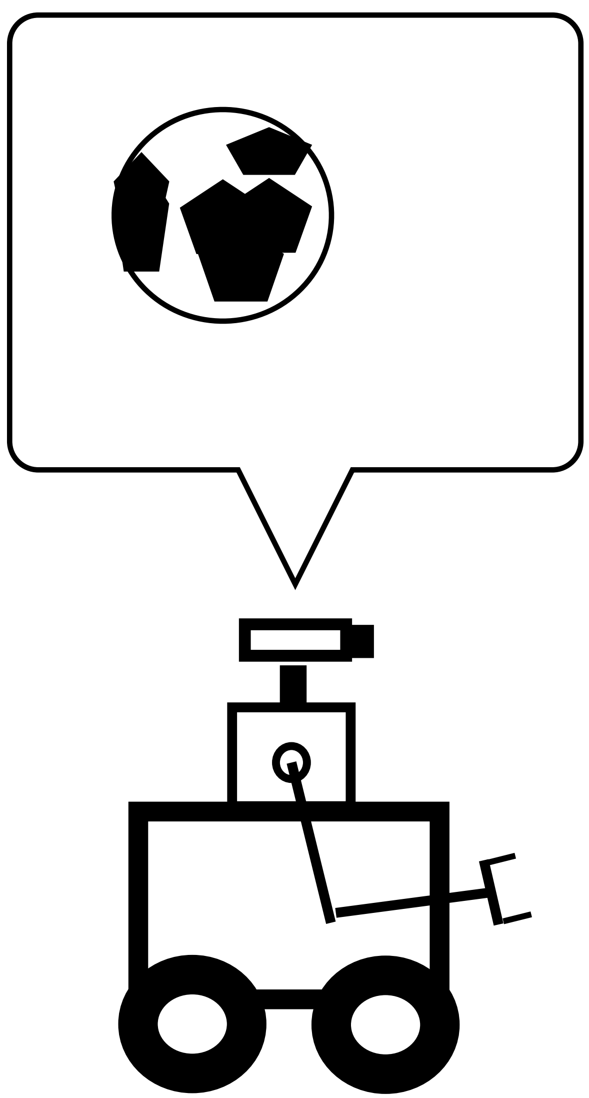
Knowledge Representation and Reasoning (Logic)

George Konidaris
gdk@cs.brown.edu

Fall 2021

Knowledge



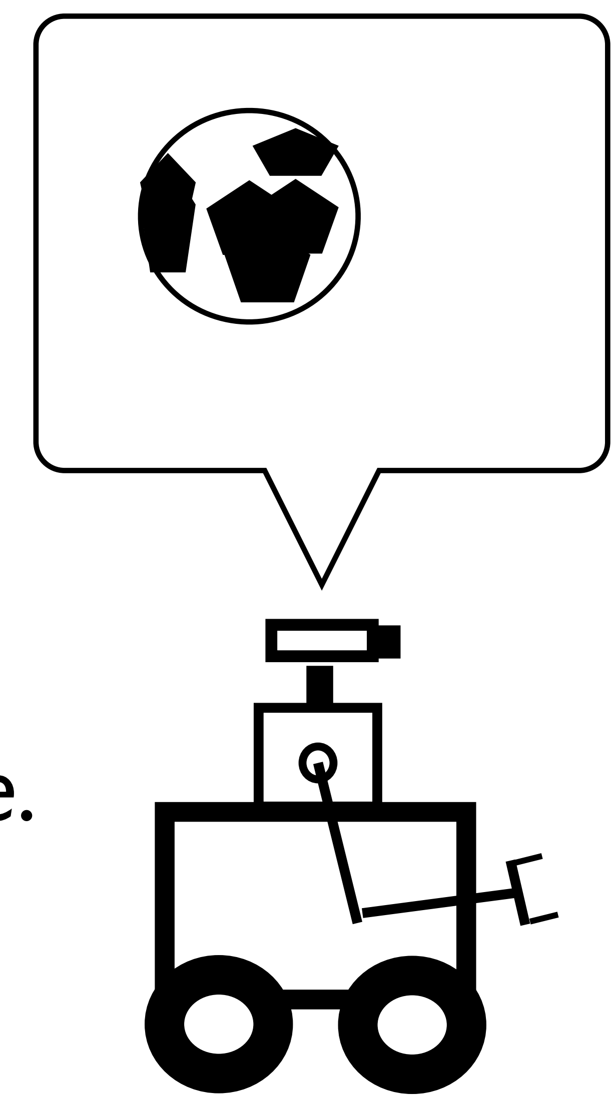
Representation and Reasoning

Represent knowledge about the world.

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

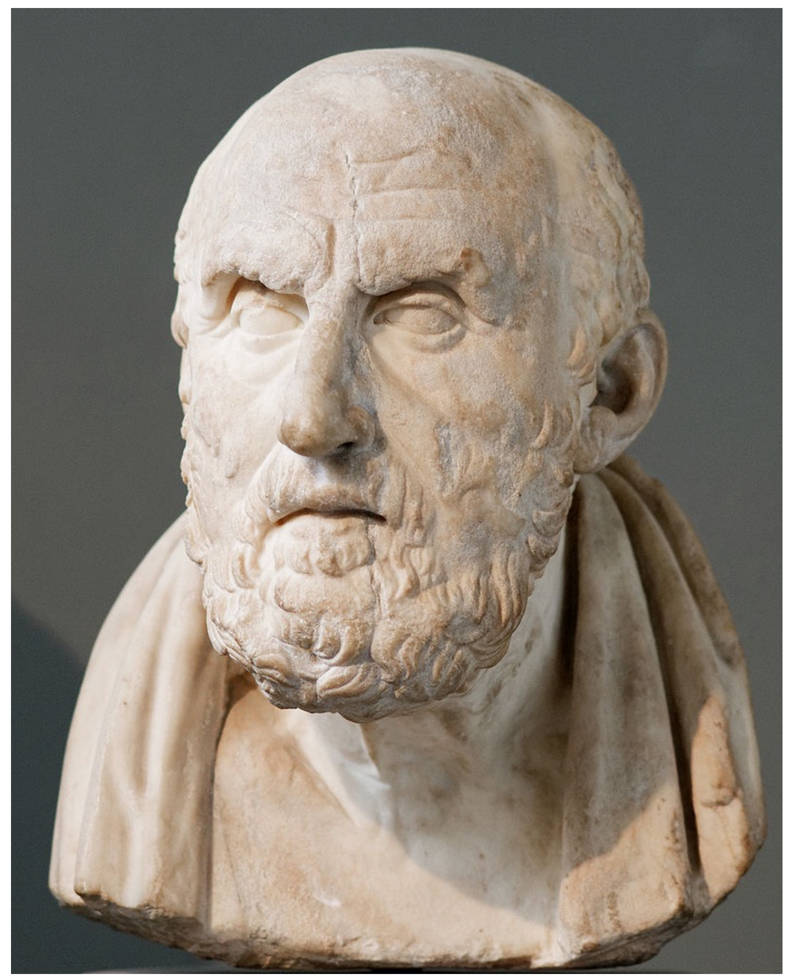
Reason using that represented knowledge.

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



Propositional Logic

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



Chrysippus of Soli, 3rd century BC

"that which is capable of being denied or affirmed as it is in itself"

Knowledge Base

A list of *propositional logic 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*.
- 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

$\dots$

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

Models and Sentences

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

$Cold$

$\neg Raining$

$(Raining \vee Cloudy)$

$Cold \iff \neg Hot$

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

Proposition	Value
Cold	False
Raining	False
Cloudy	False
Hot	False

Proposition	Value
Cold	True
Raining	False
Cloudy	True
Hot	False

...

Proposition	Value
Cold	True
Raining	True
Cloudy	True
Hot	True

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

Summary

Knowledge Base

- Set of facts *asserted to be true* about the world.

Model

- Formalization of "the world".
- An assignment of values to all variables.

Satisfaction

- Satisfies a sentence if that sentence is true in the model.
- Satisfies a KB if all sentences true in model.
- Knowledge in the KB *narrows down* the set of possible world models.

Inference

So if we have a KB, then what?

Given:

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

We'd like to ask it *questions*.

… 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**.
Only conclusions you can make about the true world.

Most frequent form of inference: $KB \models Q$

That's nice, but how do we compute?

Logical Inference

Take a KB, and produce new sentences of knowledge.

Inference algorithms: methods for finding a proof of Q using a set of *inference rules*.

Desirable properties:

- Don't make any mistakes
- Be able to prove all possible true statements

Inference (formally)

Could just enumerate worlds …

✓ ✗ Knowledge Base

✓ ✗ Query Sentence

Proposition	Value
Cold	False
Raining	False
Cloudy	False
Hot	False
✓ (Knowledge Base)	✓ (Query Sentence)

Proposition	Value
Cold	True
Raining	True
Cloudy	True
Hot	True
✗ (Knowledge Base)	✗ (Query Sentence)

Proposition	Value
Cold	True
Raining	True
Cloudy	True
Hot	True
✗ (Knowledge Base)	✓ (Query Sentence)

Proposition	Value
Cold	True
Raining	False
Cloudy	False
Hot	False
✓ (Knowledge Base)	✗ (Query Sentence)

OK (arrows to the first three tables)

Not OK (arrow to the fourth table)

Inference Rules

Often written in form:

Start with

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

Given this knowledge

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:

$Hot?$

Inference …

We want to *start* somewhere (KB).
We'd like to *apply some rules.*
But there are lots of *ways we might go.*
… in order to reach some *goal* (sentence).

Does that sound familiar?

Inference as search:

Set of states	True sentences
Start state	KB
Set of actions and action rules	Inference rules
Goal test	Q in sentences?
Cost function	1 per rule

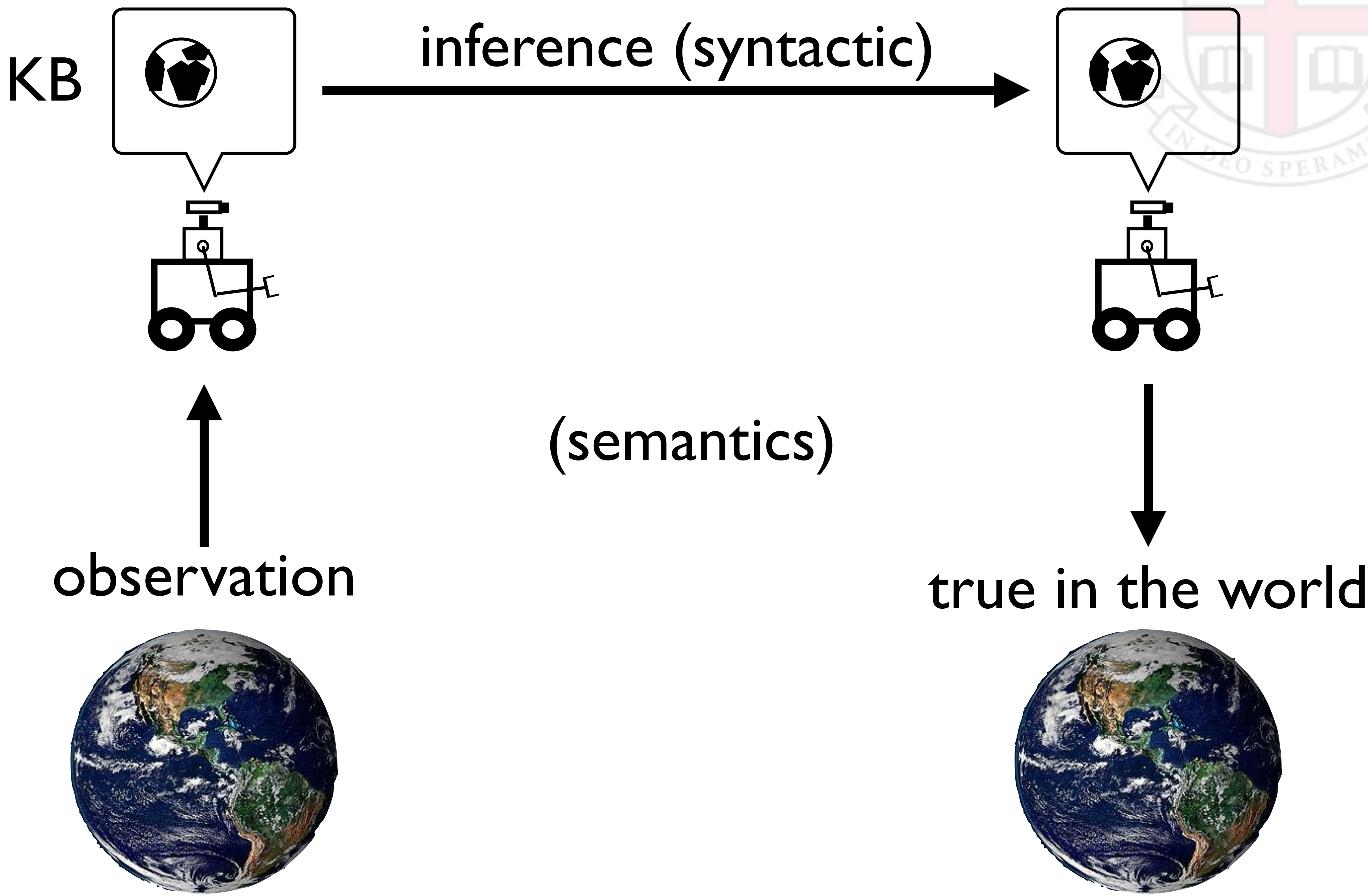
Resolution

The following inference rule is **both sound and complete:**

$$\frac{a_1 \vee \ldots \vee a_{i-1} \vee c \vee a_{i+1} \vee \ldots \vee a_n, \quad b_1 \vee \ldots \vee b_{j-1} \vee \neg c \vee b_{j+1} \vee \ldots \vee b_m}{a_1 \vee \ldots \vee a_{i-1} \vee a_{i+1} \vee \ldots \vee a_n \vee b_1 \vee \ldots \vee b_{j-1} \vee b_{j+1} \vee \ldots \vee b_m}$$

This is called **resolution**. It is sound and complete when combined with a sound and complete search algorithm.

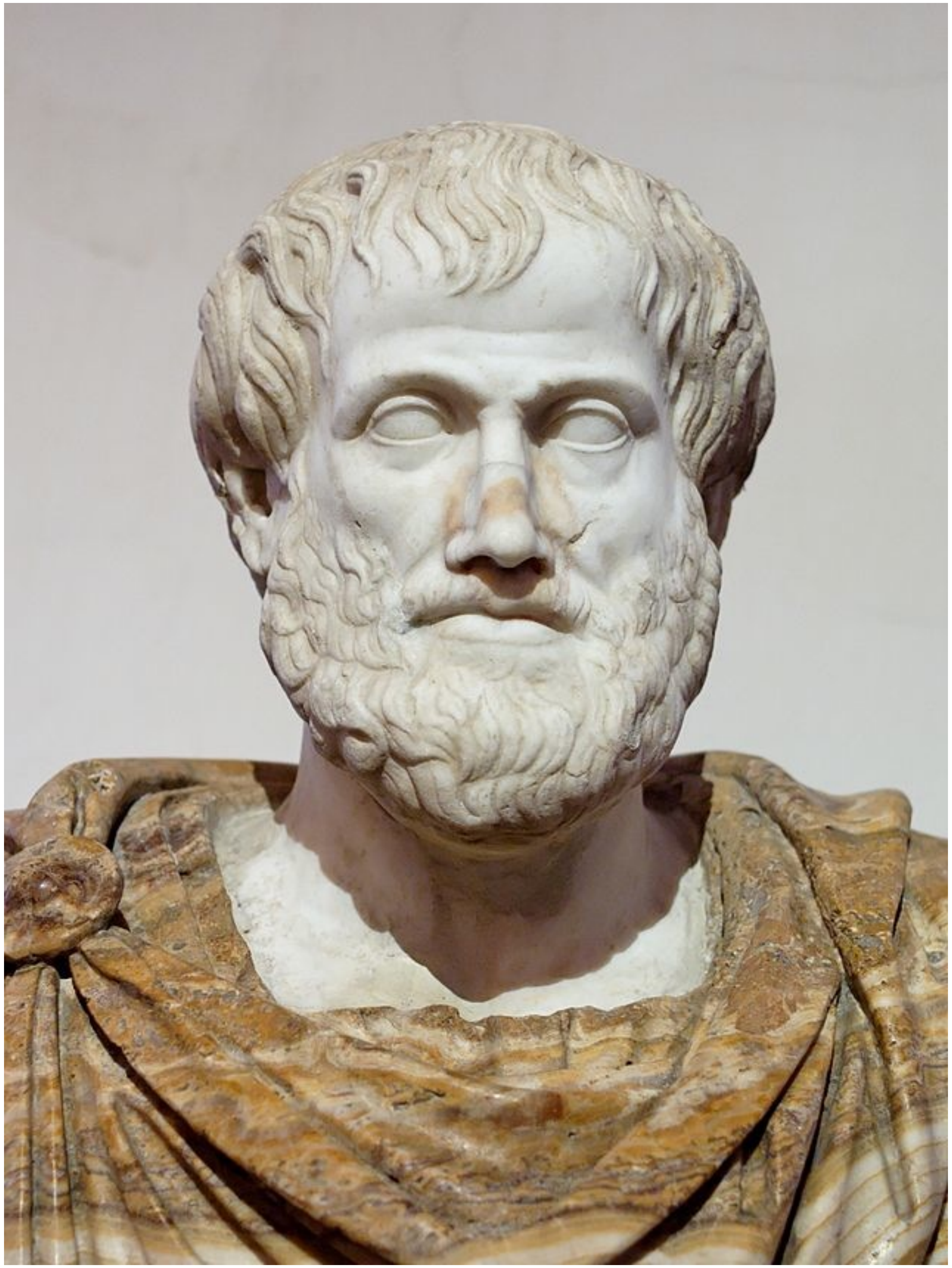
The World and the Model



Languages

Propositional logic isn't very powerful.

How might we get more power?



IN DEO SPERAMUS

First-Order Logic

More sophisticated representation language.

World can be described by:

objects

$ColorOf(\cdot)$

functions

$Adjacent(\cdot, \cdot)$

$IsApple(\cdot)$

predicates

First-Order Logic

Objects:

- A "thing in the world"
  - Apples
  - Red
  - The Internet
  - The Class of 2022
  - 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

Predicates - *replaces proposition*

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

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

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

Propositional logic: for a *model:*

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

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	**Predicates**	**Functions**
$Orange$	$IsRed(\cdot)$	$OppositeOf(\cdot)$
$Apple$	$HasVitaminC(\cdot)$	

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	**Predicates**	**Functions**
$Orange$	$IsRed(\cdot)$	$OppositeOf(\cdot)$
$Apple$	$HasVitaminC(\cdot)$	

vocabulary

$IsRed(Apple)$

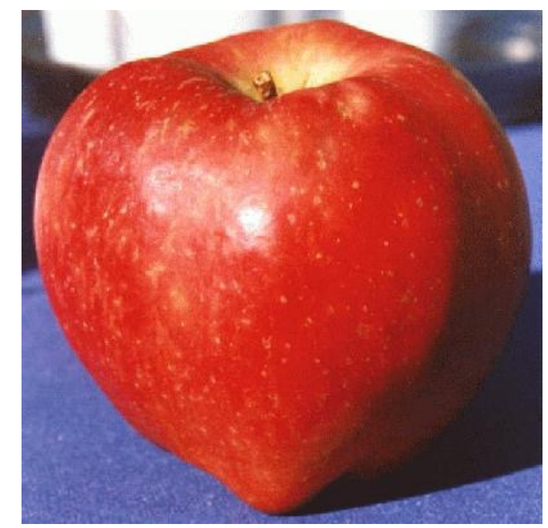
$HasVitaminC(Orange)$

Knowledge Bases in FOL

Listing everything is tedious …

- Especially when general relationships hold.



…


We would like a way to say more general things about the world than explicitly listing truth values for each object.

Quantifiers

New weapon:

- **Quantifiers.**

Make generic statements about properties that hold for the *entire collection of objects* in our KB.

Natural way to say things like:

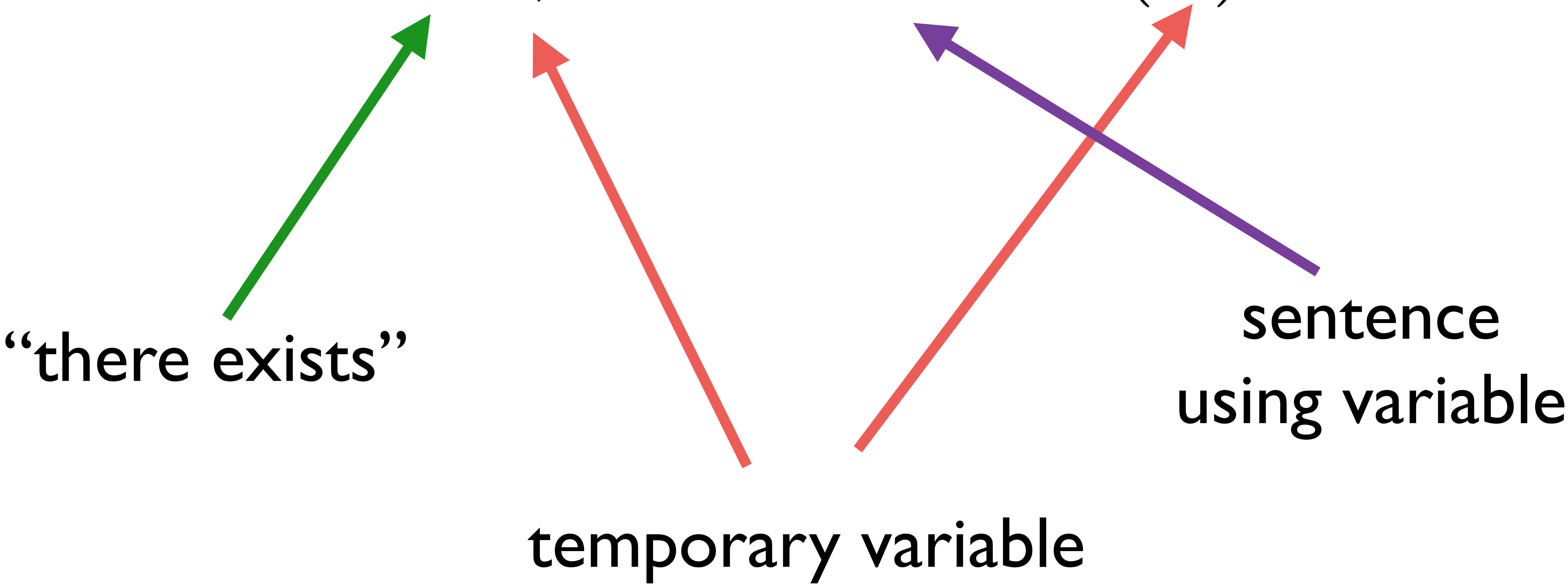
- All fish have fins.
- All books have pages.
- There is a textbook about AI.

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

Existential Quantifiers

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

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



Universal Quantifiers

A sentence holds for all object(s).

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

"for every"

temporary variable

sentence using variable

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

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

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.
Why?

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 system.

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.**