Predicate logic

18 November 2024 15:58

Propositional logic allows us to state facts

- does not allow stating **properties of** and **relations between** "objects"
- e.g., the property of numbers of being even, or odd

This brings us to a **richer** logic called **predicate logic**

- **contains** propositional logic
- also known as **first-order logic**
- Predicate logic allows us to reason about members of a (non-empty) domain

Propositional logic allows us to state facts

- does not allow stating **properties of** and **relations between** "objects"
- e.g., the property of numbers of being even, or odd

This brings us to a **richer** logic called **predicate logic**

- **contains** propositional logic
- also known as **first-order logic**
- Predicate logic allows us to reason about members of a (non-empty) domain

- **Constant**: s which stands for Socrates

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

- Symbols to represent (as yet unknown) objects in the domain
- Usually denoted by $x, y, z, \dots$
- Similar to variables from programming languages

Domain (also called universe)

- Non-empty set of objects/entities (individuals) to reason about
- Example: set of 1st year students

Variables

- Symbols to represent (as yet unknown) objects in the domain
- Usually denoted by $x, y, z, \ldots$
- Similar to variables from programming languages

Quantifiers

- **universal** quantifier
  $\forall x. \cdots$: "for all elements x of the domain"
- **existential** quantifier
  $\exists x. \cdots$: "there exists an element x of the domain such that"
- quantify over elements of the domain
- **precedence**: lower than the other connectives

11/

Functions

- Build an element of the domain from elements of the domain
- Usually denoted by $f, g, h, \ldots$
- Different functions can have different numbers of arguments
- The number of arguments of a function is called its **arity**
- A function symbol of arity 1 can only be applied to 1 argument,
  A function symbol of arity 2 can only be applied to 2 arguments, etc.
- **Notation**: We sometimes write f^k when we want to indicate that the function symbol f has arity k

Constants

- Specific objects in the domain
- Functions of arity 0
- Usually denoted by $a, b, c, \ldots$

Predicates

- Propositions are facts/statements, which may be true or false
- A predicate evaluates to true/false depending on its arguments
- Predicates can be seen as functions from elements of the domain to propositions
- **Example**: $p(x)$ means "predicate p is true for variable x"
- **Example**: $p(a)$ means "predicate p is true for constant a"

Examples of formulas in predicate logic

- $\forall x.(p(x) \wedge q(x))$
  - for all x it is true that $p(x)$ and $q(x)$
- $(\forall x.p(x)) \rightarrow \neg\forall x.q(x)$
  - if $p(x)$ is true for all x, then $q(x)$ is not true for all x
- $\exists x.(p(x) \vee \neg q(x))$
  - there is some x for which $p(x)$ is true or $q(x)$ is not true

Famous derivation in logic:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

Can we express this in propositional logic?

includes the following components:

- Domain = Men
- Socrates is one member of this domain
- Predicates are "being a man" and "being mortal"

Another example: consider a database with 3 tables

Student	
sid	**name**
0	Alice
1	Bob

Module	
mid	**name**
0	Math
1	OOP

Enroll	
sid	**mid**
0	0
1	1

These 3 tables can be seen as 3 relations:

Another example: consider a database with 3 tables

Student	
sid	**name**
0	Alice
1	Bob

Module	
mid	**name**
0	Math
1	OOP

Enroll	
sid	**mid**
0	0
1	1

These 3 tables can be seen as 3 relations:

- $Student(sid, name)$: predicate $Student$ relates student ids and names
- $Module(mid, name)$: predicate $Module$ relates module ids and names
- $Enroll(sid, mid)$: predicate $Enroll$ relates student and module ids

Domain = all possible values

A formula can be seen as a query

For example: find the Students x enrolled in the Math module

- $\exists y.\exists z.Student(y, x) \land Module(z, \text{Math}) \land Enroll(y, z)$

More examples in predicate calculus

Domain is cars, and we have 3 predicate symbols:
• $f(x)$ = "x is fast" • $r(x)$ = "x is red" • $p(x)$ = "x is purple"

How to express the following sentences in predicate logic?

- All cars are fast: $\forall x.f(x)$
- All red cars are fast: $\forall x.r(x) \rightarrow f(x)$
  - "any car that is red must be fast"
  - might not be a red car, but if a car is red it is fast
- Some red cars are fast: $\exists x.r(x) \land f(x)$
  - "there is at least one car that is red and fast"
  - Wrong answer: $\exists x.r(x) \rightarrow f(x)$ (true if at least 1 non-red car!)
- There are no red cars: $\neg\exists x.r(x)$
  - Alternative answer: $\forall x.\neg r(x)$ ("all cars must not be red")
- No fast cars are purple: $\neg\exists x.f(x) \land p(x)$
  - "there is no car that is both fast and purple"
  - Alternative: $\forall x.f(x) \rightarrow \neg p(x)$ ("all fast cars are not purple")

Examples

Consider the following domain and signature:

- Domain: $\mathbb{N}$
- Functions: $0, 1, 2, \ldots$ (arity 0); $+$ (arity 2)

Examples

Consider the following domain and signature:

- Domain: $\mathbb{N}$
- Functions: $0, 1, 2, \ldots$ (arity 0); $+$ (arity 2)
- Predicates: $\texttt{prime}$, $\texttt{even}$, $\texttt{odd}$ (arity 1); $=$, $>$, $\geqslant$ (arity 2)

Express the following sentences in predicate logic

- All prime numbers are either 2 or odd.
  $\forall x.\texttt{prime}(x) \rightarrow x = 2 \vee \texttt{odd}(x)$
- Every even number is equal to the sum of two primes.
  $\forall x.\texttt{even}(x) \rightarrow \exists y.\exists z.\texttt{prime}(y) \wedge \texttt{prime}(z) \wedge x = y + z$
- There is no number greater than all numbers.
  $\neg\exists x.\forall y.x > y$
- All numbers have a number greater than them.
  $\forall x.\exists y.y > x$

Let the domain be $\mathbb{N}$.

Provide examples of function symbols, along with their arities

- $0, 1, 2, \ldots$ are constant symbols (nullary function symbols)
- $\texttt{add}$: the binary addition function
- $\texttt{add}(m, n)$: addition applied to the two expressions m and n
- $\texttt{square}$: the unary square function
- $\texttt{square}(m)$: square applied to the expression m

One more example (from the book – section 7.6.2)

Domain is people, and we have 6 predicates

$\mathsf{politician}(x)$ $\mathsf{rich}(x)$ $\mathsf{crazy}(x)$ $\mathsf{trusts}(x, y)$ $\mathsf{knows}(x, y)$ $\mathsf{related\text{-}to}(x, y)$

Express the following sentences in predicate logic

- Nobody trusts a politician

Express the following sentences in predicate logic

- Nobody trusts a politician.
  $\neg\exists x.\exists y.\text{politician}(y) \wedge \text{trusts}(x, y)$
- Anyone who trusts a politician is crazy.
  $\forall x.(\exists y.\text{politician}(y) \wedge \text{trusts}(x, y)) \rightarrow \text{crazy}(x)$
- Everyone knows someone who is related to a politician.
  $\forall x.\exists y.\text{knows}(x, y) \wedge \exists z.\text{politician}(z) \wedge \text{related-to}(y, z)$
- Everyone who is rich is either a politician or knows a politician.
  $\forall x.\text{rich}(x) \rightarrow \text{politician}(x) \vee \exists y.\text{knows}(x, y) \wedge \text{politician}(y)$

Natural Deduction rules for ∀ and ∃?

Propositional logic: Each connective has two inference rules

- One for introduction
- One for elimination

Introduction and elimination rules for ∀ and ∃?

$$\frac{?}{\forall x.P}\,[\forall I] \qquad \frac{\forall y.P}{?}\,[\forall E]$$

$$\frac{?}{\exists x.P}\,[\exists I] \qquad \frac{\exists y.P}{?}\,[\exists E]$$

Arity of predicates

The **arity** of a predicate is the number of arguments it takes

Unary predicates (arity 1) represent facts about individuals

- $p(x)$ = "x is prime"

Binary predicates (arity 2) represent relationships between individuals, i.e., they represent relations

- Example: $m(a, b)$ = "a is married to b"
- Doesn't have to be symmetric!
- Example: $l(a, b)$ = "a likes b"

What are **nullary** predicates (arity 0)?

- Atomic propositions!

Notation: We sometimes write p^k when we want to indicate that

- Atomic propositions!

Notation: We sometimes write p^k when we want to indicate that the predicate symbol p has arity k

The syntax of predicate logic is defined by the following grammar:

$$
\begin{array}{lcl}
t & ::= & x \mid f(t, \ldots, t) \\
P & ::= & p(t, \ldots, t) \mid \neg P \mid P \wedge P \mid P \vee P \mid P \rightarrow P \mid \forall x.P \mid \exists x.P
\end{array}
$$

where:

- x ranges over variables
- f ranges over function symbols
- $f(t_1, \ldots, t_n)$ is a well-formed term only if f has arity n
- p ranges over predicate symbols
- $p(t_1, \ldots, t_n)$ is a well-formed formula only if p has arity n

The pair of a collection of function symbols, and a collection of predicate symbols, along with their arities, is called a **signature**.

The scope of a quantifier extends as far right as possible.
E.g., $P \wedge \forall x.p(x) \vee q(x)$ is read as $P \wedge \forall x.(p(x) \vee q(x))$