the right thing?
- Does doing the right thing involve thinking?
- Is an agent always able to do the right thing? (Limited rationality)

What is an agent?

There is no general consensus on a definition for an agent.

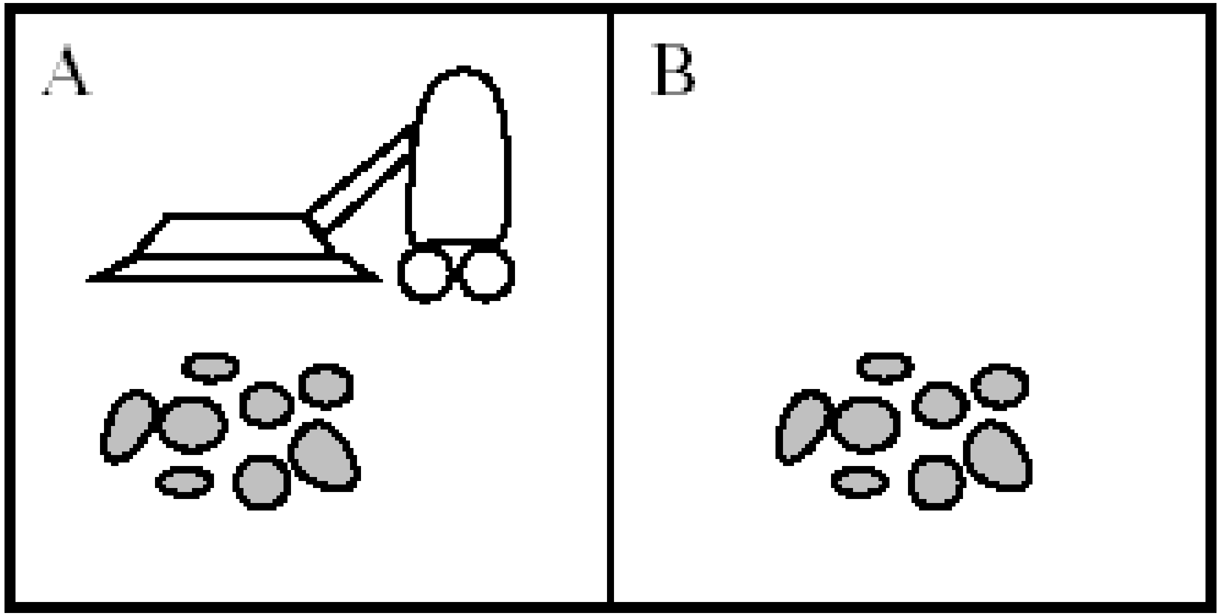


Example of an agent

Environment: squares *A* and *B*

Percepts: [*location, status*] e.g. [*A, Dirty*]

Actions: *left*, *right*, *suck*, and *no-op*



Definition of a Rational Agent

For each possible percept sequence, a *rational agent* should select an action (using an *agent function*) that is expected to maximise its performance measure, given the evidence provided by the *percept sequence* and whatever built-in *prior knowledge* the agent has.

A *percept sequence* is the complete history of anything the agent has ever perceived.

A *performance measure* is a means of calculating how well the agent has performed based on the sequence of percepts that it has received.

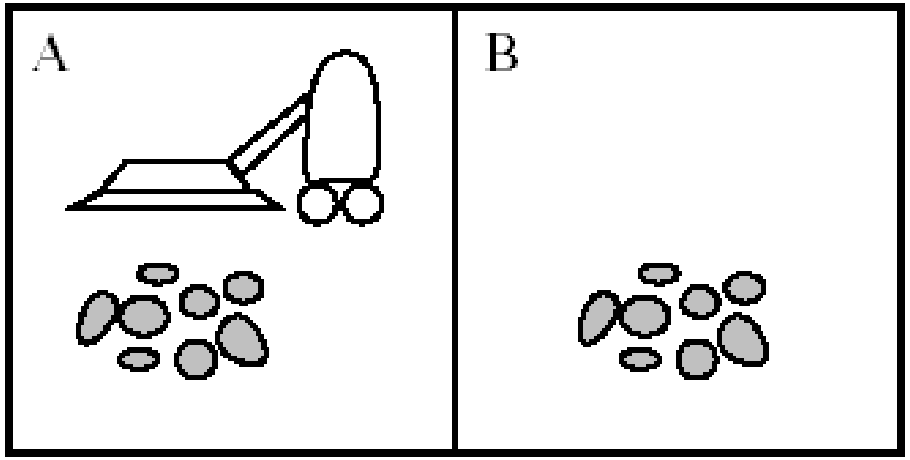
An agent's *prior knowledge* of the environment is the knowledge that the agent designer has bestowed upon the agent before its introduction to the environment.

Definition of a Rational Agent (cntd)

An *agent function* maps percept sequences to actions.

$$f : \text{seq}(P) \rightarrow A$$

Agent Function for Vacuum Cleaner Example:



Percept sequence	Action
[A,Clean]	*Right*
[A, Dirty]	*Suck*
[B, Clean]	*Left*
[B, Dirty]	*Suck*
[A, Clean],[B, Clean]	*Left*
[A, Clean],[B, Dirty]	*Suck*
…	…

Environments

To design a rational agent we must specify its task environment

PEAS description of the environment

- **Performance**
- **Environment**
- **Actuators**
- **Sensors**

Environments

To design a rational agent we must specify its task environment

PEAS description of the environment (e.g. for a automated taxi):

- **Performance**
  - E.g. Safety, destination, profits, legality, comfort
- **Environment**
  - E.g. Streets/motorways, other traffic, pedestrians, weather
- **Actuators**
  - E.g. Steering, accelerating, brake, horn, speaker/display
- **Sensors**
  - E.g. Video, sonar, speedometer, engine sensors, keyboard, GPS

Environment types

The environment for an agent may be

- Fully or partially observable
- Deterministic or stochastic
- Episodic or sequential
- Static or dynamic
- Discrete or continuous
- Single or multi-agent

Environment types

Fully vs. partially observable: an environment is fully observable when the sensors can detect all aspects that are relevant to the choice of action

	VcmClnr	Chess	Internet shopping	Taxi
Observable				
Deterministic				
Episodic				
Static				
Discrete				
Single-agent				

Environment types

Deterministic vs. stochastic: if the next environment state is completely determined by the current state and the executed action then the environment is deterministic

	VcmClnr	Chess	Internet shopping	Taxi
Observable	**PARTIAL**	**FULL**	**PARTIAL**	**PARTIAL**
Deterministic				
Episodic				
Static				
Discrete				
Single-agent				

Environment types

Episodic vs. sequential: In an episodic environment the agent's experience can be divided into atomic episodes where the agent perceives and then performs a single action; the choice of action does not depend on previous actions

	VcmClnr	Chess	Internet shopping	Taxi
Observable	**PARTIAL**	**FULL**	**PARTIAL**	**PARTIAL**
Deterministic	**NO**	**YES**	**YES**	**NO**
Episodic				
Static				
Discrete				
Single-agent				

Environment types

Static vs. dynamic: If the environment can change while the agent is choosing an action, the environment is dynamic. If the agent's performance score changes even when the environment remains the same, the environment is **semi-dynamic**.

	VcmClnr	Chess	Internet shopping	Taxi
Observable	**PARTIAL**	**FULL**	**PARTIAL**	**PARTIAL**
Deterministic	**NO**	**YES**	**YES**	**NO**
Episodic	**YES**	**NO**	**NO**	**NO**
Static				
Discrete				
Single-agent				

Environment types

Discrete vs. continuous: This distinction can be applied to the state of the environment, the way time is handled and to the percepts/actions of the agent.

	VcmClnr	Chess	Internet shopping	Taxi
Observable	**PARTIAL**	**FULL**	**PARTIAL**	**PARTIAL**
Deterministic	**NO**	**YES**	**YES**	**NO**
Episodic	**YES**	**NO**	**NO**	**NO**
Static	**NO**	**YES**	**YES**	**NO**
Discrete				
Single-agent				

Environment types

Single vs. multi-agent: Does the environment contain other agents who are also maximizing some performance measure that depends on the current agent's actions?

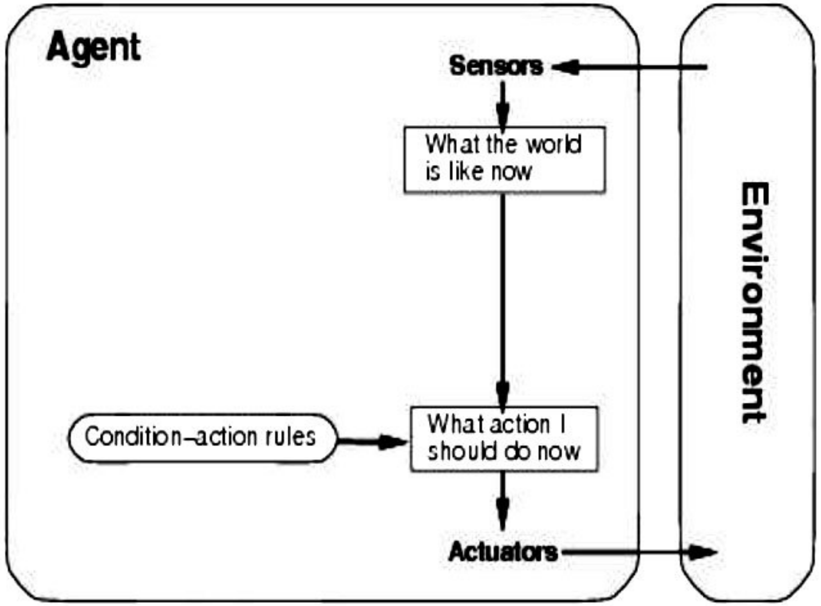
	VcmClnr	Chess	Internet shopping	Taxi
Observable	PARTIAL	FULL	PARTIAL	PARTIAL
Deterministic	NO	YES	YES	NO
Episodic	YES	NO	NO	NO
Static	NO	YES	YES	NO
Discrete	YES	YES	YES	NO
Single-agent				

Environment types

Single vs. multi-agent: Does the environment contain other agents who are also maximizing some performance measure that depends on the current agent's actions?

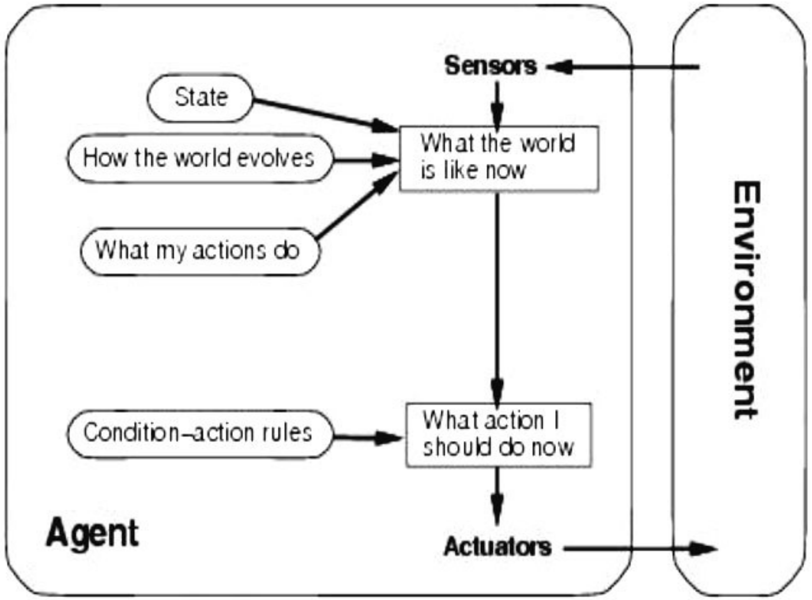
	VcmClnr	Chess	Internet shopping	Taxi
Observable	PARTIAL	FULL	PARTIAL	PARTIAL
Deterministic	NO	YES	YES	NO
Episodic	YES	NO	NO	NO
Static	NO	YES	YES	NO
Discrete	YES	YES	YES	NO
Single-agent	YES	NO	NO	NO

Agent Types: Simple Reflex



- Select action on the basis of *only the current* percept
  - E.g. the vacuum-agent
- Large reduction in possible percept/action situations
- Implemented through *condition-action rules*
  - If dirty then suck

Agent Types: Model-based



To tackle *partially observable* environments

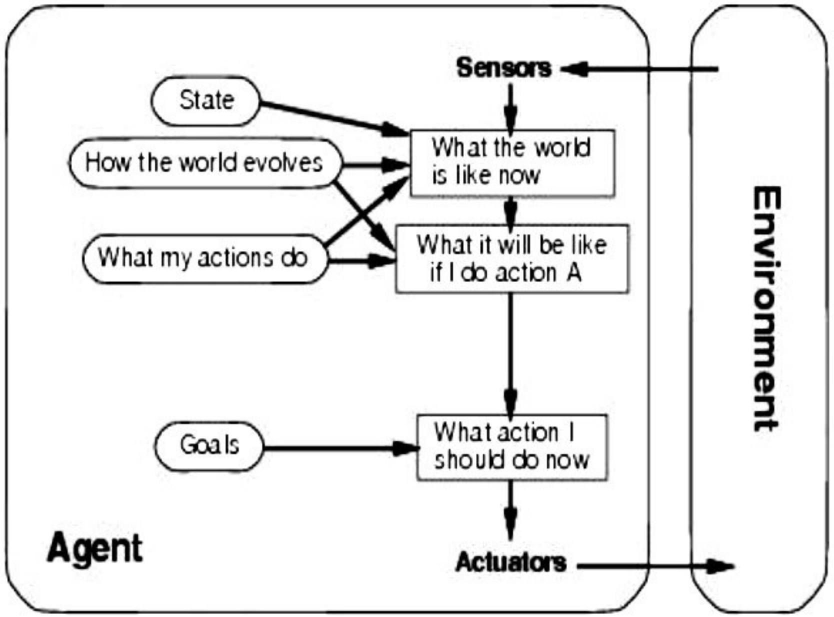
- Maintain internal state that depends on percept history

Over time update state using world knowledge

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

$\Rightarrow$ *Model of World*

Agent types: Goal-based



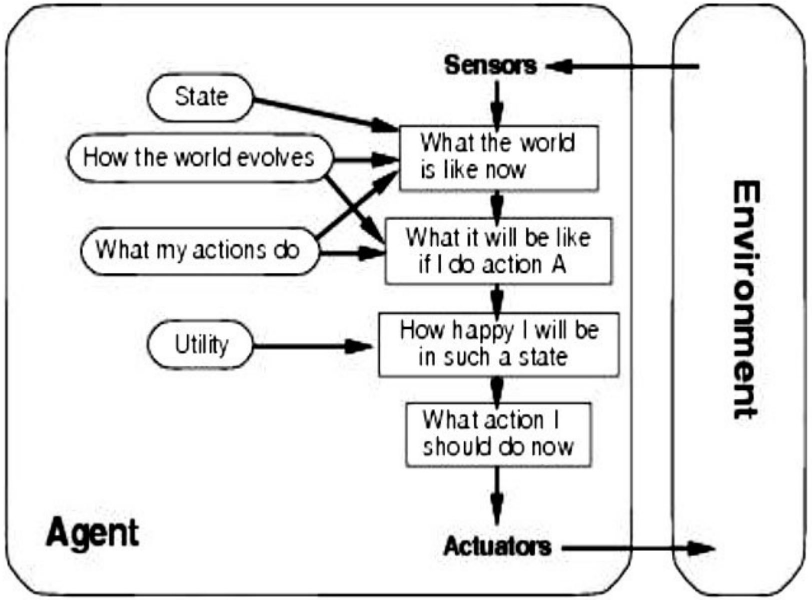
The agent needs a goal to know which situations are *desirable*

- Difficulties arise 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

Agent types: Utility-based



Certain goals can be reached in different ways

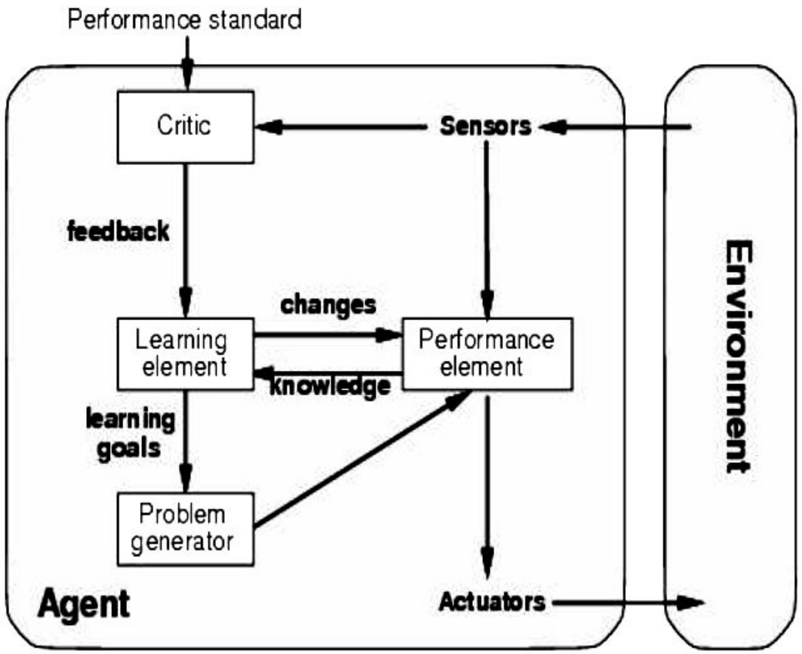
- Some are better: have a higher **utility**

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

Improves on goals:

- Selecting between conflicting goals
- Selecting between several goals based on likelihood of success and importance of goals

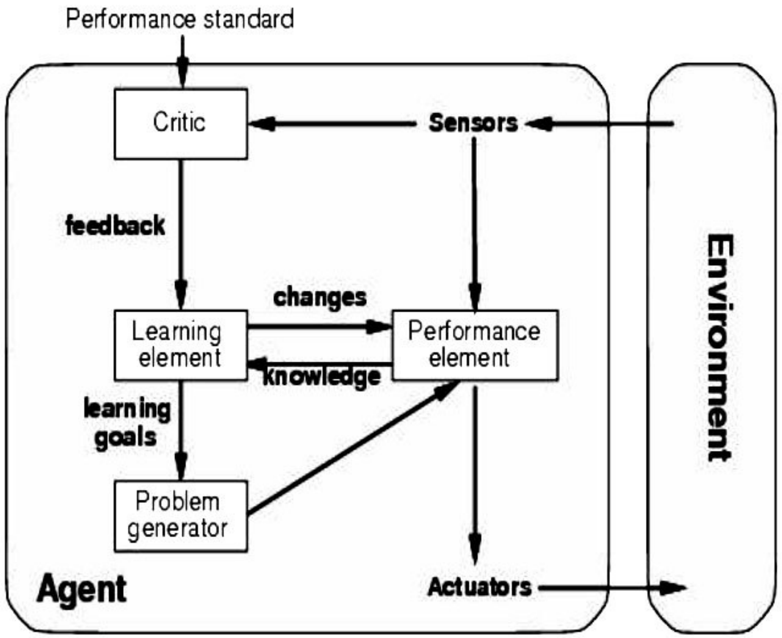
Agent types: Learning



All previous agent-programs describe methods for selecting *actions*

- Not yet explained the origin of these programs
- Learning mechanisms can be used to create the programs
- One advantage is the robustness of the program toward initially unknown environments

Agent types: Learning



- *Learning element*: introduce improvements in performance element
- Critic provides feedback on agent's performance based on fixed performance standard
- *Performance element*: selecting actions based on percepts
  - Corresponds to previous agent programs
- *Problem generator*: suggests actions that will lead to new and informative experiences

Procedural vs Declarative Programming

The emphasis in Procedural Programming is on how to produce a result.

The emphasis in Declarative Programming is on what can be discovered from the stored knowledge.

A non-procedural program (or knowledge base) often consists of a set of statements or declarations about some subject or domain.

Logic Programming

A logic programming system, such as a Prolog interpreter, mechanises the rules of logic.

E.g., if we know some facts about the world:

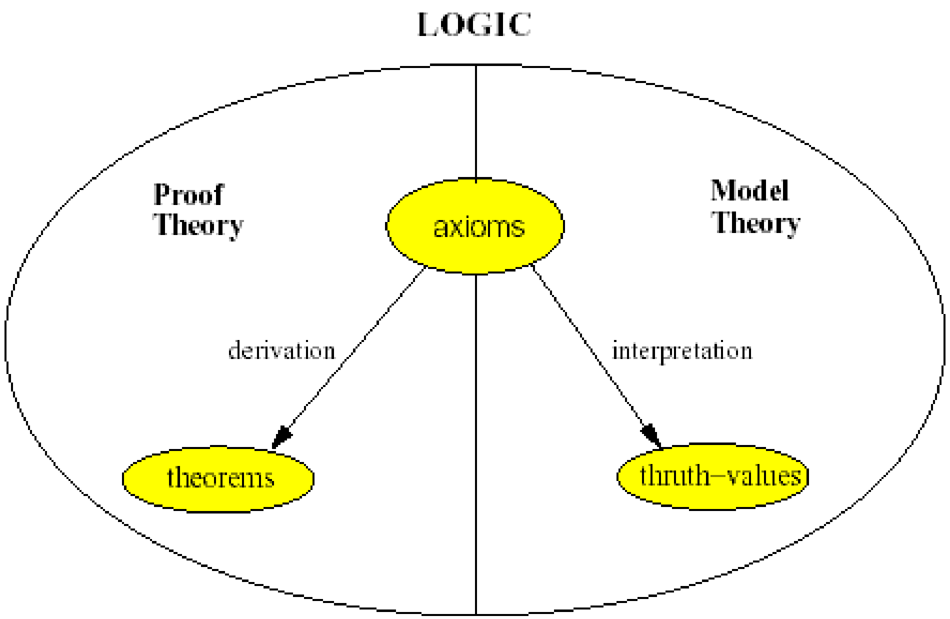
All men are mortal

Socrates is a man

We can deduce logically that Socrates is mortal! This is exactly what the Prolog interpreter would do.

What is Logic?

Definition: Logic is the branch of mathematics that investigates the relationships between premises and conclusions of arguments. Put simply, logic is the study of what follows from what, that is, what conclusions follow from a set of premises.



Logical Arguments

An argument is made up of a ***statement*** and some supporting ***reasons*** for believing the statement.

In logic, the reasons why a conclusion should be believed are known as the ***premises*** of an argument whereas the statement that the argument is intended to persuade us of is called the ***conclusion***.

An example

Since April 6th 2004, if one earns less than £91 per week in the UK, he/she pays 0% national insurance. Bob is a low-paid employee. He makes £80 per week.

Therefore, Bob does not pay any national insurance contribution.

Which are the premises and what is the conclusion of this argument?

Valid Arguments?

Consider the following ***argument***:

Socrates is a man

<u>All men are mortal</u>

Socrates is mortal

Is this a valid argument?

Valid Arguments?

Now, consider this argument:

Every foo is a bar
<u>Every bar is a baz</u>
Every foo is a baz

So, is this a valid argument?

Valid Arguments?

Now, consider this argument:

Some foo is a bar
<u>Some bar is a baz</u>
Some foo is a baz

So, is this a valid argument?

Valid Arguments?

Would it be easier if you replaced foo, bar and baz with words that made sense e.g.

Some student is a Teaching Assistant

<u>Some Teaching Assistant is a Lab Assistant</u>

Some student is a Lab Assistant

Does the argument seem valid now?

A Countermodel?

The argument can also be proved not valid using a countermodel without any other knowledge of the domain.

Imagine a world that contains exactly two people:

1. a student who is a TA but not a Lab Assistant
2. a TA who is a Lab Assistant but not a student

Is the previous argument correct in this world?

Models and Countermodels

Each way that the world can be is known as a ***model***. In propositional logic, a model is just an assignment of truth values to propositional letters.

A model satisfies a sentence if the sentence is **true** in the model A model falsifies a sentence is the sentence is **false** in the model.

A model, that satisfies all the premises of an argument but falsifies the conclusion of the argument is said to be a ***countermodel*** of the argument.

Valid Arguments

Definition: An argument is **valid** if and only if it has no countermodel i.e. if and only if every model that satisfies the argument's premises also satisfies its conclusion.

Artificial Intelligence

CSC348 Unit 1 Part 2: Propositional Logic

Syedur Rahman

Lecturer, CSE Department

North South University

Please note that the following slides are taken directly from S.Kapetanakis' lectures

Logical Symbols

To create arguments that make sense, we must agree on some terms that have a fixed meaning, such as *every*, *some*, etc, which we call **Logical symbols**.

To illustrate, consider the following argument (again):

Every foo is a bar.
Every bar is a baz.
―――――――――――――
Every foo is a baz.

What if the word *every* meant *some* in this world?

Logical vs Non-logical Symbols

The *lexicon* (i.e. non-logical symbols) of propositional logic consists of a set of proposition symbols. We shall assume that $P, Q, R, \ldots$ are the proposition symbols.

The *logical symbols* of propositional logic are: $\neg, \vee, \wedge, \rightarrow,$ and $\leftrightarrow$

Sentences of propositional logic

The sentences (aka well-formed formulae or wffs) of propositional logic are defined by:

- if α is a proposition letter, then α is an atomic sentence
- if α and β are sentences, then $\neg\alpha$, $\alpha \vee \beta$, $\alpha \wedge \beta$, $\alpha \rightarrow \beta$, $\alpha \leftrightarrow \beta$
- nothing else is a sentence

Practice with Logic

Put the following sentences into propositional logic:

1. Ed is wise
2. Ed is not wise
3. Ed is not wise and Ed is not well
4. It's not true that Ed is wise and well
5. If Ed dies he will miss class
6. Either Ed flies and he arrives on time or Ed walks and he is not on time

Symbols to use:

W Ed is wise
C Ed attends class
L Ed is well
D Ed is Dead
F Ed flies
A Ed arrives on time
K Ed Walks

$\neg$, $\vee$, $\wedge$, $\rightarrow$, and $\leftrightarrow$

Syntax of Propositional Logic

Proposition (non-logical) symbols: $P, Q, R, \ldots$
Logical symbols: (not) $\neg$, (or) $\vee$, (and) $\wedge$,
(implies) $\rightarrow$,
(if and only if) $\leftrightarrow$

The sentences (aka well-formed formulae or wffs) of propositional logic are defined by:

- if α is a proposition letter, then α is a wff
- if α and β are wffs, then $\neg\alpha$, $\alpha \vee \beta$, $\alpha \wedge \beta$, $\alpha \rightarrow \beta$, $\alpha \leftrightarrow \beta$ are wffs
- nothing else is a wff

Notation Shortcuts

To reduce strain on the reader and the writer of the logical sentences, certain shortcuts are often taken.

- Connectives are given precedence: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ from high to low. Thus, $\neg A \wedge \neg B \rightarrow C$ is shorthand for the sentence $(((\neg A) \wedge (\neg B)) \rightarrow C)$.
- The connectives $\vee$ and $\wedge$ are sometimes taken to have arbitrary arity of two or more. For example, we may write $(A \vee B \vee C)$ as a valid replacement for $(A \vee (B \vee C))$.

Models and Semantic Values

A *model* or *interpretation* in propositional logic is a *function* that maps every proposition letter to a truth value, either $TRUE$ or $FALSE$.

If α is a sentence and I is a model, then $[\![\alpha]\!]^I$ is the **semantic value** of relative to model I.

Formal Semantics of Prop. Logic

If α is a sentence and I is a model, then α 's semantic value $[\![\alpha]\!]^I$ is defined as follows:

- If α is an atomic sentence then $[\![\alpha]\!]^I = I(\alpha)$.
- If α is $(\neg\beta)$ then

$$[\![\alpha]\!]^I = \begin{cases} \mathit{TRUE} & \text{if } [\![\beta]\!]^I = \mathit{FALSE} \\ \mathit{FALSE} & \text{otherwise.} \end{cases}$$

Formal Semantics of Prop. Logic

- If α is $(\beta \vee \gamma)$ then

$$[\![\alpha]\!]^I = \begin{cases} \mathit{TRUE} & \text{if } [\![\beta]\!]^I = \mathit{TRUE} \text{ or } [\![\gamma]\!]^I = \mathit{TRUE} \\ \mathit{FALSE} & \text{otherwise.} \end{cases}$$

- If α is $(\beta \wedge \gamma)$ then

$$[\![\alpha]\!]^I = \begin{cases} \mathit{TRUE} & \text{if } [\![\beta]\!]^I = \mathit{TRUE} \text{ and } [\![\gamma]\!]^I = \mathit{TRUE} \\ \mathit{FALSE} & \text{otherwise.} \end{cases}$$

Formal Semantics of Prop. Logic

And finally. . .

- If α is $(\beta \leftrightarrow \gamma)$ then

$$[\![\alpha]\!]^I = \begin{cases} \textit{TRUE} & \text{if } [\![\beta]\!]^I = [\![\gamma]\!]^I \\ \textit{FALSE} & \text{otherwise.} \end{cases}$$

Truth Tables

P	Q	$P \to Q$
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

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

In general, we need 2^n number of rows, where n is the total number of proposition letters in our sentence.

Logical Equivalence

Question: Are the sentences P and $\neg\neg P$ logically equivalent?

- $p \vee p \equiv p$
- $p \vee q \equiv q \vee p$
- $p \wedge p \equiv p$
- $\neg(p \vee q) \equiv \neg p \wedge \neg q$

Given two formulae A_1, A_2, if $[\![A_1]\!]^I = [\![A_2]\!]^I$ for all interpretations I, then A_1 is logically equivalent to A_2, denoted $A_1 \equiv A_2$

Two propositions are logically equivalent if the final columns of their truth tables are identical.

Theorems of Equivalence

Theorem 1 Let A and A' be two logically equivalent formulas. If B is a formula that contains A as a subformula then B is logically equivalent to the formula that results from substituting A' for A in B.

Theorem 2 Two sentences, A and B, are logically equivalent if and only if the sentence $A \leftrightarrow B$ is valid.

Theorems of Equivalence (cntd)

Theorem 2 Two sentences, A and B, are logically equivalent if and only if the sentence $A \leftrightarrow B$ is valid.

Proof

A and B are logically equivalent

iff $[\![A]\!]^I = [\![B]\!]^I$ for every model I

iff $[\![A \leftrightarrow B]\!]^I = T$ for every model I

iff $A \leftrightarrow B$ is valid.

Note that a sentence is valid when it is a tautology, i.e. when it is true in all interpretations.

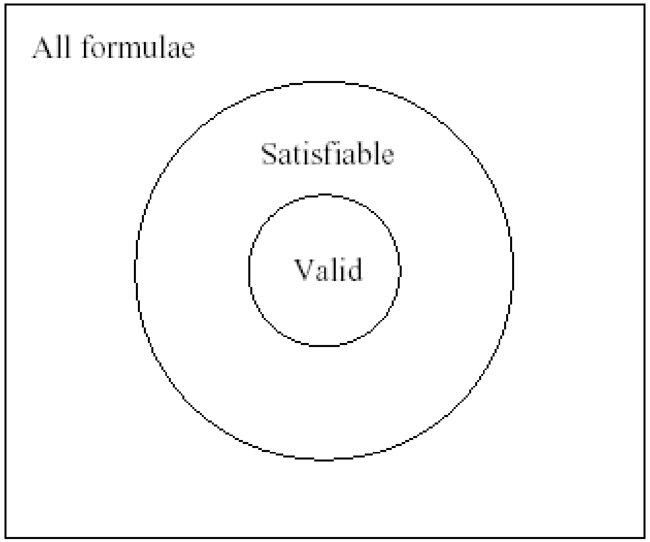
Some useful facts

Let α be a sentence of propositional logic, A be a set of sentences and I be a model/interpretation. Then:

1. I *satisfies* α iff $[\![\alpha]\!]^I = $ *TRUE*.
2. I *satisfies* A iff I satisfies every $a \in A$.
3. I *falsifies* α iff $[\![\alpha]\!]^I = $ *FALSE*.
4. A (or α) is *satisfiable* iff it is satisfied by some model.
5. A (or α) is *unsatisfiable* iff it is not satisfied by *any* model i.e. if it is *FALSE* in *all* models.

Some useful facts (cntd)

1. α is *valid* or *unfalsifiable* iff it is satisfied by every model.
2. α is *non-valid* or *falsifiable* iff it is *FALSE* in some model



Logical Entailment

Let $A, A_1, \ldots, A_n$ be sentences. Then $\{A_1, \ldots, A_n\}$ entails A if and only if all models that make $\{A_1, \ldots, A_n\}$ true also make A true.

Based on the above definition, we can prove the theorem: Let $A, A_1, \ldots, A_n$ be sentences. Then $\{A_1, \ldots, A_n\}$ entails A if and only if the sentence $A_1 \wedge \cdots \wedge A_n \rightarrow A$ is valid.

An Example

Let us assume:

W	George wakes up by 8:00am
C	George catches the 8:30am bus
B	The 8:30am bus is on time and takes 15 mins to reach school
S	School begins at 9:00am
T	George reaches school in time

We can say $\{W, C, B, S\}$ entails T

Since $W \wedge C \wedge B \wedge S \Rightarrow T$ *is valid*

Summary: Connectives and Rules in Propositional Logic

high priority	
$\neg$	not
$\wedge$	and
$\vee$	or
$\Rightarrow$	implies
$\Leftrightarrow$	if and only if
low priority	

xor	**exclusive or**	$p \oplus q \equiv (p \vee q) \wedge \neg(p \wedge q)$
$\downarrow$	**negated or**	$p \downarrow q \equiv \neg(p \vee q)$
$\uparrow$	**negated and**	$p \uparrow q \equiv \neg(p \wedge q)$
$\Rightarrow$	**implication**	$p \Rightarrow q \equiv \neg p \vee q$
$\Leftrightarrow$	if and only if	$p \Leftrightarrow q \equiv (q \Rightarrow p) \wedge (p \Rightarrow q)$

$$\left.\begin{array}{lll} p \vee p & \equiv & p \\ p \wedge p & \equiv & p \end{array}\right\} \textit{idempotence}$$

$$\left.\begin{array}{lll} (p \vee q) \vee r & \equiv & p \vee (q \vee r) \\ (p \wedge q) \wedge r & \equiv & p \wedge (q \wedge r) \end{array}\right\} \textit{associativity}$$

$$\left.\begin{array}{lll} p \vee \neg p & \equiv & T \\ p \wedge \neg p & \equiv & F \end{array}\right\} \textit{excluded middle}$$

$$\left.\begin{array}{lll} p \vee q & \equiv & q \vee p \\ p \wedge q & \equiv & q \wedge p \end{array}\right\} \textit{commutativity}$$

$$\left.\begin{array}{lll} p \vee F & \equiv & p \\ p \wedge T & \equiv & p \end{array}\right\} \textit{identity}$$

$$\left.\begin{array}{lll} p \vee (q \wedge r) & \equiv & (p \vee q) \wedge (p \vee r) \\ p \wedge (q \vee r) & \equiv & (p \wedge q) \vee (p \wedge r) \end{array}\right\} \textit{distributivity}$$

$$\left.\begin{array}{lll} p \vee T & \equiv & T \\ p \wedge F & \equiv & F \end{array}\right\} \textit{strictness}$$

$$\left.\begin{array}{lll} \neg(p \vee q) & \equiv & (\neg p) \wedge (\neg q) \\ \neg(p \wedge q) & \equiv & (\neg p) \vee (\neg q) \end{array}\right\} \textit{de Morgan}$$

$$\neg\neg p \equiv p \quad \textit{double negation}$$

$$\left.\begin{array}{l} p \vee (p \wedge q) \equiv p \\ p \wedge (p \vee q) \equiv p \end{array}\right\} \textit{absorption}$$

End of Unit 1

Thank You