Artificial Intelligence

CSE 4205/3201

First-Order Logic

First-Order Logic (Predicate Logic)

- Much more powerful than propositional (Boolean) logic
  - Greater expressive power than propositional logic
    - We no longer need a separate rule for each similar sentence
      - Peter is a man → P
      - Paul is a man → Q
      - John is a man → R
    - In PL, can not draw any conclusions about similarities between P, Q, R.

- Sentences like "All men are mortal" can not be represented in PL.

- In Predicate Logic (logical extension of PL) such sentences can be easily represented.
    - MAN(Peter)
    - MAN(Paul)
    - MAN(John)
  - The limitations of propositional logic are removed to some extent.
  - Similarity between these facts that they are all man can be easily derived.

First-Order Logic (Predicate Logic)

Representing simple facts

It is raining
RAINING

It is sunny
SUNNY

It is windy
WINDY

If it is raining, then it is not sunny
RAINING $\rightarrow \neg$SUNNY

First-Order Logic (Predicate Logic)

- First-Order Logic assumes that the world contains:
  - Objects
    - E.g. people, houses, numbers, theories, colors, football games, …
  - Relations
    - E.g. red, round, prime, part of, has color, owns, comes between, …
  - Functions
    - E.g. father of, best friend, third quarter of, one more than, beginning of, …

Syntax of First-Order Logic

• Constants	KingJohn, 2, …
• Predicates	Brother, >, …
• Functions	Sqrt, LeftArmOf, …
• Variables	x, y, a, b, …
• Connectives	$\wedge \vee \neg \Rightarrow \Leftrightarrow$
• Equality	=
• Quantifiers	$\exists \; \forall$

Components of First-Order Logic

- Term
  - Constant, e.g. Red
  - Function of constant, e.g. Color(Block1)

- Atomic Sentence
  - Predicate relating objects (no variable)
    - Brother (John, Richard)
    - Married (Mother(John), Father(John))

- Complex Sentences
  - Atomic sentences + logical connectives
    - Brother (John, Richard) $\land \neg$ Brother (John, Father(John))

Components of First-Order Logic

- Quantifiers
  - Each quantifier defines a variable for the duration of the following expression, and indicates the truth of the expression…
  - Universal quantifier “for all” $\forall$
    - The expression is true for every possible value of the variable
  - Existential quantifier “there exists” $\exists$
    - The expression is true for at least one value of the variable

Truth in First-Order Logic

- Sentences are true with respect to a model and an interpretation
- *"x loves y"* is represented as *LOVE(x, y)* which maps it to true or false when *x* and *y* get instantiated to actual values.
- *"john's father loves john"* is represented as *LOVE(father(john), john)*.
  - Here *father* is a function that maps *john* to his father.
- *x is greater than y* is represented in predicate calculus as *GT(x, y)*.
- It is defined as follows:

$$GT(x, y) \quad = \quad True\text{ , if } x > y$$
$$\quad = \quad False\text{ , otherwise}$$

- Symbols like *GT* and *LOVE* are called predicates
  - Predicates two terms and map to T or F depending upon the values of their terms.

Truth in First-Order Logic

- Translate the sentence *"Every man is mortal"* into Predicate formula.
  - Representation of statement in predicate form
    - “x is a man” and “MAN(x),
    - x is mortal” by MORTAL(x)
  - Every man is mortal :

    $(\forall x)\ (MAN(x) \rightarrow MORTAL(x))$

Universal Quantification

- $\forall$ <variables> <sentence>
- $\forall x$ P is true in a model m iff P with x being each possible object in the model
  - Every man is mortal :

    $(\forall x)\ (MAN(x) \rightarrow MORTAL(x))$
- Equivalent to the conjunction of instantiations of P
  - $At(Mike, KSU) \Rightarrow Smart(Mike) \wedge$
  - $At(Laurie, KSU) \Rightarrow Smart(Laurie) \wedge$
  - $At(Sarah, KSU) \Rightarrow Smart(Sarah) \wedge$
  - …
- Every gardener likes the sun.
  -- $\forall x\ gardener(x) \rightarrow likes(x,Sun)$
- All purple mushrooms are poisonous.
  -- $\forall x\ (mushroom(x) \wedge purple(x)) \rightarrow poisonous(x)$

A Common Mistake to Avoid

- Typically $\Rightarrow$ is the main connective with $\forall$
- Common mistake: using $\wedge$ as the main connective with $\forall$
- $\forall x\ At(x, KSU) \wedge Smart(x)$

Existential Quantification

- $\exists$ <variables> <sentence>
- $\exists$ x P is true in a model m iff P with x being at least one possible object in the model
  - Someone likes McDonalds

    $\exists$x, likes(x, McDonalds)

- Equivalent to the disjunction of instantiations of P
  - At(Mike, KSU) $\wedge$ Smart(Mike) $\vee$
  - At(Laurie, KSU) $\wedge$ Smart(Laurie) $\vee$
  - At(Sarah, KSU) $\wedge$ Smart(Sarah) $\vee$
  - …

Existential Quantification

Note on ‘*any’ (ex: anyone)*:

- In ordinary english, ‘any’ can sometimes mean "some" and sometimes mean "all". Sometimes *anyone* corresponds to $\exists$ and sometimes to $\forall$; you have to think about the meaning of the whole sentence. Many papers have been written exploring the issue of how best to account for the distribution of meanings of *any.*

- *‘Any’* in different cases can be viewed either as a wide-scope universal (with scope over the *if*-clause) or as a narrow-scope existential (with scope inside the *if*-clause). The fact that these are equivalent, is part of the source of debates about *any.*

Another Common Mistake to Avoid

- Typically, $\wedge$ is the main connective with $\exists$
- Common mistake: using $\Rightarrow$ as the main connective with $\exists$
- $\exists$ x At(x, KSU) $\Rightarrow$ Smart(x)

Examples

- Everyone likes McDonalds
  - $\forall x,\ likes(x, McDonalds)$
- Someone likes McDonalds
  - $\exists x,\ likes(x, McDonalds)$
- All children like McDonalds
  - $\forall x,\ child(x) \Rightarrow likes(x, McDonalds)$
- Everyone likes McDonalds unless they are allergic to it
  - $\forall x, \neg allergic\ (x, McDonalds) \Rightarrow likes(x, McDonalds)$

Properties of Quantifier

- $\forall x\ \forall y$ is the same as $\forall y\ \forall x$
- $\exists x\ \exists y$ is the same as $\exists y\ \exists x$
- $\exists x\ \forall y$ is not the same as $\forall y\ \exists x$
  - $\exists x\ \forall y$ Loves(x, y)
    - “There is a person who loves everyone in the world”
  - $\forall y\ \exists x$ Loves(x, y)
    - “Everyone in the world is loved by at least one person”

Nesting Quantifier

- Someone likes all kinds of food

  $\exists y\ \forall x,\ \ food(x) \wedge likes(y, x)$

- Every food has someone who likes it

  $\forall x\ \exists y,\ \ food(x) \wedge likes(y, x)$

Examples

- Quantifier Duality
  - Not everyone like McDonalds

    $\neg(\forall x,\ \ likes(x, McDonalds))$

    $\exists x, \neg likes(x, McDonalds)$

  - No one likes McDonalds

    $\neg(\exists x,\ \ likes(x, McDonalds))$

    $\forall x, \neg likes(x, McDonalds)$

Fun with Sentence

- Brothers are siblings

  $\forall x,y \ \ Brother(x,y) \Rightarrow Sibling(x, y)$

- Sibling is "symmetric"

  $\forall x,y \ \ Sibling(x,y) \Leftrightarrow Sibling(y, x)$

- One's mother is one's female parent

  $\forall x,y \ \ Mother(x,y) \Leftrightarrow (Female(x) \wedge Parent(x,y))$

Other Comments About Quantification

- To say "everyone likes McDonalds", the following is too broad!
  - $\forall x$, likes(x, McDonalds)

- We mean: Every one (who is a human) likes McDonalds
  - $\forall x$, person(x) $\Rightarrow$ likes(x, McDonalds)

- Essentially, the left side of the rule declares the class of the variable x

- Constraints like this are often called "domain constraints"

Using Predicate Logic

1. Marcus was a man.
2. Marcus was a Pompeian.
3. All Pompeians were Romans.
4. Caesar was a ruler.
5. All Pompeians were either loyal to Caesar or hated him.
6. Every one is loyal to someone.
7. People only try to assassinate rulers they are not loyal to.
8. Marcus tried to assassinate Caesar.

Using Predicate Logic

1. Marcus was a man.

   man(Marcus)

Using Predicate Logic

2. Marcus was a Pompeian.

   Pompeian(Marcus)

Using Predicate Logic

3. All Pompeians were Romans.

   $\forall x$: Pompeian(x) $\rightarrow$ Roman(x)

Using Predicate Logic

4. Caesar was a ruler.

   ruler(Caesar)

Using Predicate Logic

5. All Pompeians were either loyal to Caesar or hated him.

   inclusive-or

   $\forall x$: Pompeian (x) $\rightarrow$ loyalto(x, Caesar) $\vee$ hate(x, Caesar)

   exclusive-or

   $\forall x$: Pompeian (x) $\rightarrow$ (loyalto(x, Caesar) $\wedge$ $\neg$hate(x, Caesar)) $\vee$

   ($\neg$loyalto(x, Caesar) $\wedge$ hate(x, Caesar))

Using Predicate Logic

6. Every one is loyal to someone.

   $\forall x: \exists y: loyalto(x, y)$ $\exists y: \forall x: loyalto(x, y)$

Using Predicate Logic

7. People only try to assassinate rulers they are not loyal to.

$$\forall x: \forall y: person(x) \wedge ruler(y) \wedge tryassassinate(x, y)$$

$$\rightarrow \neg loyalto(x, y)$$

Using Predicate Logic

8. Marcus tried to assassinate Caesar.

   tryassassinate(Marcus, Caesar)

Using Predicate Logic

Was Marcus loyal to Caesar?

man(Marcus)

ruler(Caesar)

tryassassinate(Marcus, Caesar)

$\Downarrow$ $\forall x$: man(x) $\rightarrow$ person(x)

$\neg$loyalto(Marcus, Caesar)

Using Predicate Logic

- Many English sentences are ambiguous.
- There is often a choice of how to represent knowledge.
- Obvious information may be necessary for reasoning
- We may not know in advance which statements to deduce (P or $\neg$P).