4CCS1ELA – Elementary Logic with Applications

4 – Introduction to Predicate Logic

Odinaldo Rodrigues

odinaldo.rodrigues@kcl.ac.uk

Room S1.26, +44 (0)20 7848 2087

Department of Informatics

King's College London

Outline

INTRODUCTION

Motivation

The type of arguments representable in propositional logic is somewhat limited because a proposition is an indivisible statement and there is no way to directly refer to the objects mentioned by it.

Consider the statements:

- P: Blue cars are fast.
- Q: My car is blue.
- R: My car is fast.

Ideally, we would like our system to be able to conclude that the last statement is a logical consequence of the first two.

Motivation

However, propositional logic does not allow us to derive R from P and Q. In other words,

$$P, Q \not\models R$$

(take, for instance, the interpretation v such that $v(P) = v(Q) = 1$ and $v(R) = 0$).

This limitation is because the atomic propositions P, Q and R cannot be decomposed further to allow us to refer to the properties of the objects they refer to.

Using a more fine-grained logic

We need a logic that can look deeper into the structure of the statements and that is capable of representing relationships about individuals of a particular domain, e.g., the collection of all cars.

In such a logic, the atomic components would be used to express properties or relationships between objects of the domain.

Predicate logic

Thus,

- A statement such as P would refer to *all* elements of the domain
- A statement such as Q would refer to a particular element of the domain (i.e., my car)

Provided we also had a mechanism to instantiate general statements (such as P) to a particular object ("my car", in this case), then we would be able to conclude that R follows from P and Q.

A logic of this type of is commonly known as predicate logic and sometimes also called first-order logic.

Re-examining our example

Let us assume the domain of all cars.

The statement '*blue cars are fast*', can be re-worked in the following way (where x is a variable):

- For all elements x of the domain, it follows that if x is a blue car, then x is fast.
- For all x (x is a blue car $\rightarrow$ x is fast)
- Let $\forall x$ stand for "For all x"; $Blue(x)$ mean "x is blue" and $Fast(x)$ mean "x is fast", we then get the formula

$$\forall x(Blue(x) \rightarrow Fast(x))$$

Understanding a predicate logic formula

In the formula $\forall x(Blue(x) \rightarrow Fast(x))$ ('Blue cars are fast'), *Blue* and *Fast* are one-place (or unary) **predicate symbols**, designating objects that are blue and fast, respectively.

$Blue(x)$ and $Fast(x)$, will be 'true' depending on whether the object referred to by the variable x has the corresponding property.

The $\forall$ symbol means that x runs through every element of the domain. The formula which it refers to is therefore said to be "universally quantified".

In this example, we say that the occurrences of the variable x in the formula are **bound** by the quantifier $\forall$.

QUANTIFIERS

Universal quantifier, example 2

Now, suppose we would like to represent the statement:

'Any integer greater than 5 *is also greater than* 3*'*

If our domain contains numbers in general, we would need two predicate symbols: one to designate the integer numbers and the other to represent the fact that one number is greater than another.

If our domain only contains integers, then we do not need the first predicate, so the definition of the domain plays an important role in the representation process.

Representing the statement in predicate logic

- Let the predicate *Int* be used to designate that a number is integer and the predicate $>(x, y)$ be used to indicate that the integer in the variable x is greater than the integer in the variable y
- We could 'name' the number **5** as the constant symbol 5 and the number **3** as the constant symbol 3
- So we want to say that "for all x, if x is an integer and x is greater than 5, then x is greater than 3", which is represented as:

$$\forall x\big((Int(x) \wedge >(x,5)) \rightarrow >(x,3)\big)$$

Understanding a predicate logic formula (contd.)

Now let us de-construct the previous formula in stages.

$$\forall x\big((Int(x) \wedge >(x,5)) \rightarrow >(x,3)\big)$$

We say that:

- 5 and 3 are individual *constants* representing the numbers **5** and **3**
- *Int* is a unary predicate symbol denoting the property that the object referred to by it is integer (*Int* is a unary relation!)
- $>$ is a binary predicate symbol such that $>(x,y)$ denotes that 'x is greater than y' ($>$ is a binary relation!)

Understanding a predicate logic formula (contd.)

For convenience, we may choose the more familiar *infix* form of the predicate $>$ and simply write $x > y$ instead of $>(x,y)$. This would give us

$$\forall x\big((Int(x) \wedge x > 5) \rightarrow x > 3\big)$$

which is easier to read.

Existential quantifier

Sometimes, we want to specify a special element of the domain without explicitly naming it. This is useful to represent statements such as “there is a smallest natural number”.

Whereas the universal quantifier applies to all elements of the domain, statements such as the one above refer to a particular element.

For example, suppose we want to represent the statement ‘Somebody likes somebody’ (we may not know who likes whom). This is where the “existential” quantifier comes into play.

Existential quantifier

Representing ‘somebody likes somebody’ (assume the domain of all people).

- There exists (at least) a person x, such that x likes somebody
- There exists (at least) a person x and (at least) a person y such that x likes y.
- Let $\exists x$ stand for “there exists an element of the domain x” and $Likes(x, y)$ mean that x likes y, we get: $\exists x\, \exists y\, Likes(x, y)$.

We want to leave our options options as to whom x likes, so we choose a different variable y. y may still point to same person as x, but this is not necessarily the case.

Domain

Domain of discourse

When we translated '*Somebody likes somebody*' into $\exists x \exists y\, Likes(x, y)$, we simplified the representation by *assuming* that the quantifiers operated over a certain set (the set of all people).

The set on which the quantification operates is called the *domain (or universe) of discourse*.

If our domain of discourse contained more than just people, we would need to distinguish the things in the domain that are people, for instance with a predicate $Person(x)$.

$$\exists x \exists y (Person(x) \wedge Person(y) \wedge Likes(x, y))$$

Domain of discourse

- The domain of discourse is fixed at the outset and unique for a set of statements.
- You cannot have one domain for some formulae and another for some other formulae as they will be interpreted against one single domain.
- All your quantifiers will apply to this domain.

Combining quantifiers

Anybody who likes John likes at least one of John's brothers.

Domain: the set of all people. Step-by-step transformation:

- For all x, if x likes John, then there exists a y such that y is a brother of John and x likes y
- Let $Likes(x, y)$ denote 'x likes y'
- Let $Brother(x, y)$ denote 'x is a brother of y'
- Let the constant a denotes the object 'John'

We then have

$$\forall x(Likes(x, a) \to \exists y(Brother(y, a) \wedge Likes(x, y)))$$

FUNCTION SYMBOLS

Function symbols

Every even integer greater than 2 *is the sum of two prime integers*

<u>Domain:</u> the set of positive integers. Step-by-step transformation:

- Let $Even(x)$ denote 'x is even'
- Let $Prime(x)$ denote 'x is a prime integer'
- Let $=$ be a special binary predicate – the equality relation
- Let $>$ be a special binary predicate – where $>(a, b)$ means that the number a is greater than the number b
- Let $+$ be a binary *function symbol* that forms the individual **term**

$$+(y, z)$$

out of the individual terms y and z.

Function symbols

Unlike predicate symbols, which have a true/false interpretation, function symbols interpret to an element in the domain.

We want $+(y, z)$ to interpret to the number corresponding to the sum of y and z.

We then have

$$\forall x\big((Even(x) \land >(x, 2)) \to \exists y \exists z(Prime(y) \land Prime(z) \land =(x, +(y, z)))\big)$$

Some simplifications

We normally use the more common infix form for $=$, $>$ and $+$ and hence the formula

$$\forall x\big((Even(x) \land >(x, 2)) \to \exists y \exists z(Prime(y) \land Prime(z) \land =(x, +(y, z)))\big)$$

can be re-written as

$$\forall x\big((Even(x) \land (x > 2)) \to \exists y \exists z(Prime(y) \land Prime(z) \land (x = (y + z)))\big)$$

which is easier to read.

Reading formulae

Exercises

Translate the following formulae into English, taking the domain of discourse to be the set of rational numbers $\mathbb{Q}$. Assume that the predicates '$<$' and '$=$' and the function symbol $*$ have their usual meanings.

- $\forall x \forall y (x < y \rightarrow \exists z (x < z \wedge z < y))$
- $\forall x \forall y (\neg (y = 0) \rightarrow \exists z (x = y * z))$

We are now ready to introduce the language of predicate logic more formally.

TERMS

Terms

A **term** is either a variable, a constant, or a function symbol applied to arguments that are terms.

- **Individual variables** (x, y, z, etc., possibly with indexes) are 'placeholders' for arbitrary objects from the domain.
- **Individual constants** (a, b, c, etc., possibly with indexes) denote particular objects from the domain.
- **Function symbols** (f, g, h, etc., possibly with indexes) denote particular functions over the domain.

Examples: x, a and $f(x, g(b))$, $x + g(b)$ are terms.

Atomic formulae

Atomic formulae are made of predicate symbols applied to arguments that are terms (they have no connectives).

Unary predicate symbols denote *subsets* of a specified domain.

Binary predicate symbols denote binary *relations*, and so forth.

For example, $P(x)$ can denote the properties 'x is even', 'x is red', 'x is larger than 0' and so on. $Q(x, y)$ can denote the binary relations 'x is larger than y', 'x lives in y' and so on. We give an intended meaning when we interpret the formula.

Examples: $P(x)$, $P(f(x, y))$, $x + x = 3$, $Q(x, y)$.

Terms vs. atomic formulae

It is important to understand that although functional terms and atomic formulae can have any number of arguments they are interpreted differently.

A term (including variables, constants and function terms) always evaluates to an element of the domain, whereas an atomic formula always evaluates to true or false.

Well-Formed Formulae

Atomic formulae can be combined via logical connectives to form more complex formulae. A complex formulae that is correctly constructed is called a *well-formed formula* (wff).

The set of all well-formed formulae is defined inductively as follows:

1. Any atom is a wff (e.g. $P(x),\ Q(x,y),\ x > y,\ x = a$)
2. If $\mathcal{F}$ and $\mathcal{G}$ are wffs, then $\neg(\mathcal{F})$, $(\mathcal{F} \wedge \mathcal{G})$, $(\mathcal{F} \vee \mathcal{G})$, $(\mathcal{F} \rightarrow \mathcal{G})$ and $(\mathcal{F} \leftrightarrow \mathcal{G})$ are wffs
3. If $\mathcal{F}$ is a wff and x is a variable then $\forall x(\mathcal{F})$ and $\exists x(\mathcal{F})$ are wffs

A 'substring' of a wff $\mathcal{F}$ which is itself well-formed is a *subformula* of $\mathcal{F}$.

Connective precedence

We assume that the logical propositional connectives have the same precedence as before.

In addition, $\forall$ and $\exists$ have the same precedence as $\neg$.

Example

$\forall x(\neg(\exists y(\forall z(P(x,y,z)))))$ (parenthesised form)

$\forall x \neg \exists y \forall z P(x,y,z)$ (unparenthesised form)

Scope

Variable Scope and Binding

In the wff $\forall x(\mathcal{F})$, the formula $\mathcal{F}$ is the **scope** of $\forall x$. When $\forall x(\mathcal{F})$ is a subformula in a wff $\mathcal{G}$ then the scope of $\forall x$ in $\mathcal{G}$ is $\mathcal{F}$. The same applies to the existential quantifier $\exists$.

Example. In $\exists x(P(x)) \wedge Q(x)$, the scope of $\exists x$ is $P(x)$ whereas in $\exists x(P(x) \wedge Q(x))$ the scope is $P(x) \wedge Q(x)$.

An occurrence of a variable x in a wff is **bound** iff it occurs within the scope of a quantifier or x is the variable in $\forall x$ or $\exists x$. An occurrence of a variable is **free** if it is not bound.

Variable Scope and Binding

A variable can have free and bound occurrences in a wff.

A wff that contains no free variables is said to be **closed** and a wff with free variables is said to be **open**.

A closed wff is also called a **sentence**.

Example One

1. $\exists x P(x, y) \to Q(x)$.

This is the same as $\exists x(P(x, y)) \to Q(x)$ because of the priority of the connectives.

The scope of $\exists x$ is $P(x, y)$. Therefore, the first two occurrences of x (the one directly next to the quantifier $\exists$ and the one in $P(x, y)$) are bound, whereas the third occurrence is free.

The only occurrence of y is free.

Example Two

2. $\exists x \exists y(P(x, y) \rightarrow Q(x))$

The scope of $\exists x$ is $\exists y(P(x, y) \rightarrow Q(x))$. Therefore all three occurrences of x are bound.

The scope of $\exists y$ is $P(x, y) \rightarrow Q(x)$. Therefore, both occurrences of y are bound.

Note that this formula is a sentence.

SEMANTICS

Introducing Predicate Logic Interpretations

An interpretation $\mathcal{I}$ for a predicate logic language consists of a **nonempty** domain D of objects, over which the variables may range, together with an assignment of a meaning to the predicate, constant and function symbols.

An interpretation enables a truth-value to be assigned to a sentence.

Example. Consider the sentence $\exists x(x + x = 3)$.

(i) If the domain is $\mathbb{N}$ (natural numbers), $+$ is interpreted as sum, and 3 denotes the natural number 3, then the sentence is false.

(ii) If the domain is $\mathbb{Q}$ (rational numbers), then the sentence is true.

Another Example

Let us give an interpretation to the wff

$$\forall x \exists y S(x, y)$$

Let the domain be $\mathbb{N}$ and $S(x, y)$ mean that y is the successor of x. Then the given wff can be translated in the following ways:

- *For every natural number x there exists a natural number y such that y is the successor of x.*
- *There is no greatest natural number.*

Note. This formulae is true in this interpretation.

Semantics of the Logical Connectives

The logical connectives are interpreted in the same way as in the propositional case.

If $\mathcal{F}$ and $\mathcal{G}$ are sentences and $\mathcal{I}(\mathcal{F})$ and $\mathcal{I}(\mathcal{G})$ are the truth-values of $\mathcal{F}$ and $\mathcal{G}$ under the interpretation $\mathcal{I}$. Then:

- $\mathcal{I}(\neg(\mathcal{F})) = 1$ iff $\mathcal{I}(\mathcal{F}) = 0$.
- $\mathcal{I}(\mathcal{F} \vee \mathcal{G}) = 1$ iff $\mathcal{I}(\mathcal{F}) = 1$ or $\mathcal{I}(\mathcal{G}) = 1$.
- $\mathcal{I}(\mathcal{F} \wedge \mathcal{G}) = 1$ iff $\mathcal{I}(\mathcal{F}) = 1$ and $\mathcal{I}(\mathcal{G}) = 1$.
- $\mathcal{I}(\mathcal{F} \rightarrow \mathcal{G}) = 0$ iff $\mathcal{I}(\neg\mathcal{F} \vee \mathcal{G}) = 1$.
- $\mathcal{I}(\mathcal{F} \leftrightarrow \mathcal{G}) = 0$ iff $\mathcal{I}(\mathcal{F} \wedge \mathcal{G}) = 1$ or $\mathcal{I}(\neg\mathcal{F} \wedge \neg\mathcal{G}) = 1$.

Substituting for a variable

Given the domain D, let us use the elements of D as individual constants in formulae interpreted over D.

Let $\mathcal{F}$ be a wff interpreted over D and $d \in D$. Then the expression $\mathcal{F}(x/d)$ denotes the wff obtained from $\mathcal{F}$ by replacing all **free** occurrences of x by d.

Example. Let $\mathcal{F}$ be $P(x) \wedge \exists x \exists y P(x, y)$ and $D = \mathbb{N}$. Then $\mathcal{F}(x/0)$ is $P(0) \wedge \exists x \exists y P(x, y)$.

Interpreting the Quantifiers

Let $\mathcal{I}$ be an interpretation over domain D. The meaning of the quantifiers is defined as follows.

- $\mathcal{I}(\forall x(\mathcal{F})) = 1$ iff **for every** substitution for x of an element d taken from D, we have that $\mathcal{I}(\mathcal{F}(x/d)) = 1$. Otherwise $\mathcal{I}(\forall x(\mathcal{F})) = 0$.
- $\mathcal{I}(\exists x(\mathcal{F})) = 1$ iff **for some** substitution for x of an element d taken from D, we have that $\mathcal{I}(\mathcal{F}(x/d)) = 1$. Otherwise $\mathcal{I}(\exists x(\mathcal{F})) = 0$.

Example

Let our domain be $D = \{c_1,\ c_2,\ c_3,\ c_4,\ c_5\}$ and assume *Blue* and *Fast* are unary predicate symbols and *my_car* is an individual constant.

Let *Blue* be interpreted as $\{c_1, c_2, c_3\}$, i.e., $Blue(x)$ is true iff $x \in \{c_1, c_2, c_3\}$; let *Fast* be interpreted as $\{c_1, c_2, c_3,\ c_4\}$; and let the constant *my_car* be interpreted as c_1. Then we have that

- '*Blue cars are fast*' can be translated as the sentence $\forall x(Blue(x) \rightarrow Fast(x))$
- '*My car is blue*' and '*My car is fast*' can be translated as $Blue(my\_car)$ and $Fast(my\_car)$, respectively
- and all three sentences are true under the above interpretation

Exercises

Let $\mathcal{F} = \exists x \forall y (P(y) \rightarrow x = y)$. Determine whether $\mathcal{F}$ is true or false in each one of the interpretations that follow.

1. Let $D = \{a\}$ and $P(a)$ be true.
2. Let $D = \{a\}$ and $P(a)$ be false.
3. Let $D = \{a, b\}$ and both $P(a)$ and $P(b)$ be true.
4. Let $D = \{a, b\}$ and both $P(a)$ and $P(b)$ be false.
5. Let $D = \{a, b\}$, and $P(a)$ be true and $P(b)$ be false.

Categories of predicate logic sentences

- A predicate logic sentence is **logically valid** (**tautology**) if it is true under **every** interpretation. **Example.** $\forall x P(x) \rightarrow \exists x P(x)$.
- A predicate logic sentence is **satisfiable**, if it is **true under some** interpretation. **Example.** $\forall x (P(x) \rightarrow Q(x))$.
- A predicate logic sentence is **unsatisfiable** (**contradiction**), if it is **false under every** interpretation. **Example.** $\exists x (P(x) \land \neg P(x))$.

As in propositional logic, two sentences $\mathcal{F}$ and $\mathcal{G}$ are **logically equivalent** ($\mathcal{F} \equiv \mathcal{G}$) if $\mathcal{F} \leftrightarrow \mathcal{G}$ is logically valid.

Relationships between quantifiers

$\neg\forall \boldsymbol{x}\mathcal{F} \;\equiv\; \exists \boldsymbol{x}\neg\mathcal{F}$

Example

$\neg\forall x(C(x)\rightarrow T(x)) \equiv \exists x\neg(C(x)\rightarrow T(x)) \equiv \exists x(C(x)\wedge\neg T(x))$

$\neg\exists \boldsymbol{x}\mathcal{F} \;\equiv\; \forall \boldsymbol{x}\neg\mathcal{F}$

Example

$\neg\exists x(L(x)\wedge V(x)) \equiv \forall x\neg(L(x)\wedge V(x)) \equiv \forall x(\neg L(x)\vee\neg V(x))$

Practical example

Problem. Negate and represent the following statement

> “There is a person who walked on the moon”

Solution. We can always obtain the negation of a statement by placing the phrase *“it is not the case that”* in front of it.

Thus, the negation of the statement is

> “It is not the case that there is a person who walked on the moon.”

Or, equivalently, *“No person walked on the moon.”*

Example (cont.)

"(It is not the case that) there is a person who walked on the moon"

Let $W(x)$ mean "x *walked on the moon*" and take the domain of discourse to consist of all people. Then "there is a person who walked on the moon" can be translated as

$$\exists x W(x)$$

The negation of this is the formula $\neg\exists x W(x)$, which is equivalent to $\forall x \neg W(x)$: whatever value we pick for x, it is not the case that the person referred to by x walked on the moon, and hence , *"No person walked on the moon"*.

Example (cont.)

Note that "There is a person who did not walk on the moon" is not a correct translation of the negation of the original statement.

This would give us $\exists x \neg W(x)$, and in fact, both $\exists x \neg W(x)$ and $\exists x W(x)$ are simultaneously true.

Quantifier replacement rules

$$\forall x \mathcal{F} \equiv \neg\neg\forall x \mathcal{F} \equiv \neg\exists x \neg\mathcal{F}$$

$$\exists x \mathcal{F} \equiv \neg\neg\exists x \mathcal{F} \equiv \neg\forall x \neg\mathcal{F}$$

Examples

$$\begin{aligned} &\forall x(C(x) \to T(x)) &&\equiv \\ &\neg\neg\forall x(C(x) \to T(x)) &&\equiv \\ &\neg\exists x\neg(C(x) \to T(x)) &&\equiv \\ &\neg\exists x(C(x) \land \neg T(x)) && \end{aligned}$$

$$\begin{aligned} &\exists x(L(x) \land V(x)) &&\equiv \\ &\neg\neg\exists x(L(x) \land V(x)) &&\equiv \\ &\neg\forall x\neg(L(x) \land V(x)) &&\equiv \\ &\neg\forall x(\neg L(x) \lor \neg V(x)) && \end{aligned}$$

To know more...

In the “Elementary Logic with Applications” book

- Chapter 6.