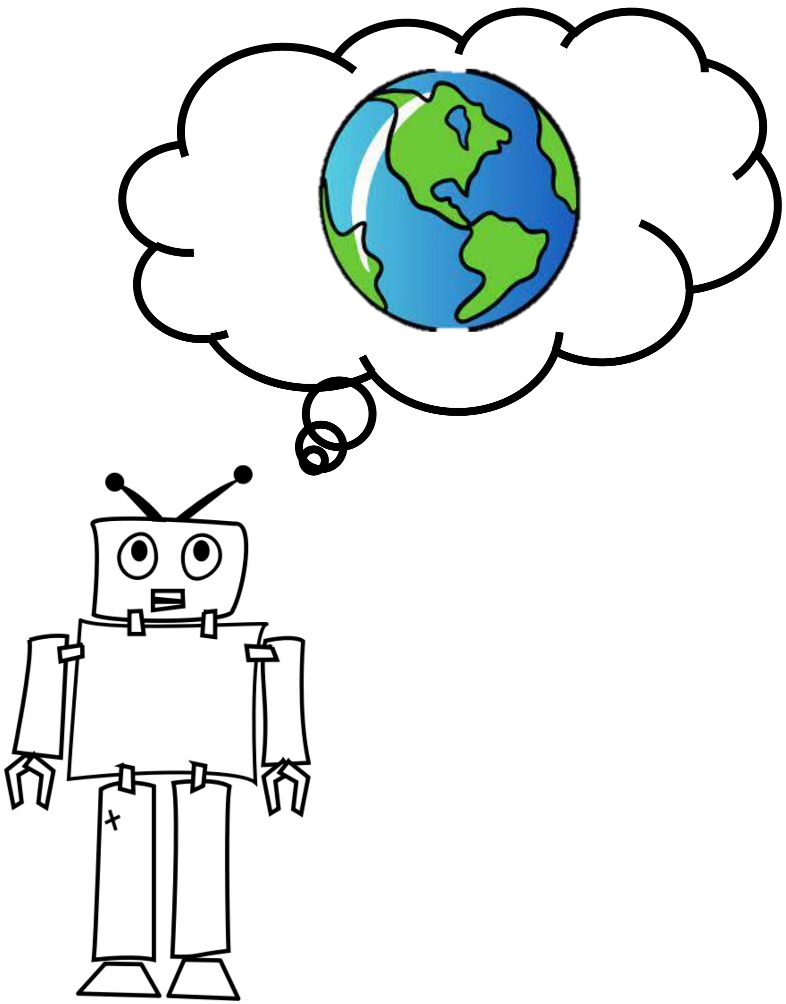
Artificial Intelligence

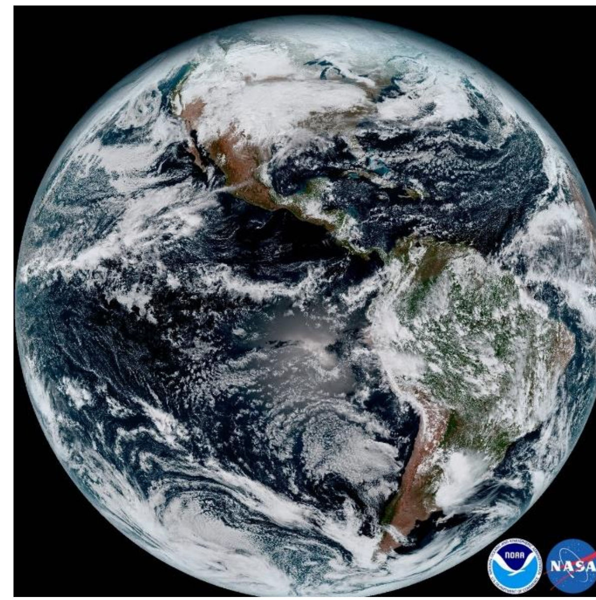
Steve James

Knowledge Representation & Reasoning

(Logic)

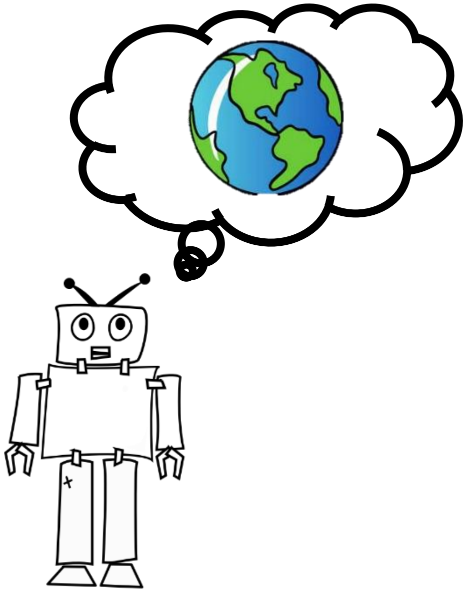
Knowledge





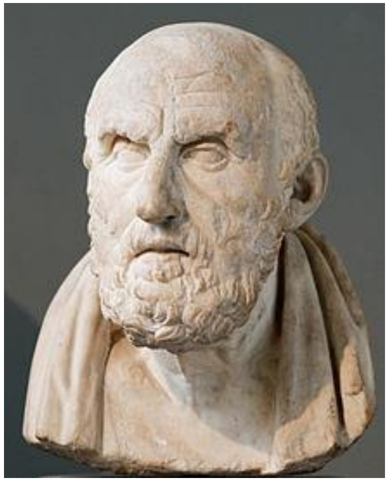
Representation and reasoning

- Represent knowledge about the world
  - Representation language
  - Knowledge base
  - Declarative: facts and rules
- Reasoning using that knowledge
  - Often asking questions
  - Inference procedure
  - Depends on representation language



Propositional logic

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



“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. E.g.

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

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

Knowledge base

- Model is a formalisation of a "world":
  - Set value of each variable in KB to True/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 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

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

- 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	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

Summary

- Knowledge base
  - Set of facts asserted to be true about the world.
- Model
  - Formalisation 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 we have a KB. Now what?
- Given:

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

- We'd like to ask it questions!
  - Hot?
- Inference: process of deriving new facts from given facts

Inference (formally)

- KB A entails sentence B if and only if:
  - Every model which satisfies A, satisfies B

$$A \vDash B$$

- i.e. If A is true, then B must be true
  - Only conclusions you can make about the true world
- Most frequent form of inference: $KB \vDash Q$
- 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 (sound)
  – Be able to prove all possible true statements (complete)

Inference

- Could just enumerate models

Proposition	Value
Cold	False
Raining	False
Cloudy	False
Hot	False

Proposition	Value
Cold	True
Raining	True
Cloudy	True
Hot	True

Proposition	Value
Cold	True
Raining	True
Cloudy	True
Hot	True

Okay

Proposition	Value
Cold	True
Raining	False
Cloudy	False
Hot	False

Not okay

Inference rules

- Often written in the form:

Start with

Given this

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

Can infer this

Proofs

- For example, given KB:

$$Cold$$
$$\neg Raining$$
$$(Raining \lor Cloudy)$$
$$Cold \Leftrightarrow \neg Hot$$

- We ask: Hot?

Inference:

$$Cold = True$$
$$True \Leftrightarrow \neg Hot$$
$$\neg Hot = True$$
$$Hot = False$$

Inference

- We want to start somewhere (KB)
- We'd like to apply some rules
- But there are lots of ways we might go
  - To reach some goal (sentence)
- Sound familiar?
- Inference as search:
  - Set of states
  - Start state
  - Set of actions and action rules
  - Goal test
  - Cost function

Inference rules

- Rules must be sound

- Modus ponens: $\frac{P \Rightarrow Q, P}{Q}$
- Modus tollens: $\frac{P \Rightarrow Q, \neg Q}{\neg P}$
- Simplification: $\frac{P \wedge Q}{P}$ or $\frac{P \wedge Q}{Q}$
- Resolution: $\frac{(P \vee C),\ (Q \vee \neg C)}{P \vee Q}$

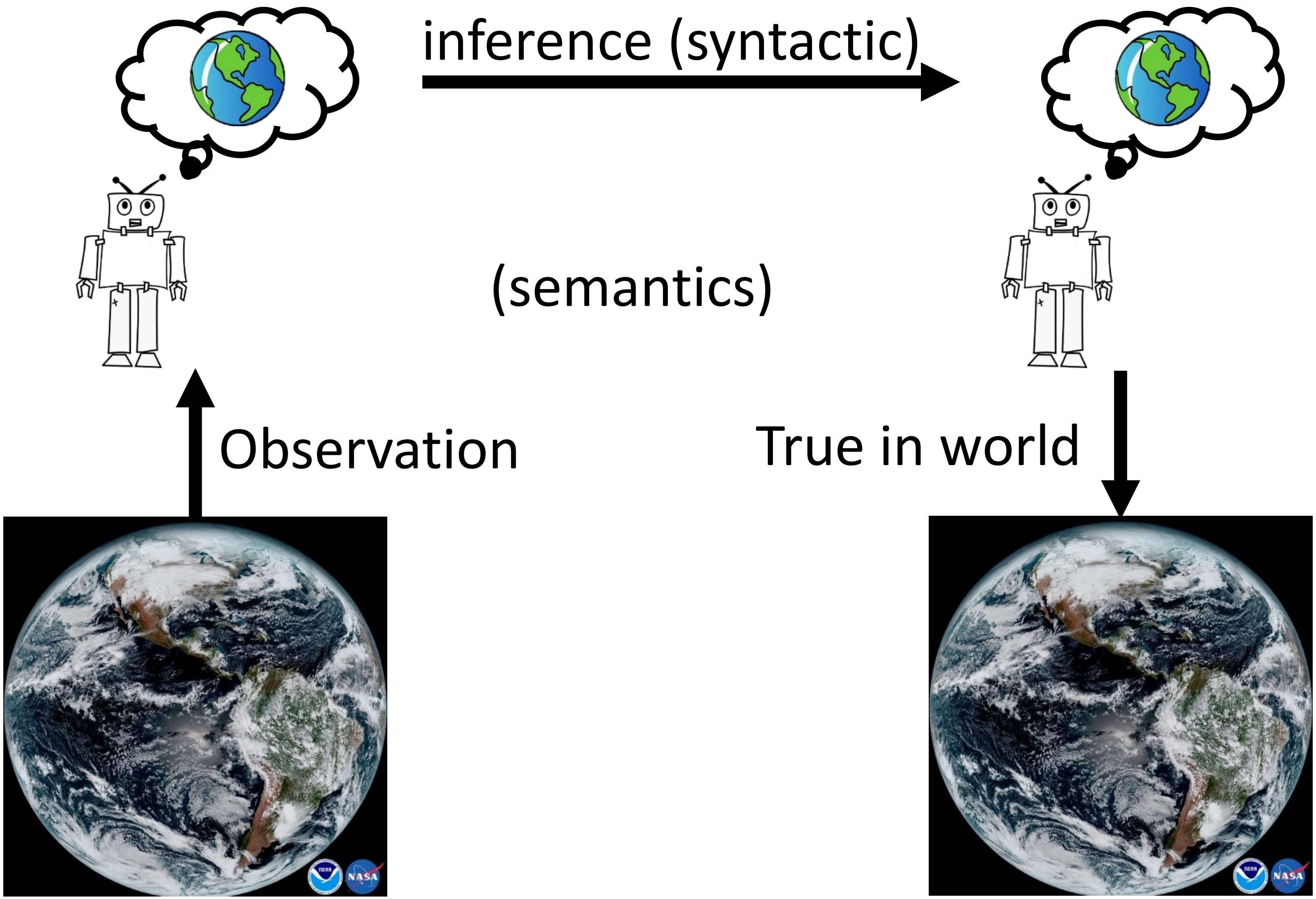
Resolution

- Resolution is sound and complete
  - And essentially all you need

$$\frac{a_1 \lor \cdots \lor a_{i-1} \lor c \lor a_{i+1} \lor \cdots a_n,\ \ b_1 \lor \cdots \lor b_{j-1} \lor \neg c \lor b_{j+1} \lor \cdots b_m}{a_1 \lor \cdots \lor a_{i-1} \lor a_{i+1} \lor \cdots a_n \lor b_1 \lor \cdots \lor b_{j-1} \lor b_{j+1} \lor \cdots b_m}$$

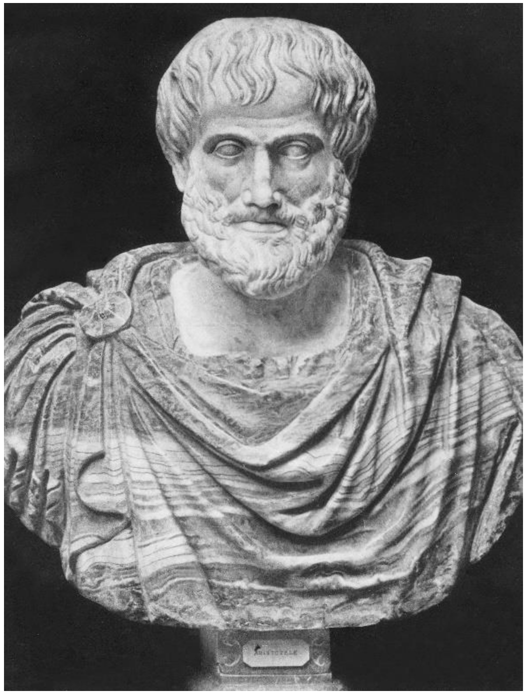
- Combine with a sound and complete search algorithm

The world and the model



Languages

- Propositional logic isn't very powerful
  - How to get more power?



First-order logic

- More sophisticated representation language
- World can be described by

$ColourOf(\cdot)$

$Adjacent(\cdot,\cdot)$
$IsApple(\cdot)$

objects functions predicates

First-order logic

- Objects:
  - A "thing in the world"
    - Apples
    - Green
    - The internet
    - AI class of 2024
    - ~~Twitter~~ $\mathbb{X}$
  - A name that references something
    - $MyApple271$
    - $TheInternet$

First-order logic

- Functions
  - Operators that map object(s) to single object
    - $ColourOf(\cdot)$
    - $ObjectNextTo(\cdot)$
    - $DateOfBirth(\cdot)$
    - $Spouse(\cdot)$

$$ColourOf(MyApple271) = Green$$

First-order logic

- Predicates – replaces proposition

- Like a function, but returns true or false

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

First-order logic

- Can build complex sentences using logical connectives
  - $Fruit(X) \Rightarrow Sweet(X)$
  - $Food(X) \Rightarrow (Savoury(X) \vee Sweet(X))$
  - $ParentOf(Bob, Alice) \wedge ParentOf(Alice, Eve)$
  - $Fruit(X) \Rightarrow Tasty(X) \vee (IsTomato(X) \wedge \neg Tasty(X))$
- Predicates can appear where a proposition appears in propositional logic, but functions cannot

Models for first-order logic

- Model in propositional logic
  - Set value of every variable in KB to true/false
  - 2^n models for n propositions

- More complex in FOL
- Model consists of
  - Set of objects
  - Set of functions and values for all inputs
  - Set of predicates and values for all inputs

Models for first-order logic

- Consider
  - Objects: $Orange, Apple$
  - Predicates: $IsGreen(\cdot), HasVitaminC(\cdot)$
  - Functions: $OppositeOf(\cdot)$

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

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

Knowledge bases in FOL

- KB is now:
  - Set of objects
  - Set of predicates
  - Set of functions
  - Set of sentences using predicates, functions, objects and asserted to be true

- Vocabulary: objects + predicates + functions

Knowledge bases in FOL

- Listing everything is tedious
  - Especially when general relationships hold

- Would like to say more general things that explicitly listing truth values for each object

Quantifiers

- Make generic statements that hold for entire collection of objects in KB

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

- Key idea: variable + binding rule

Existential quantifiers

- There exists objects such that a sentence holds

$$\exists x, isViceChancellor(x)$$

Universal quantifiers

- A sentence holds for all objects:

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

Quantifiers

- Difference in strength
- Universal quantifier is very strong
  - So use weak sentence

$$\forall x, Human(x) \Rightarrow Mortal(x)$$

- Existential quantifier is very weak
  - So use strong sentence

$$\exists Car(x) \land ParkedIn(x, E45)$$

Compound quantifiers

- Every person has a name

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

Splitting hairs

- The barber is the "one who shaves all those, and those only, who do not shave themselves"
  - $shaves(x, y)$: x shaves person y
  - $person(x)$: x is a person

$$\exists x, person(x) \wedge$$
$$(\forall y, person(y) \Rightarrow shaves(x, y) \Leftrightarrow \neg shaves(y\ , y))$$

- But now assign x to y. Gives
  - $shaves(x, x) \Leftrightarrow \neg shaves(x\ , x)$ which is false

Common mistakes

$$\forall x, Human(x) \Rightarrow Mortal(x)$$

vs

$$\forall x, Human(x), Mortal(x)$$

$$\exists x, Car(x) \wedge ParkedIn(x, E45)$$

vs

$$\exists x, Car(x) \Rightarrow ParkedIn(x, E45)$$

Inference in first-order logic

- Ground term or literal
  - An actual object: $MyApple271$
- A variable
  - Free placeholder: x
- If you have only ground terms, convert to propositional representation and proceed:

$$IsTasty(Apple271): IsTastyApple271$$

Instantiation

- Get rid of variables: instantiate to a literal

- Universally quantified
  - Write out each rule in KB with variables substituted
  - $\forall x, Fruit(x) \Rightarrow Tasty(x)$

$$Fruit(Apple) \Rightarrow Tasty(Apple)$$
$$Fruit(Orange) \Rightarrow Tasty(Orange)$$
$$Fruit(MyCar) \Rightarrow Tasty(MyCar)$$
$$Fruit(TheSky) \Rightarrow Tasty(TheSky)$$

Instantiation

- Existentially quantified
  - Invent new name (Skolem constant)

$$\exists x, Car(x) \wedge ParkedIn(E45)$$

$$Car(C) \wedge ParkedIn(C, E45)$$

  - Use unique name
  - Rule can then be discarded

PROLOG

- PROgramming in LOGic (Colmerauer, 1970s)
  - General-purpose AI programming language
  - Based on First-Order Logic
  - Declarative
- Use centred in Europe and Japan
  - Fifth-Generation Computer Project
  - Some parts of Watson (pattern matching over NLP)
  - Often used as component of a system