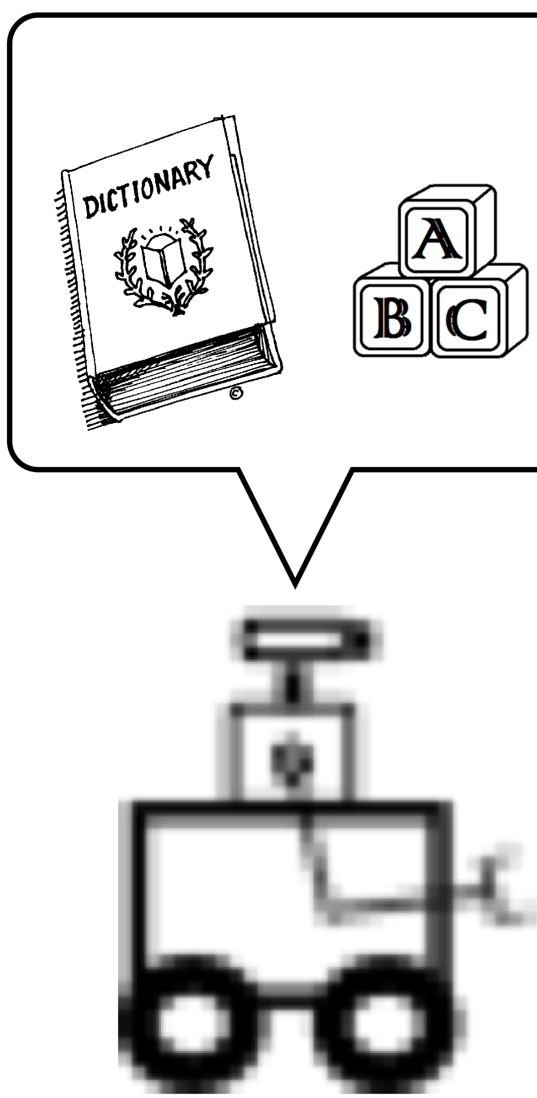
Knowledge Representation & Predicate Logic

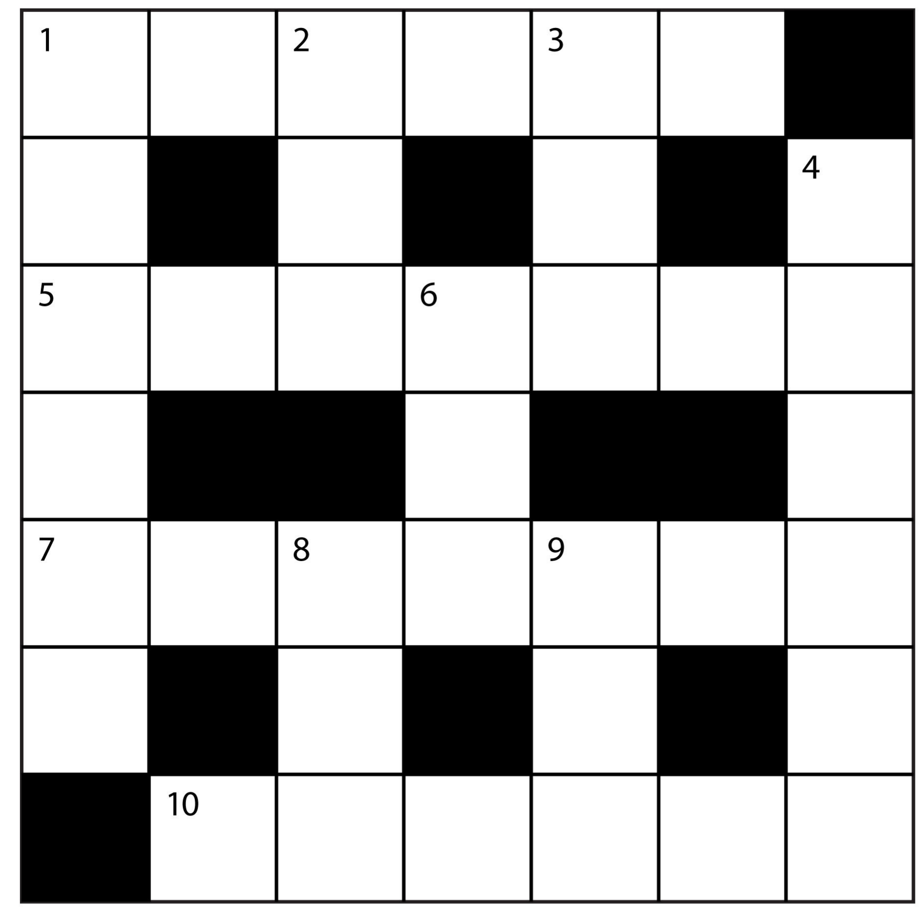
CS 6300
Artificial Intelligence
Spring 2018
Presented By Rush Sanghrajka

Many slides from
George Konidaris
& Mike Stilman

Some examples frm
Michael Huth
& Mark Ryan

(Domain) Knowledge





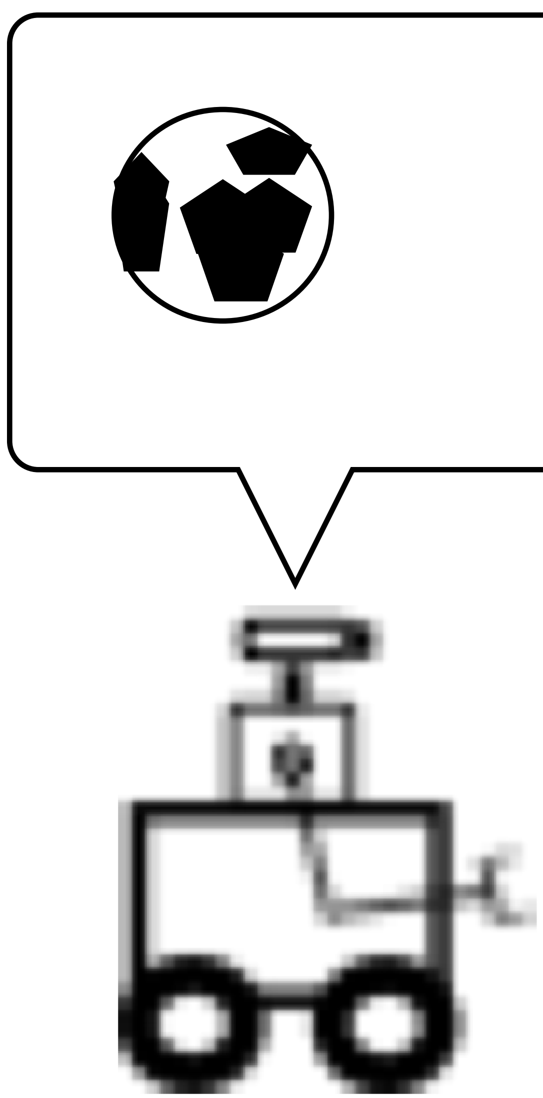
Across

1 By the way, in the centre of the Earth is a dead body (6)
5 Reminder for men engrossed in cryptic tome (7)
7 Left to consume her hide (7)
10 Cold and stiff, following failure (6)

Down

1 Go along with company and imply I am missing (6)
2 Sheep butt? (3)
3 Alas, I never can hide an evil deed (3)
4 Drunkenly rode up and filled a glass (6)
6 Devour meaty innards (3)
8 Element of pretension? (3)
9 Gigantic, endless embrace (3)

Knowledge



State Representations

So far how have we represented states?

- Vectors, matrices, nodes, etc.
- These are all *explicit* representations

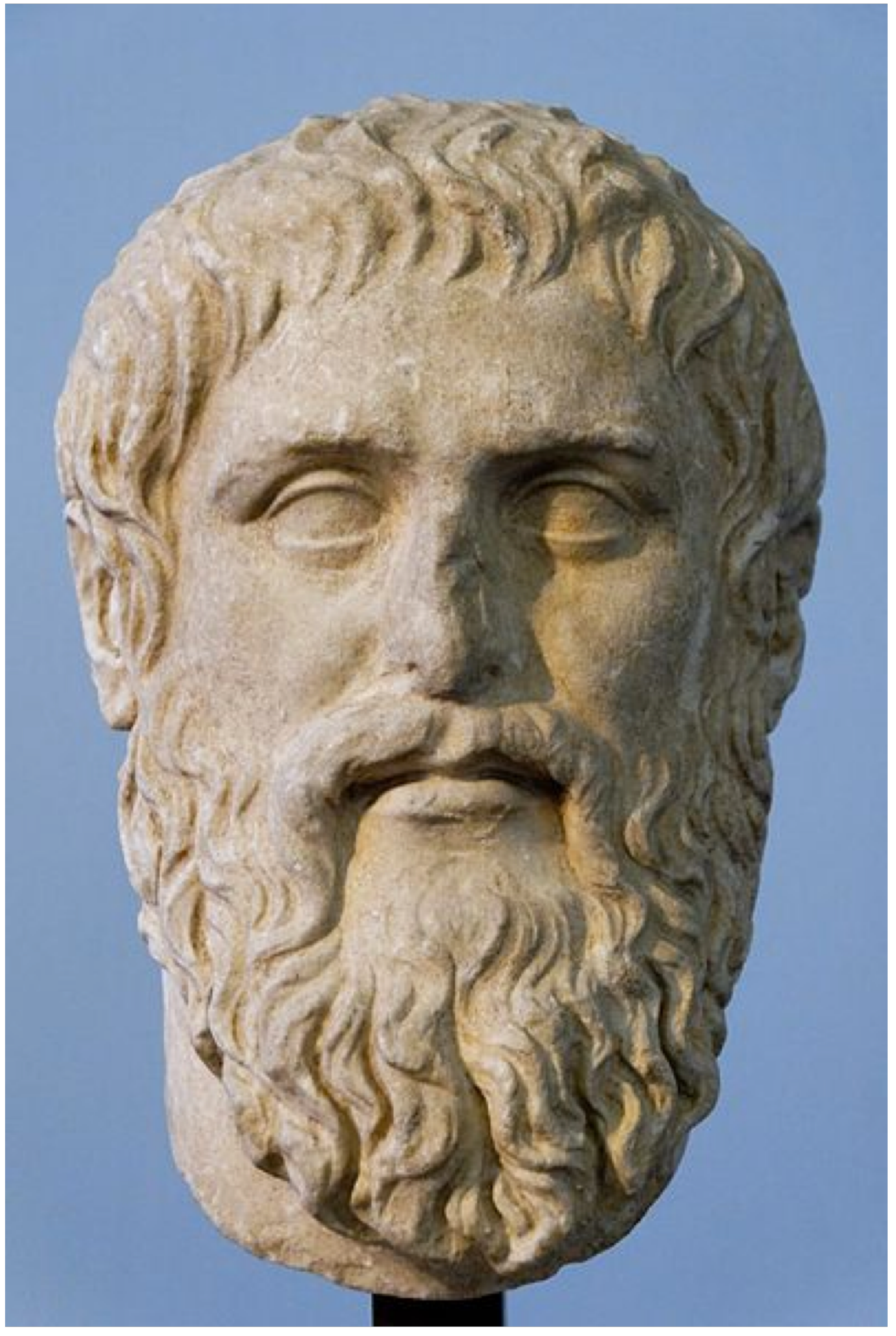
What is the alternative?

- An *implicit* representation,
  but what does that mean?
- Just represent what is known!

Which is better?

Other thoughts?

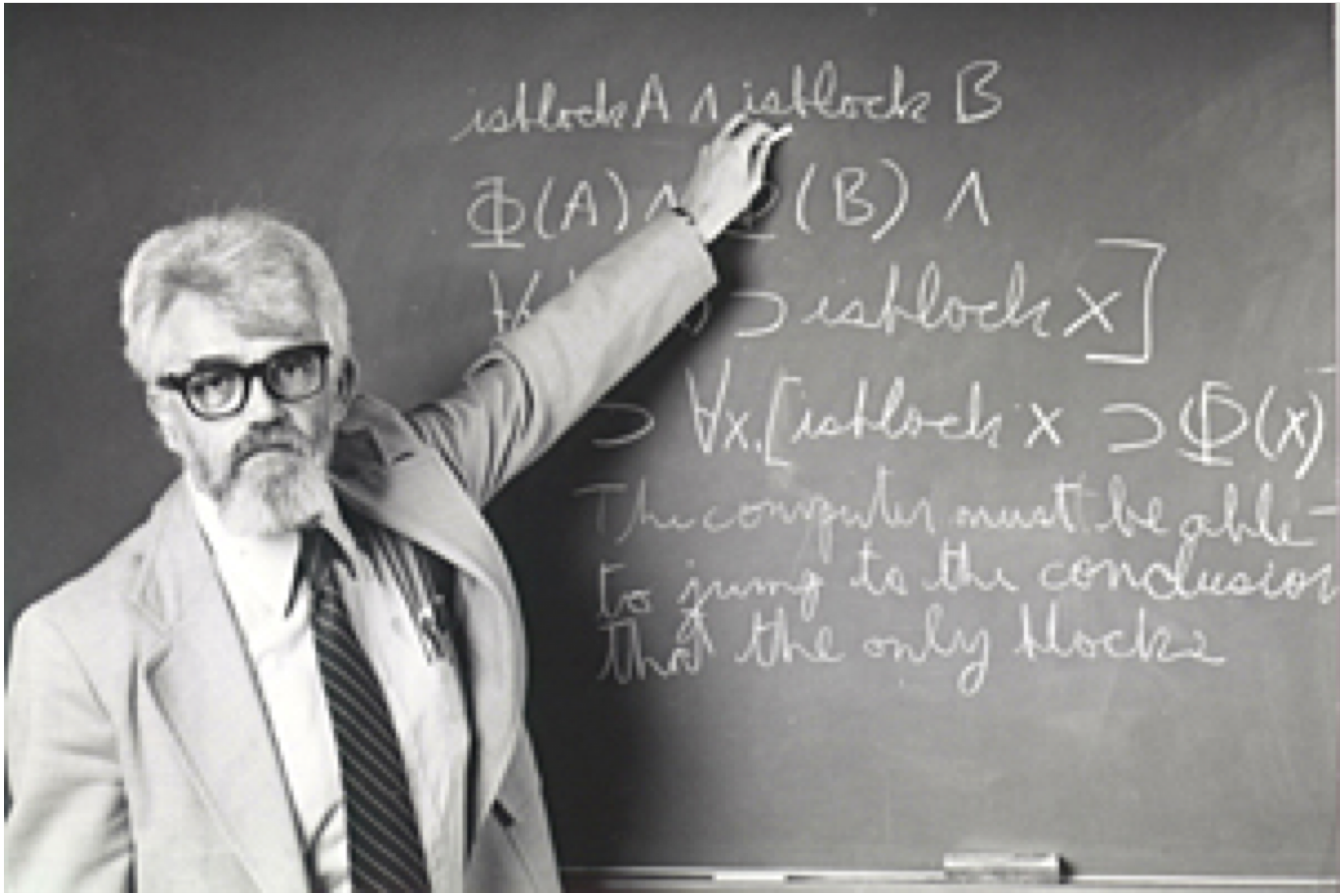
Starting from the Beginning



Plato's Dialogue
In
Theaetetus

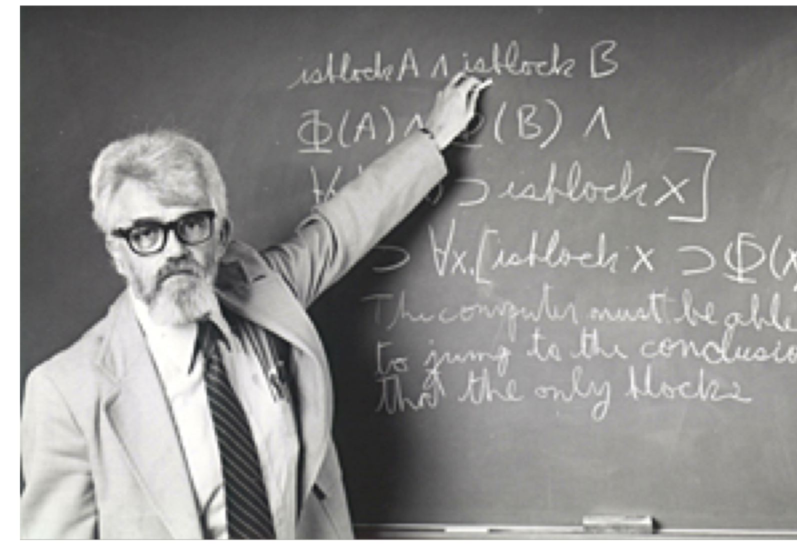
Knowledge is a matter of recollection.
"Justified true belief"

Starting from the Beginning…of AI



Starting from the Beginning…of AI

- 1958 Advice Taker - McCarthy
- 1969 McCarthy and Hayes
  "Some Philosophical Problems From the Standpoint of Artificial Intelligence"



- **Use a Model of the World to answer:**
  - What will happen next in a certain aspect of the situation?
  - What will happen if I do a certain action?
  - What is 3+3?
  - What does he want?
  - Can I figure out how to do this or must I get information from someone else or something else?

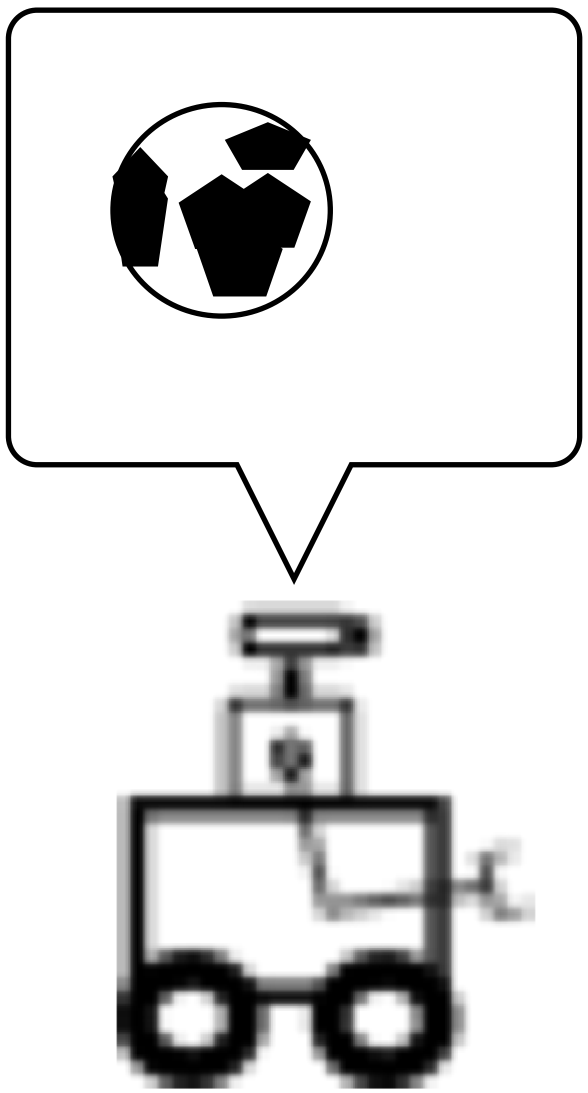
Representation and Reasoning

Represent knowledge about the world.

- Representation language.
- Knowledge base.
- Declarative - *facts* and *rules*.

Reason using that represented knowledge.

- Often *asking questions.*
- Inference procedure.
- Heavily dependent on language.



Example

If the train arrives late and there are no taxis at the station, then John is late for his meeting.

John is not late for his meeting.

The train did arrive late.

Therefore, there were taxis at the station.

Example 2

If it is raining and Jane does not have her umbrella, she will get wet.

Jane is not wet.

It is raining.

Therefore, *Jane has her umbrella.*

Propositional Logic

Representation language and *set of inference rules* for reasoning about facts that are either true or false.

Model the world as a set of *propositions*:

- *Raining*
- *JaneHasUmbrella*
- *TrainIsLate*

Each proposition is either *True* or *False* (though we may not know which).

Propositional Logic

Can combine propositions using **logical operators** to make **sentences *(syntax vs. semantics)*:**

Connectives :

$\neg A$	(not A - A is *False*)
$A \vee B$	(A or B - one (or both) of A or B is *True*)
$A \wedge B$	(A and B - both A and B are *True*)
$A \implies B$	(A implies B - if A is *True*, so is B)
$A \iff B$	(A iff B - A and B both *True* or both *False*)

Two uses of sentences:

- Fact
- Question

Knowledge Base

A list of *sentences that apply to the world*.

For example:

$$Cold$$

$$\neg Raining$$

$$(Raining \vee Cloudy)$$

$$Cold \iff \neg Hot$$

A knowledge base describes *a set of worlds in which these facts and rules are true.*

Knowledge Base

A *model* is a formalization of a "world":

- Set the value of every variable in the KB to *True* or *False*.
- How many models are possible?
- *2ⁿ* models possible for *n* propositions.

Proposition	**Value**
Cold	False
Raining	False
Cloudy	False
Hot	False

Proposition	**Value**
Cold	True
Raining	False
Cloudy	False
Hot	False

...

Proposition	**Value**
Cold	True
Raining	True
Cloudy	True
Hot	True

Models and Worlds

Each sentence has a *truth value* in each model.

Proposition	Value
Cold	True
Raining	False
Cloudy	True
Hot	True

If sentence *a* is true in model *m*, then *m* **satisfies** (or **is a model of**) *a*.

$Cold$	True
$\neg Raining$	True
$(Raining \vee Cloudy)$	True
$Cold \iff \neg Hot$	False

Models and Worlds

The KB specifies a subset of all possible models - those that satisfy all sentences in the KB.

Proposition	Value
Cold	False
Raining	False
Cloudy	True
Hot	True

Proposition	Value
Cold	True
Raining	False
Cloudy	False
Hot	False

…

Proposition	Value
Cold	True
Raining	True
Cloudy	True
Hot	True

Each new piece of knowledge narrows down the set of possible models.

Inference

So if we have a KB, then what?

We'd like to ask it *questions*.

Given:

$$Cold$$

$$\neg Raining$$

$$(Raining \vee Cloudy)$$

$$Cold \iff \neg Hot$$

… we can ask:

$$Hot?$$

Inference: process of deriving new facts from given facts.

Inference (Formally)

KB *A* **entails** sentence *B*

$$A \models B$$

if and only if:

every model which satisfies *A*, satisfies *B*.

In other words: if *A* is true then *B* **must be true**.

That's nice, but how do we compute?

Could just enumerate worlds …

Logical Inference

Take a KB, and produce new sentences of knowledge.
Most frequently, determine whether $KB \models Q$

Inference algorithms: search process to find a proof of *Q* using a set of *inference rules.*

Desirable properties:

- Soundness (or truth-preserving)
- Completeness

Inference Rules

Form	Description
$(A \land B) \equiv (B \land A)$	Commutivity of $\land$.
$(A \lor B) \equiv (B \lor A)$	Commutivity of $\lor$.
$((A \land B) \land C) \equiv (A \land (B \land C))$	Associativity of $\land$.
$((A \lor B) \lor C) \equiv (A \lor (B \lor C))$	Associativity of $\lor$.
$\neg(\neg A) \equiv A$	Double negative elimination.
$(A \Rightarrow B) \equiv (\neg B \Rightarrow \neg A)$	Contraposition.
$(A \Rightarrow B) \equiv (\neg A \lor B)$	Implication elimination.
$(A \Leftrightarrow B) \equiv (A \Rightarrow B) \land (B \Rightarrow A)$	Biconditional elimination.
$\neg(A \land B) \equiv (\neg A \lor \neg B)$	De Morgan.
$\neg(A \lor B) \equiv (\neg A \land \neg B)$	De Morgan.
$(A \land (B \lor C)) \equiv ((A \land B) \lor (A \land C))$	Distributivity of $\land$ over $\lor$.
$(A \lor (B \land C)) \equiv ((A \lor B) \land (B \lor C))$	Distributivity of $\lor$ over $\land$.

If $A \implies B$, and *A* is true, then *B* is true.

If $A \land B$, then *A* is true, and *B* is true.

Inference Rules

Often written in form:

Given this knowledge

Start with

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

can infer this

Proofs

For example, given KB:

$Cold$

$\neg Raining$

$(Raining \vee Cloudy)$

$Cold \iff \neg Hot$

Inference:

$Cold = True$

$True \iff \neg Hot$

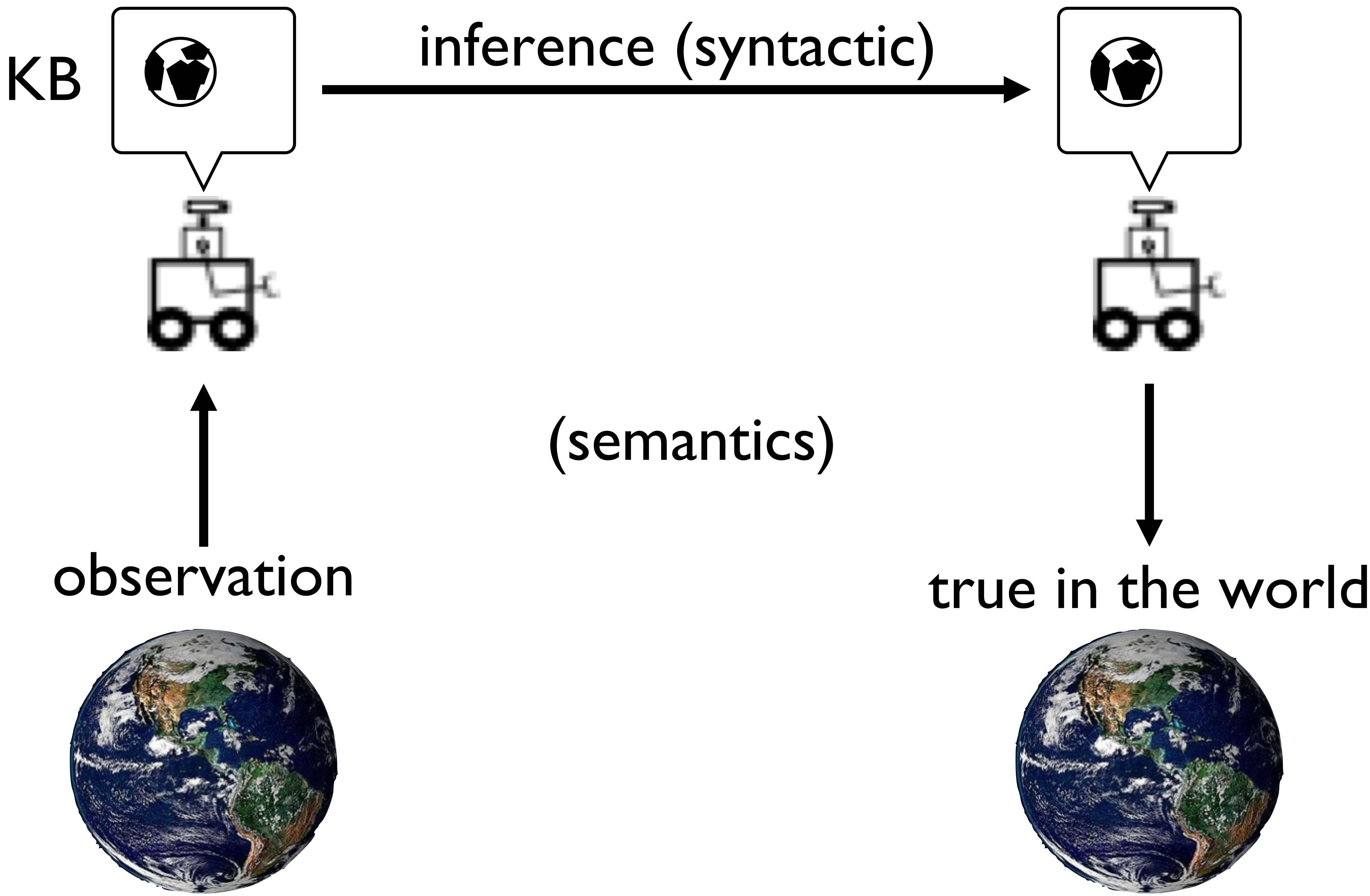
$\neg Hot = True$

$Hot = False$

We ask:

$Hot?$

The World and the Model



DENDRAL and MYCIN

"**Expert Systems**" - knowledge based.

DENDRAL: (Feigenbaum et al. ~1965)

- Identify unknown organic molecules
- Eliminate most "chemically implausible" hypotheses.

MYCIN: (Shortliffe et al., 1970s)

- Identify bacteria causing severe infections.
- "research indicated that it proposed an acceptable therapy in about 69% of cases, which was *better than the performance of infectious disease experts*."

Major issue: **the Knowledge Bottleneck.**

Predicate Logic (First-Order Logic)

More sophisticated representation language.

World can be described by:

Objects (constants)

$Color(\cdot)$

functions

$Adjacent(\cdot, \cdot)$

$IsApple(\cdot)$

relations

First-Order Logic

Objects:

- A “thing in the world”
  - Apples
  - Red
  - The Internet
  - Team Edward
  - Reddit
- A *name* that references something.
- Cf. a *noun*.

$$MyApple271$$

$$TheInternet$$

$$Ennui$$

First-Order Logic

Functions:

- Operator that maps object(s) to single object.
  - $ColorOf(\cdot)$
  - $ObjectNextTo(\cdot)$
  - $SocialSecurityNumber(\cdot)$
  - $DateOfBirth(\cdot)$
  - $Spouse(\cdot)$

$$ColorOf(MyApple271) = Red$$

First-Order Logic

Relations (also called *predicates*):

Like a function, but returns *True* or *False* - holds or does not.

- $IsApple(\cdot)$
- $ParentOf(\cdot, \cdot)$
- $BiggerThan(\cdot, \cdot)$
- $HasA(\cdot, \cdot)$

These are like *verbs* or *verb phrases*.

First-Order Logic

We can build up complex sentences using logical connectives, as in propositional logic:

- $Fruit(X) \implies Sweet(X)$
- $Food(X) \implies (Savory(X) \vee Sweet(X))$
- $ParentOf(Bob, Alice) \wedge ParentOf(Alice, Humphrey)$
- $Fruit(X) \implies Tasty(X) \vee (IsTomato(X) \wedge \neg Tasty(X))$

Predicates can appear where a propositions appear in propositional logic, but *functions cannot*.

Models for First-Order Logic

Recall from Propositional Logic!

A *model* is a formalization of a “world”:

- Set the value of every variable in the KB to *True* or *False*.
- 2^n models possible for n propositions.

Proposition	Value
Cold	False
Raining	False
Cloudy	False
Hot	False

Proposition	Value
Cold	True
Raining	False
Cloudy	False
Hot	False

…

Proposition	Value
Cold	True
Raining	True
Cloudy	True
Hot	True

Models for First-Order Logic

The situation is much more complex for FOL.

A model in FOL consists of:

- A set of objects.
- A set of functions + *values for all inputs.*
- A set of predicates + *values for all inputs.*

Models for First-Order Logic

Consider:

Objects

$Orange$

$Apple$

Predicates

$IsRed(\cdot)$

$HasVitaminC(\cdot)$

Functions

$OppositeOf(\cdot)$

Example model:

Predicate	**Argument**	**Value**
$IsRed$	$Orange$	$False$
$IsRed$	$Apple$	$True$
$HasVitaminC$	$Orange$	$True$
$HasVitaminC$	$Apple$	$True$

Function	**Argument**	**Return**
$OppositeOf$	$Orange$	$Apple$
$OppositeOf$	$Apple$	$Orange$

Knowledge Bases in FOL

A KB is now:

- A set of objects.
- A set of predicates.
- A set of functions.
- A set of sentences using the predicates, functions, and objects, and **asserted to be true**.

Objects

$Orange$

$Apple$

Predicates

$IsRed(\cdot)$

$HasVitaminC(\cdot)$

Functions

$OppositeOf(\cdot)$

$IsRed(Apple)$

$HasVitaminC(Orange)$

Quantifiers

We also have one extra weapon:

- **Quantifiers.**

Quantifiers allow us to make generic statements about properties that hold for the *entire collection of objects* in our KB.

Natural way to say things like:

- All fish have fins.
- All books have pages.
- There is a textbook about AI.

Key idea: **variable + binding rule**.

Existential Quantifiers

There exists object(s) such that a sentence holds.

$$\exists x, IsPresident(x)$$

"there exists"

temporary variable

sentence
using variable

Existential Quantifiers

Examples:

- $\exists x, Person(x) \wedge Name(x, George)$
- $\exists x, Car(x) \wedge ParkedIn(x, E23)$
- $\exists x, Course(x) \wedge Prerequisite(x, CS270)$

Universal Quantifiers

A sentence holds for all object(s).

$$\forall x, HasStudentNumber(x) \implies Person(x)$$

"for every"

sentence
using variable

temporary variable

Universal Quantifiers

Examples

- $$\forall x, Fruit(x) \implies Tasty(x)$$
- $$\forall x, Bird(x) \implies Feathered(x)$$

$$\forall x, Book(x) \rightarrow HasAuthor(x)$$

Quantifiers

Difference in strength:

- Universal quantifier is **very strong**.
  - So use **weak sentence**.

$$\forall x, Bird(x) \implies Feathered(x)$$

- Existential quantifier is **very weak**.
  - So use **strong sentence**.

$$\exists x, Car(x) \land ParkedIn(x, E23)$$

Compound Quantifiers

$$\forall x, \exists y, Person(x) \implies Name(x, y)$$

“every person has a name”

Common Pitfalls

$$\forall x, Bird(x) \land Feathered(x)$$

Common Pitfalls

$$\exists x, Car(x) \implies ParkedIn(x, E23)$$

Inference in First-Order Logic

Ground term, or **literal -** an actual object:

$MyApple12$

vs. a **variable**:

x

If you have only ground terms, you can convert to a propositional representation and proceed from there.

$$IsTasty(Apple) : IsTastyApple$$

Instantiation

Getting rid of variables: **instantiate** a variable to a literal.

Universally quantified:

$\forall x, Fruit(x) \implies Tasty(x)$

$$Fruit(Apple) \implies Tasty(Apple)$$

$$Fruit(Orange) \implies Tasty(Orange)$$

$$Fruit(MyCar) \implies Tasty(MyCar)$$

$$Fruit(TheSky) \implies Tasty(TheSky)$$

For every object in the KB, just write out the rule with the variables substituted.

Instantiation

Existentially quantified:

- Invent a new name **(Skolem constant)**

$$\exists x, Car(x) \wedge ParkedIn(x, E23)$$

$$Car(C) \wedge ParkedIn(C, E23)$$

- Name cannot be one you've already used.
- Rule can then be discarded.

PROLOG

PROgramming in LOGic (Colmerauer, 1970s)

- General-purpose AI programming language
- Based on First-Order Logic
- Declarative

- Use centered in Europe and Japan
- Fifth-Generation Computer Project

- Some parts of Watson (pattern matching over NLP)
- Often used as component of a larger system.

Next time…

We can assert what is true or false in the world…

How can we change that?

- Logical planning
- What is search now?