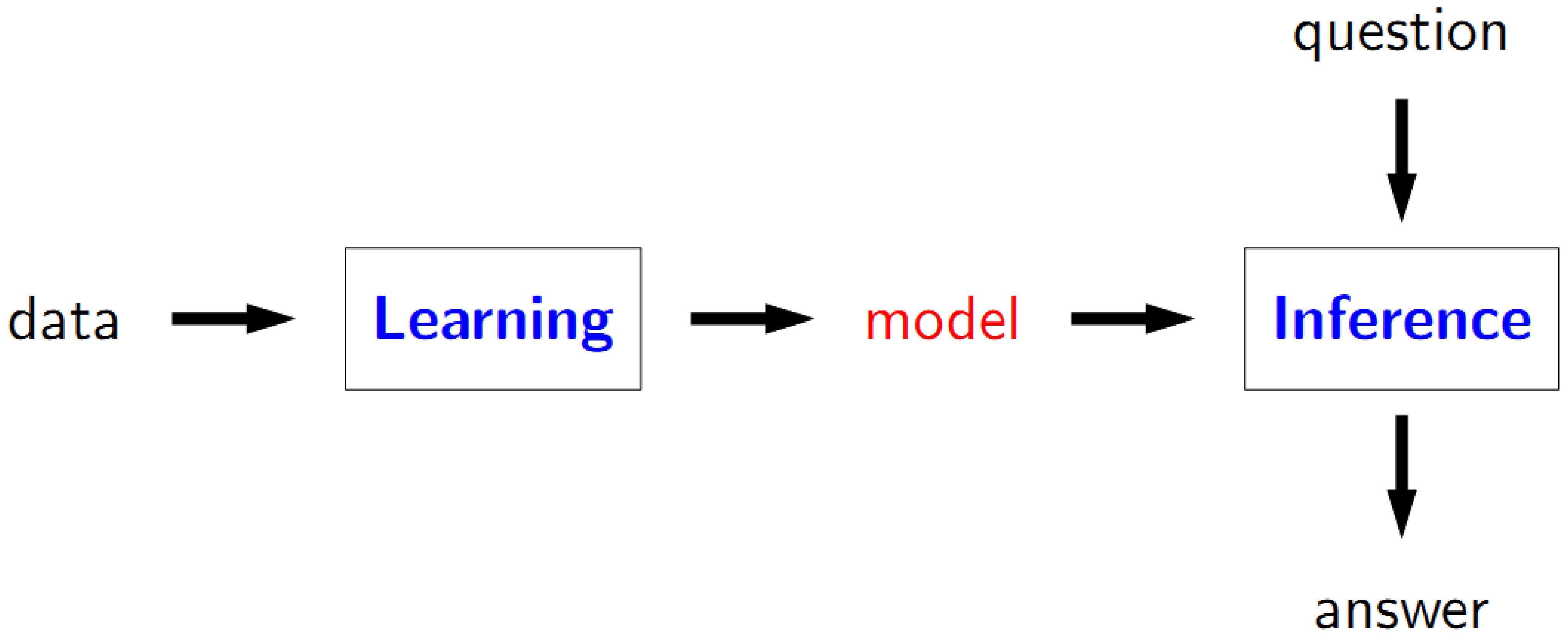
FIRST ORDER LOGIC

Dr. LEKEHALI Soumia

PLAN

- Overview
- FOL
  - Syntax
  - Semantic


question
data
Learning
model
Inference
answer

Modeling paradigms

Logic-based models: propositional logic, first-order logic

Applications: theorem proving, verification, reasoning

Think in terms of **logical formulas and inference rules**

State-based models: search problems, MDPs, games

Applications: route finding, game playing, etc.

Think in terms of **states, actions, and costs**

Variable-based models: CSPs, Bayesian networks

Applications: scheduling, tracking, medical diagnosis, etc.

Think in terms of **variables and factors**

- Logic was dominant paradigm in AI before 1990s

```
(ING/BY
   (PUSH NP/ T
        (SETR SUBJ *)
        (TO NP/VP)

           (* IF THE SUBJECT WAS NOT PROPERLY DETERMINED IN A
           POSS-ING COMPLEMENT, LOOK FOR IT HERE.)

))

(NP/
   (CAT DET T

         ((GETF POSSPRO                                    (* START OF THE NP
                                                           NETWORK.))
           (ADDL ADJS (BUILDQ (POSS (NP (PRO *)))))
           (SETRQ DET THE

          (* IF THE DETERMINER IS A POSSESSIVE PRONOUN
          (MY, YOUR), CONSTRUCT THE POSSESSIVE MODIFIER AND USE
          'THE' FOR THE DETERMINER)

))
         (T (SETR DET *)))
      (TO NP/ART))
   (CAT PRO T
      (SETR N (BUILDQ (PRO *))                             (* A PRONOUN MAY PICK UP
                                                           PP MODIFIERS IN NP/HEAD)
            )
      (SETR NU (GETF NUMBER))
      (TO NP/NP))
   (MEM (WHETHER IF)
      T
      (SETR NTYPE *)
      (TO COMPL/NTYPE

          (* CONSTRUCT THE COMPLEMENT STRUCTURE FOR SENTENCES
          SUCH AS 'I DON'T KNOW WHETHER HE LEFT.')
```

- Problem 1: deterministic, didn't handle **uncertainty** (probability a

- Problem 2: rule-based, didn't allow fine tuning from **data** (machi
this)

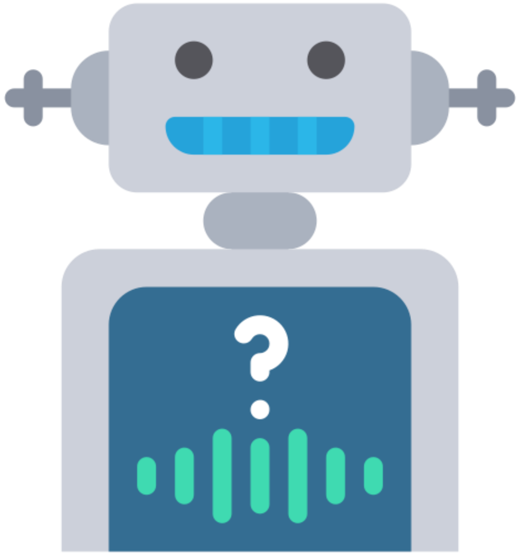
- Strength: provides **expressiveness** in a compact way

Question? If $X1 + X2 = 10$ and $X1 - X2 = 4$, what is X1?

- When we add $X1 + X2 = 10$ and $X1 - X2 = 4$, we get:
- $2X1 = 14$
- Simplifying the equation, we can see that X1 is equal to:
- $X1 = 7$

Smart personal assistant

ASK questions



TELL information

Use natural language

Demo python[nli.py]

>all students like 1cs.

>>>>> I learned something.

>Alice does not like 1cs.

>>>>> I learned something.

>is Alice student

>>>>> no

Natural language

Example:

- A dime is better than a nickel.
- A nickel is better than a penny.
- Therefore, a dime is better than a penny.

Example:

- A penny is better than nothing.
- Nothing is better than world peace.
- Therefore, a penny is better than world peace??? another example similar to this

Language

Language is a mechanism for expression.

- **Natural languages (informal):**

  English: Two divides even numbers.

  French : Deux divise les nombres pairs.

- **Programming languages (formal):**

  Java : public static boolean even(int x)

  { return x % 2 == 0; }

  Python: def even(x): return x % 2 == 0

Logical languages (formal):

- **First-order-logic**: $\forall x.Even(x) \rightarrow Divides(x, 2)$

Two goals of a logic language

- Represent knowledge about the world
- Reason with that knowledge

In which we design agents that can form representations of a complex world, use a process of inference to derive new representations about the world, and use these new representations to deduce what to do.

Ingredients of logic

Syntax

knowledge bases consist of sentences which are expressed according to the syntax of the representation language

"x + y =4" is a well-formed sentence (Formulas)

whereas **"x4y+="** is not.

Semantics (meaning of sentences)

for each formula, specify a set of models (assignments / configurations of the world)

"**x + y =4"** is true in a world where **x** is 2 and y is **2**, but false in a world where **x** is 1 and **y** is 1

Inference rules: given f, what new formulas g can be added that are guaranteed to follow $(\frac{f}{g})$?

Syntax versus semantics

Syntax: what are valid expressions in the language?

Semantics: what do these expressions mean?

Different syntax, same semantics (5):

$$2 + 3 \quad \Leftrightarrow \quad 3 + 2$$

Same syntax, different semantics (1 versus 1.5):

$$3 \;/\; 2 \text{ (Python 2.7)} \quad \not\Leftrightarrow \quad 3 \;/\; 2 \text{ (Python 3)}$$

- Any ‘formal system’ can be considered a logic if it has:
- – a well-defined syntax;
- – a well-defined semantics; and
- – a well-defined proof-theory.

Logics

- Propositional logic with only Horn clauses
- Propositional logic
- Modal logic
- First-order logic with only Horn clauses
- **First-order logic**
- Second-order logic

Logics

- Propositional logic with only Horn clauses
- Propositional logic
- Modal logic
- First-order logic with only Horn clauses
- **First-order logic**
- Second-order logic

Limitations of propositional logic

- ***Limited expressive power***:

PL can only represent simple propositional statements using logical connectives like **AND, OR, NOT, and IMPLIES**.

Example:

PL can represent "**It is raining**" using the proposition "**p**", and "It is windy" using the proposition "q",

but it cannot represent more complex statements like

"**If it rains, then the streets will be wet**",

which requires FOL's conditional connective.

Limitations of propositional logic

- **Lack of variables**:

PL does not have variables, which means that it cannot express statements that generalize over a range of objects or situations.

Example :PL cannot represent the statement **"All dogs bark",** which requires the use of a variable to stand for the range of dogs.

Limitations of propositional logic

- **No distinction between objects**:

PL treats all propositions as equal, and there is no way to distinguish between different objects or individuals.

Example: PL cannot distinguish between "John is taller than Mary" and "**Mary is taller than John**", which requires FOL's use of variables and relations.

Limitations of propositional logic

- **No notion of scope:**

PL does not have a notion of scope, which means that it cannot express statements like "there exists an x such that..." or "for all x...".

Example: PL cannot express the statement **"There is a person who speaks French"**, which requires the use of a **quantifier** in FOL.

Limitations of propositional logic

- **Not suitable for complex reasoning**:
- While PL is useful for simple, deductive reasoning, it is not well-suited for more complex forms of reasoning.

Example: PL cannot represent the statement "**If it is raining, and John forgot his umbrella, then John will get wet**", which requires FOL's use of variables, relations, and conditionals.

All students know arithmetic.

$$\forall x\, \mathsf{Student}(x) \rightarrow \mathsf{Knows}(x, \mathsf{arithmetic})$$

Propositional logic syntax

Propositional symbols (atomic formulas): A, B, C

Logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

Build up formulas recursively—if f and g are formulas, so are the following:

- Negation: $\neg f$
- Conjunction: $f \wedge g$
- Disjunction: $f \vee g$
- Implication: $f \rightarrow g$
- Biconditional: $f \leftrightarrow g$

FOL syntax

Terms (refer to objects):

- Constant symbol (e.g., arithmetic)
- Variable (e.g., x)
- Function of terms (e.g., $\text{Sum}(3, x)$)

Other terms : A, 123, x, f(A) , f(g(x)) , + (x,1)

Formulas (refer to truth values):

- Atomic formulas (atoms): predicate applied to terms (e.g., $\text{Knows}(x, \text{arithmetic})$)
- Connectives applied to formulas (e.g., $\text{Student}(x) \to \text{Knows}(x, \text{arithmetic})$)
- Quantifiers applied to formulas (e.g., $\forall x\, \text{Student}(x) \to \text{Knows}(x, \text{arithmetic})$)

FOL syntax

Terms (refer to objects):

By convention, we will write constant terms in capital letters , and variabls in Lower case .

- Atomic formulas (atoms): predicate applied to terms (e.g., Knows(x, arithmetic))
- Connectives applied to formulas (e.g., Student(x) $\rightarrow$ Knows(x, arithmetic))
- Quantifiers applied to formulas (e.g., $\forall x$ Student(x) $\rightarrow$ Knows(x, arithmetic))

FOL syntax

Quantifiers

Universal quantification ($\forall$):

Think conjunction: $\forall x\, P(x)$ is like $P(A) \wedge P(B) \wedge \cdots$

Existential quantification ($\exists$):

Think disjunction: $\exists x\, P(x)$ is like $P(A) \vee P(B) \vee \cdots$

Some properties:

- $\neg\forall x\, P(x)$ equivalent to $\exists x\, \neg P(x)$
- $\forall x\, \exists y\, \mathrm{Knows}(x, y)$ different from $\exists y\, \forall x\, \mathrm{Knows}(x, y)$

FOL syntax

A variable not bound by a quantifier in a formula is called **free**. The formula:

$$\underline{\exists y} Q(x, \underline{y})$$

contains two variables, one free (x) and one bound by a quantifier (y).

A **sentence**, or a **closed** formula is a formula without free variables.

FOL syntax

Natural language quantifiers

Universal quantification ($\forall$):

Every student knows arithmetic.

$\forall x \, \mathsf{Student}(x) \rightarrow \mathsf{Knows}(x, \mathsf{arithmetic})$

Existential quantification ($\exists$):

Some student knows arithmetic.

$\exists x \, \mathsf{Student}(x) \wedge \mathsf{Knows}(x, \mathsf{arithmetic})$

Note the different connectives!

Examples

There is some course that every student has taken.

$$\exists y \, \mathsf{Course}(y) \wedge [\forall x \, \mathsf{Student}(x) \rightarrow \mathsf{Takes}(x, y)]$$

Every even integer greater than 2 is the sum of two primes.

$$\forall x \, \mathsf{EvenInt}(x) \wedge \mathsf{Greater}(x, 2) \rightarrow \exists y \, \exists z \, \mathsf{Equals}(x, \mathsf{Sum}(y, z)) \wedge \mathsf{Prime}(y) \wedge \mathsf{Prime}(z)$$

If a student takes a course and the course covers a concept, then the student knows that concept.

$$\forall x \, \forall y \, \forall z \, (\mathsf{Student}(x) \wedge \mathsf{Takes}(x, y) \wedge \mathsf{Course}(y) \wedge \mathsf{Covers}(y, z)) \rightarrow \mathsf{Knows}(x, z)$$

Propositional logic semantics

Definition: model

A **model** w in propositional logic is an **assignment** of truth values to propositional symbols.

Definition: interpretation function

Let f be a formula.

Let w be a model.

An **interpretation function** $\mathcal{I}(f, w)$ returns:

- true (1) (say that w satisfies f)
- false (0) (say that w does not satisfy f)

Example: interpretation function

Formula: $f = (\neg A \land B) \leftrightarrow C$

Model: $w = \{A : 1, B : 1, C : 0\}$

Interpretation:

$\mathcal{I}((\neg A \land B) \leftrightarrow C, w) = 1$

- $\mathcal{I}(\neg A \land B, w) = 0$
  - $\mathcal{I}(\neg A, w) = 0$
    - $\mathcal{I}(A, w) = 1$
  - $\mathcal{I}(B, w) = 1$
- $\mathcal{I}(C, w) = 0$

Models in FOL

Recall a model represents a possible situation in the world.

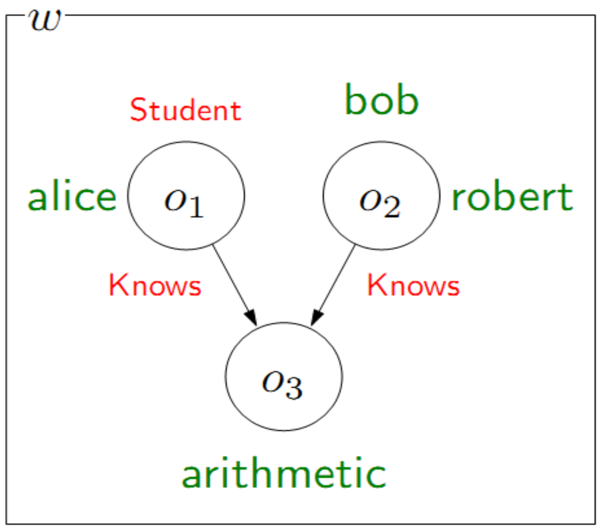
Propositional logic: Model w maps propositional symbols to truth values.

$$w = \{\text{AliceKnowsArithmetic} : 1, \text{BobKnowsArithmetic} : 0\}$$

First Order Logic ?

Graph representation of a model

If only have unary and binary predicates, a model w can be represented as a directed graph:



- Nodes are objects, labeled with constant symbols
- Directed edges are binary predicates, labeled with predicate symbols; unary predicates are additional node labels

Models in FOL

A model w in first-order logic maps:

- constant symbols to objects

$$w(\text{alice}) = o_1, w(\text{bob}) = o_2, w(\text{arithmetic}) = o_3$$

- predicate symbols to tuples of objects

$$w(\text{Knows}) = \{(o_1, o_3), (o_2, o_3), \dots\}$$

……………………