4CCS1ELA: Elementary Logic with Applications

Introduction to First-Order Logic

Enrico Malizia

Department of Informatics
King's College London, UK

Term 1, 2020/21

Introductory example

- Propositional logic is not very flexible, because propositional statements are in some way "indivisible", i.e. propositional statements cannot be decomposed into the elements they refer to
- Consider the following propositional statements:

P: All friends of Max are friends of John

Q: Paul is a friend of Max

R: Paul is a friend of John

- We would like to have a system in which R follows from P and Q
- However, this cannot be achieved in propositional logic, because in propositional logic, for this specific case, we have that

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

Introductory example

- Remember that, in propositional logic, $P, Q \models R$ iff all models of P, Q are models of R
- But the truth value of R does not depend on the truth values of P and Q, as there is no special link between them
- E.g., consider the assignment σ such that $\sigma(P) = \sigma(Q) = \texttt{true}$ and $\sigma(R) = \texttt{false}$; this assignment is a model of P and Q, but not a model of R
- In the propositional symbols 'P', 'Q', and 'R', there is no mention of the people that the statements refer to
- For this reason, in propositional logic it is not possible to derive R from P and Q
- We need a richer language!

Objects/Individuals & Domain

- First, the statements in this new richer language must have the possibility to refer to objects/individuals
- In the formulas of this new language, intuitively **terms** will be names (i.e., symbols) for the objects/individuals that we want to refer to
- If in a formula the same term is used multiple times, intuitively we are stating multiple things about the same individual (i.e., distinct statements can be linked by the fact that they refer to the same object(s), and hence they are not so "independent" like in propositional logic)
- Intuitively, the domain is the set of objects (named by terms) which the statements of this new language talk about
- In the previous example, objects are Max, John, and Paul, which can be denoted by the terms max, $john$, and $paul$, respectively
- **Notation convention:** term symbols begin with a lower-case letter

Predicates: Properties and Relationships

- *Predicates* are the building blocks of statements in first-order logic
- A predicate can be used to assert a property of an object or a relationship between objects
- Applying a predicate to one or more terms gives rise to an **atom**, which is also a simple formula of this language, called **atomic formula**
- A **monadic** (or **unary**) **predicate** can be used to ascribe a property to a single individual (denoted by a term); the notation has the form

$$Predicate(term)$$

Examples:

$White(tajmahal)$	The Taj Mahal is white
$Angry(john)$	John is angry
$Prime(29)$	29 is prime
$Flower(daisy)$	The daisy is a flower

Predicates: Properties and Relationships

- A **polyadic predicate** can be used to ascribe a relation to an ordered list (tuple) of individuals (denoted by terms); the notation has the form

$$Predicate(term_1, \ldots, term_n),$$

where $Predicate$ is an n-place (or, n-ary) predicate; we say that n is the arity of the predicate

Examples:

$Older(john, mary)$	John is older than Mary
$Student(john, kcl)$	John is a student at KCL
$In(helsinki, finland)$	Helsinki is in Finland
$Divides(13, 91)$	13 divides (i.e., is a factor of) 91
$Between(john, mary, anne)$	John is between Mary and Anne
$QuotRem(11, 50, 4, 6)$	11 divides 50 four times with remainder 6

Functions: references to other objects

- A **function** symbol is used to specify an individual by means of one or more other individuals (denoted by terms)
- The notation, for an n-ary function symbol, is

$$function(term_1, \ldots, term_n),$$

  also in this case we say that n is the arity of the function symbol

Examples:

$mother(john)$	John's mother
$prime\_minister(uk)$	The Prime Minister of UK
$square\_root(17)$	The square root of 17
$product(19, 91)$	19 times 91 (i.e. the number result of the product)

Predicates vs. Functions

- Syntactically, functions and predicates look pretty similar!
- What is the difference between functions and predicates?
- Intuitively, the result of applying a function to a list of terms is another term; whereas the result of applying a predicate to a list of terms is a (very simple) formula, i.e. an atomic formula
- What is the difference between a term and a formula, then?
- A term refers to an individual, it is like a *name*; whereas a formula asserts that something is the case, it is like a *statement*

Predicates vs. Functions — Example

- The term $mother(john)$ denotes a person, John's mother
- The formula $Hungry(john)$ asserts that John is hungry
- In a given situation, it is either true or false that "John is hungry"; it doesn't make sense to say it is true or false that "John's mother"
- I can ask who "John's mother" is; it doesn't make sense to ask who (or what) "John is hungry" is
- **Notation convention:** function symbols begin with a lower-case letter; predicates with a capital letter

A combination of functions and predicates

- Since functional terms, like $mother(john)$, evaluate to a term, then they can be used in combinations with other functions and predicates

Examples:

$Hungry(mother(john))$	John's mother is hungry
$mother(mother(john))$	John's grandmother
$Hungry(mother(mother(john)))$	John's grandmother is hungry
$Greater(square\_root(17), 4)$	The square root of 17 is greater than 4
$Founded(kcl, day(14, august, 1829))$	KCL was founded on the 14^{th} of August 1829

Equality

- First-order logic is equipped with a special symbol '='
- It states that two *objects* equal to one another
- The notation has the form

$$term_1 = term_2,$$

  and also this one is an **atomic formula**
- Examples of correct use of the equality symbol can be

$$kcl = universityof(paul)$$
$$1 = successor(0)$$
$$\neg(1 = successor(2))$$

- Examples of non-correct use can be

$$Person(paul) = Friend(x, john)$$
$$\neg(kcl = Student(john, ucl))$$

  as the equality can be used only between objects, and not predicates

More complex formulas

- More complex formulas are built up from *atomic formulas* using propositional logical connectives like in propositional logic

Examples:

$Hungry(john) \vee \neg Hungry(john)$ John is hungry or not

$Hot(sun) \wedge Star(sun)$ The sun is hot and a star

$Closed(office(john)) \rightarrow (Happy(john) \wedge Goes(john, park))$
If John's office is closed then John is happy and goes to the park

$(mother(john) = mother(anne)) \rightarrow Siblings(john, anne)$
If John's mother is the same of Anne's mother, then John and Anne are siblings

$(Happy(bob) \leftrightarrow Goes(bob, park)) \rightarrow (\neg Happy(bob) \rightarrow \neg Goes(bob, park))$
From the fact that Bob is happy if and only if he goes to the park follows that if Bob is not happy then Bob is not at the park

Variables and Quantifiers

- Terms, predicates, and functions, are still not enough to encode a statement like "All friends of Max are friends of John", because this statement refers to "unnamed people"
- For statements like the one above, we need to introduce symbols that are "placeholders" for unnamed individuals/objects
- These placeholders are the **variables**, such as x, y, z, x_1, y_2, etc.
- A variable is a *term* which does not refer to a fixed individual but is free to vary its reference over the *whole* range of available individuals
- Via variables we can have more generic statements, e.g.:

$Friend(x, john)$	x is a friend of John
$Student(paul, x)$	Paul is a student at x
$Person(x)$	x is a person

Variables and Quantifiers

- Focus again on the statement "All friends of Max are friends of John"
- We need a way to express that the statement refers to *all* people that are friends of Max (and whom are not more specifically named)
- Quantification is a means of expressing statements which do not refer to any named individuals
- There are two kinds of **quantifiers**:
  - **existential quantifiers**
  - **universal quantifiers**
- Intuitively, existential quantification is used for saying that *at least one thing* has a certain property
- Intuitively, universal quantification is used for saying that *everything* has a certain property

Existential quantifier

- To say "Someone (i.e., at least one person) likes Jane", we write that, for some x, x likes Jane:

$$(\exists x)(Likes(x, jane))$$

- More in general, the notation for existential quantifiers is

$$(\exists x)(\phi),$$

  where ϕ is a formula in which x, and possibly other variables, may appear (parentheses can be dropped if does not lead to confusion)
- The symbol '$\exists$' is the **existential quantifier**
- Intuitively, $(\exists x)(\phi)$ asserts that there exists *at least one* individual a in the domain such that ϕ holds when all occurrences of the variable x in ϕ are substituted for a

Existential quantifier — Examples

$\exists x(Boy(x) \wedge Likes(x, jane))$	Some boy likes Jane
$\exists x(Boy(x) \wedge Likes(jane, x))$	Jane likes some boy
$\neg\exists x Likes(x, jane)$	No one likes Jane
$\exists x(Likes(jane, x) \wedge Likes(x, mary))$	Jane likes someone who likes Mary
$\exists x \exists y(Likes(x, y) \wedge \neg(x = y))$	Someone likes someone else
$\exists x(\neg\exists y(Likes(x, y)))$	Someone does not like anyone

Universal quantifier

- To say "Everyone likes Jane", we write that, for every x, x likes Jane:

$$(\forall x)(Likes(x, jane))$$

- More in general, the notation for universal quantifiers is

$$(\forall x)(\phi),$$

  where ϕ is a formula in which x, and possibly other variables, may appear (parentheses can be dropped if does not lead to confusion)
- The symbol '$\forall$' is the **universal quantifier**
- Intuitively, $(\forall x)(\phi)$ asserts that, for every individual a in the domain, ϕ holds when all the occurrences of the variable x in ϕ are substituted for a

Universal quantifier — Examples

$\forall x(Boy(x) \rightarrow Likes(x, jane))$ Every boy likes Jane

$\neg\forall x Likes(x, jane)$ Not everyone likes Jane

$\forall x \neg Likes(x, jane)$ Everyone does not like Jane

$\forall x(Likes(x, jane) \rightarrow \neg Likes(x, bob))$

Everyone who likes Jane does not like Bob

$\forall x(Likes(x, jane) \rightarrow Likes(jane, x))$ Jane likes everyone who likes her

$\forall x \forall y(Likes(x, y) \rightarrow Likes(y, x))$

Everyone is liked by each person that they like

Mixing quantifiers — Examples

In the following examples, we assume that the domain of individuals contains people and universities:

$\forall x(Person(x) \rightarrow \exists y(University(y) \wedge Student(x, y)))$

Everyone studies at some university

$\forall x(University(x) \rightarrow \exists y(Person(y) \wedge Student(y, x)))$

Every university has at least one student

$\exists x(Person(x) \wedge \forall y(University(y) \rightarrow Student(x, y)))$

Someone studies in all universities

$\exists x(University(x) \wedge \forall y(Person(y) \rightarrow Student(y, x)))$

There is a university in which everyone studies

$\forall x(Person(x) \rightarrow \neg(\exists y \exists z(University(y) \wedge University(z) \wedge \neg(y = z) \wedge Student(x, y) \wedge Student(x, z))))$

Everyone studies in at most one university

(Literally: for every person, there are no two distinct universities in which that person studies)

Scope of the quantifiers

- Consider the quantified formula

$$(Qx)(\phi),$$

  where Q is a quantifier (either $\exists$ or $\forall$) and ϕ is a formula
- In the formula $(Qx)(\phi)$, the **scope** of Qx is ϕ
- In the formula $(Qx)(\phi)$, the occurrence of the variable x immediately after the quantifier symbol Q is said to be **bound**
- In the formula $(Qx)(\phi)$, all the occurrences of the variable x in ϕ, i.e. all the occurrences of x in the scope of Qx, are said to be **bound**
- An occurrence of a variable that is not bound is said to be **free**
- A formula without free variables is called a **sentence**

Scope of the quantifiers — Examples

Examples:

$(\exists x)P(x,y) \wedge R(x)$	The scope of $\exists x$ is $P(x,y)$ The first two occurrences of x are bound, whereas the third occurrence of x is free The occurrence of y is free
$(\exists x)(P(x,y) \wedge R(x))$	The scope of $\exists x$ is $P(x,y) \wedge R(x)$ All occurrences of x are bound The occurrence of y is free
$(\exists x)((\forall y)P(x,y) \wedge R(x))$	The scope of $\exists x$ is $(\forall y)P(x,y) \wedge R(x)$ The scope of $\forall y$ is $P(x,y)$ The occurrences of all variables are bound

- The third formula is a sentence, whereas the first two formulas are not

Concluding with the introductory example

- Consider again the three initial statements:

  All friends of Max are friends of John

  Paul is a friend of Max

  Paul is a friend of John

- We can encode them in a first-order logic language:

$$(\forall x)(Friend(x, max) \rightarrow Friend(x, john))$$
$$Friend(paul, max)$$
$$Friend(paul, john)$$

- In a first-order logic system, the third statement can be derived from the first two
- To achieve this, we need to formally define the semantics of the logic, and the notion of logical consequence in FOL