ECE 479/579
Principles of Artificial Intelligence – Part I
Spring 2005

Dr. Michael Marefat (marefat@ece.arizona.edu)

Required text

"*Artificial Intelligence: A Modern Approach, Second Edition*"

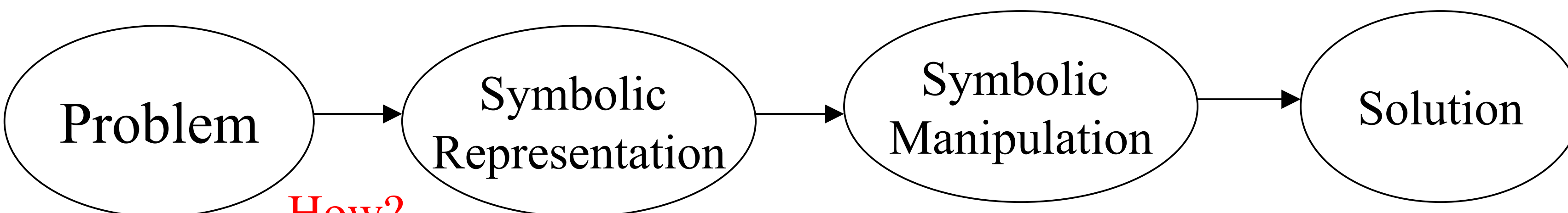
by

Stuart Russell & Peter Norvig, Prentice Hall, 2003. ISBN:0-13-790395-2.

What is Artificial Intelligence?

Computational methods for enabling computers to perform tasks attributed to having having intelligence, such as Design, Configuration, Learning, etc.

Common Scenario:



How?

Logical Language

<<< Refer to Chapter 1 of the text >>>

First Order Predicate Logic(FOPL)

It is a language.

It is defined by its syntax and semantics.

Well formed formulas (wffs) are expressions in FOPL.

<<< Refer to Chapter 8 of the text >>>

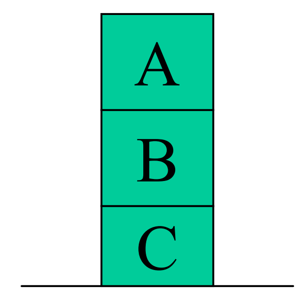
Representations of Knowledge

Logic based methods

First Order Predicate Logic: A language.

Everything is represented as a sentence: wffs, well-formed formulas.

Logical Connectives	Predicates	Terms	Quantifiers
$\wedge$	On( )	Constants A	$\forall$
$\vee$	On(A,B)	Variables ?X	$\exists$
$\neg$		Function f(?X)	
$\Rightarrow$			



<<< Refer to Section 8.1 of the text >>>

Bigdog has a high speed, and it is accessible to all engineering students, or Bigdog cannot be a college-wide computer. All computers that are college-wide computers are Unix-Servers.

Has_High_Speed (Bigdog)

Boolean Combinations:

1. $\wedge$: "and"
2. $\vee$: "or"
3. $\neg$: "not"
4. $\Rightarrow$: "Implies"

Example:

A: Antecedent.

Job (Person1, Engineer) $\Rightarrow$ Is_A (Person1, Middle_Class)

C: Conclusion

A	C	$A \Rightarrow C$
T	T	T
T	F	F
F	T	T
F	F	T

Ex: Represent the following in Predicate Logic:

1. John lives in a Yellow House.

Is_A (John, Person) $\wedge$ Lives_in (House1, John) $\wedge$ Color (House1, Yellow)

2.If the car is red , it is a Chevy.

[Is_A (Car1, Car) $\wedge$ Color (Car1, Red)] $\Rightarrow$ Is_A (Car1, Chevy)

Formulas in which all arguments are constants or functions of constants are ***Ground formulas***. Non-ground formulas have variables.

Fido is a dog,

Fido belongs to John,

John is at school,

John’s dog goes wherever he goes.

Where is Fido?

Is_A(Fido, Dog)

In order to interpret Predicate Logic representations:

1. To each Predicate symbol, we must assign a corresponding relationship in the domain.

2. To each Constant, We must assign a unique entity.

3. To each function, we must assign a mapping in the domain.

These assignments together define the semantics of the given Predicate Logic language.

Once semantics is established, we determine the truth(T) or False(F) of each wff sentence. The value of wff should be T if the corresponding interpretation is correct in the domain.

In converting Problems to Symbolic Representations we must come up with an Ontology.

Ontology consists of
- Predicates
- Constants
- Variables

This process is referred to as Ontologic Engineering.

<<< Refer to Section 10.1 of the text >>>

Ex: (with variable)

Married [father(?x), mother (?x)]

Predicate   function   function

By convention:

- Predicated started with Caps.
- functions started with lower case.
- Constants started with Caps.
- Vars. written with ? as first letter : ?x

We can substitute for Variables (but not for constants):

Married [father (Person1), mother (Person1)]

Ex: If the building is single-story, it is not built by Duncan.

[Is-A (Building 1, Building) $\wedge$

No_of_Stories (Building1, 1)] $\Rightarrow$

$\neg$ Built (Building1, Duncan)

Properties of Wffs

If the truth values of two wffs are the same regardless of their interpretation, these are equivalent. The equivalences below can be established by truth tables:

Formula	Equivalent
$\neg(\neg W1)$	$W1$
$W1 \vee W2$	$\neg W1 \Rightarrow W2$
$\neg(W1 \vee W2)$	$\neg W1 \wedge \neg W2$
$\neg(W1 \wedge W2)$	$\neg W1 \vee \neg W2$
$W1 \wedge (W2 \wedge W3)$	$(W1 \wedge W2) \wedge W3$
$W1 \vee (W2 \vee W3)$	$(W1 \vee W2) \vee W3$
$W1 \Rightarrow W2$	$\neg W2 \Rightarrow \neg W1$

Quantification

Allows us to distinguish between assignments to variable values.

a. Universal Quantifier

∀?x in front of W(?x), we get ∀?x W(?x), this is T, if for all values of ?x, W(?x) is true.

(∀?x) [Animal_type (?x, Elephant) ⇒ Color (?x, Gray)]

b. Existential Quantifier

(∃?x) in front of a formula W(?x), (∃?x) W(?x), it is true, if some assignment of value to ?x in domain, makes W(?x) correct.

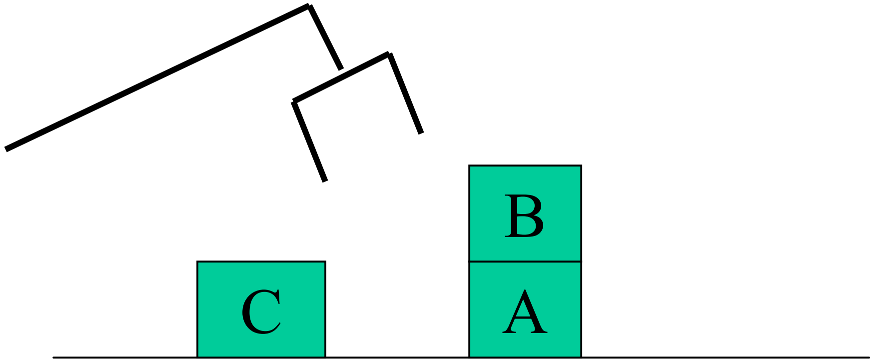
(∃?x)[President_of (US, ?x)]

<<< Refer to Section 8.2 pg. 249 of the text >>>

Example:

All blocks on top of moved blocks, or attached to moved blocks, are also moved.

$(\forall ?x)\ (\forall ?y)\ \{\{Is\_A(?x, Block) \wedge Is\_A(?y, Block) \wedge [On\_top\_of(?x,?y) \vee Attached(?x,?y)] \wedge Done(?y, Move)\} \Rightarrow Done(?x,Move)\}$



For Every set x, there is a set y, such that the cardinality of y is greater than the cardinality of x.

Some inference Rules applied to wffs in FOPL.

Rules of inference are applied to wffs to produce new results.

Universal Specialization:

$(\forall ?x)\ W1(?x)$	a wff in database
A	a constant in database

→ Deduce W1(A) ——— Theorem

Modus Ponens:

$W1(?x) \Rightarrow W2(?x)$ a wff in DB

$W1(A)$ a wff in DB

Deduce → $W2(A)$

$W2(?y) \Rightarrow W3(?y,?z)$ → $W3(A,?z)$

Modus Tollens:

$W1(?x) \Rightarrow W2(?x)$ a wff in DB

$\neg W2(A)$ a wff in DB

Deduce → $\neg W2(A)$

Theorems

Derived wffs are called theorems. A proof is a sequence of inference rules used to derive a theorem.

Equivalence Properties for Quantified formulas

Formula	Equivalent
$\neg(\exists ?x)\ W(?x)$	$(\forall ?x)\ [\neg W(?x)]$
$\neg(\forall ?x)\ W(?x)$	$(\exists ?x)\ [\neg\ W(?x)]$
$(\forall ?x)[W1(?x) \wedge W2(?x)]$	$(\forall ?x)\ W1(?x) \wedge (\forall ?y)\ W2(?y)$
$(\exists ?x)[W1(?x) \vee W2(?x)]$	$(\exists ?x)\ W1(?x) \vee (\exists ?y)\ W2(?y)$