ECM3404: Logic and Computation

Logical Preliminaries I: The Language of First-order Logic

Antony Galton, February 2009

ECM3404: Logic and Computation

Atomic Formulae

Atomic Propositions and Monadic Predication

An **atomic proposition** ascribes a property or relation to one or more individuals. In logic, it is expressed by an **atomic formula** constructed by applying a **predicate** to one or more **terms**.

A **monadic predicate** is used to ascribe a property to a single individual, denoted by a term called an *individual constant*. The standard notation has the form $Predicate(term)$.

Examples

$White(tajmahal)$	The Taj Mahal is white
$Angry(john)$	John is angry
$Prime(29)$	29 is prime
$Cow(daisy)$	Daisy is a cow

Polyadic Predication

A **polyadic predicate** is used to ascribe a relation to an ordered list (tuple) of individuals. The standard notation has the form

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

where *Predicate* is an n-place (or n-ary) predicate.

Examples

$Older(john, mary)$	John is older than Mary
$Loves(john, mary)$	John loves Mary
$In(helsinki, finland)$	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

A **function symbol** is used to specify an individual in terms of one or more other individuals.

The standard pattern, for an n-ary function symbol, is

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

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

Some Frequently Asked Questions

What is the difference between a function symbol and a predicate?

The result of applying a function symbol to a term or list of terms is another term.
The result of applying a predicate to a term or list of terms is a formula.

What is the difference between a term and a formula?

A term refers to an individual. It is like a **name**.
A formula asserts that something is the case. It is like a **statement**.

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.

Prefix vs Infix Notation

In standard logical practice, predicates and function symbols are written as **prefixes**, i.e., before the terms they apply to.

This is different from ordinary language and standard mathematical notation, where various different conventions are used.

Examples

Infix: *John loves Mary* instead of $Loves(mary, john)$.
$45 > 23$ instead of $Greater(45, 23)$.
19×91 instead of $product(19, 91)$.

Postfix: *John's mother* instead of $mother(john)$.
3^2 instead of $square(3)$

Identity

The symbol '$=$' is used as a distinguished binary predicate, written as an infix rather than a prefix, i.e., $term_1 = term_2$.

This means that $term_1$ denotes the very same individual as $term_2$.

Examples

$anne = mother(james)$	Anne is James's mother
$product(19, 91) = 1729$	19 times 91 is 1729.

ECM3404: Logic and Computation

Compound Formulae: Connectives

Compound Formulae

Compound formulae are built up from atomic formulae using **connectives**.

In classical logic, the connectives are **Boolean** or **truth-functional**. This means that the truth value of a compound formula is a function of the truth values of its components.

The main boolean connectives are:

conjunction
disjunction
negation
conditional
biconditional

Boolean Connectives I

Conjunction

For any formulae A and B, the conjunction $A \wedge B$ is true if and only if A and B are both true.

Example

$Hungry(john) \wedge Thirsty(mary)$
(John is hungry and Mary is thirsty)

Truth table

A	B	$A \wedge B$
True	*True*	*True*
True	*False*	*False*
False	*True*	*False*
False	*False*	*False*

Boolean Connectives II

Disjunction

The disjunction $A \vee B$ is true if and only if at least one of A and B is true.

Example

$Hungry(john) \vee Thirsty(mary)$
(John is hungry or Mary is thirsty)

Truth table

A	B	$A \vee B$
True	*True*	*True*
True	*False*	*True*
False	*True*	*True*
False	*False*	*False*

Boolean Connectives III

Negation

The negation $\neg A$ is true if and only if A is false.

Example

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

Truth table

A	$\neg A$
True	*False*
False	*True*

Boolean Connectives IV

Conditional

The (material) conditional $A \rightarrow B$ is equivalent to $\neg A \vee B$: so it is only *false* if A is true and B is false.

Example

$Hungry(john) \rightarrow Thirsty(mary)$
(If John is hungry, then Mary is thirsty)

Truth table

A	B	$A \rightarrow B$
True	*True*	*True*
True	*False*	*False*
False	*True*	*True*
False	*False*	*True*

Boolean Connectives IV

Biconditional

$A \leftrightarrow B$ is true so long as A and B are both true or both false.

Example

$Hungry(john) \leftrightarrow Thirsty(mary)$
(John is hungry if and only if Mary is thirsty)

Truth table

A	B	$A \leftrightarrow B$
True	*True*	*True*
True	*False*	*False*
False	*True*	*False*
False	*False*	*True*

Exercise

Translate these sentences into logical notation, using the key

A	Anne will go	C	Carol will go
B	Bill will go	D	Dan will go

If Bill goes then Dan will go	$B \to D$
If Anne goes then Bill will not go	$A \to \neg B$
Anne will go if Bill and Carol don't go	$\neg B \wedge \neg C \to A$
If Bill goes but Carol doesn't then either Anne or David will go	$(B \wedge \neg C) \to (A \vee D)$
Bill won't go unless Carol goes	$\neg C \to \neg B$ $B \to C$ $\neg B \vee C$
Anne and Carol will only go if David does not go	$(A \wedge C) \to \neg D$ $D \to \neg(A \wedge C)$

ECM3404: Logic and Computation

Compound Formulae: Quantifiers

Quantification

Quantification is a means of expressing propositions which do not refer to any named individuals.

There are two types: **universal** and **existential** quantification.

Universal quantification is used for saying that **everything** has a certain property.

Existential quantification is used for saying that **at least one thing** has a certain property.

Combined with negation, these can also be used for saying that **nothing** or **not everything** has a certain property.

Quantifiers and Variables I

Quantification is expressed using symbols called **quantifiers** and **variables**.

A **variable** (typically written x, y, z, ...) 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.

Variables correspond approximately to pronouns in ordinary language:

e.g., $Loves(x, jane)$ means something like 'he/she loves Jane'.

Quantifiers and Variables II

To say 'Everybody loves Jane', we say that for every x, x loves Jane:

$$\forall x Loves(x, jane).$$

The symbol $\forall$ is the **universal quantifier**.

Similarly, to say 'Somebody (i.e., at least one person) loves Jane', we say that for some x, x loves Jane:

$$\exists x Loves(x, jane).$$

The symbol $\exists$ is the **existential quantifier**.

Universal Quantifier examples

$\forall x(Boy(x) \to Loves(x, jane))$ — Every boy loves Jane

$\neg\forall x Loves(x, jane)$ — Not everyone loves Jane

$\forall x \neg Loves(x, jane)$ — No-one loves Jane

$\forall x(Loves(x, jane) \to Loves(x, bob))$
Everyone who loves Jane also loves Bob

$\forall x(Loves(x, jane) \to \neg Loves(x, bob))$
No-one who loves Jane also loves Bob

$\forall x(Loves(x, jane) \to Loves(jane, x))$
Jane loves everyone who loves her

$\forall x \forall y(Loves(x, y) \to \neg Blackmails(x, y))$
No-one blackmails someone they love

Existential Quantifier examples

$\exists x(Boy(x) \wedge Loves(x, jane))$ — Some boy loves Jane

$\exists x(Boy(x) \wedge Loves(jane, x))$ — Jane loves some boy

$\neg\exists x Loves(x, jane)$ — Nobody loves Jane

$\exists x(Loves(jane, x) \wedge Loves(x, mary))$
Jane loves somebody who loves Mary

$\exists x \exists y(Loves(x, y) \wedge \neg x = y)$
Somebody loves somebody else

Mixed quantifier examples

$\forall x(Girl(x) \to \exists y(Boy(y) \wedge Loves(x, y)))$
Every girl loves some boy

$\forall x(Girl(x) \to \exists y(Boy(y) \wedge Loves(y, x)))$
Every girl is loved by some boy

$\exists x(Girl(x) \wedge \forall y(Boy(y) \to Loves(x, y)))$
There is a girl who loves every boy

$\exists x(Girl(x) \wedge \forall y(Boy(y) \to Loves(y, x)))$
There is a girl whom every boy loves

Exercise

Translate these sentences into logical notation, using the key

a	Anne	$R(x,y)$	x respects y
b	Bill	$A(x,y)$	x admires y

Someone admires Anne	$\exists x A(x,a)$
Bill does not respect himself	$\neg R(b,b)$
Nobody admires both Anne and Bill	$\neg\exists x(A(x,a) \wedge A(x,b))$
Everyone who admires Anne respects Bill	$\forall x(A(x,a) \rightarrow R(x,b))$
Bill respects everyone who admires Anne	$\forall x(A(x,a) \rightarrow R(b,x))$
Anne respects everyone who admires her	$\forall x(A(x,a) \rightarrow R(a,x))$
Someone Bill respects admires Anne	$\exists x(R(b,x) \wedge A(x,a))$

ECM3404: Logic and Computation

Models and Interpretations

Interpretation

$\forall x \neg R(x,x)$
$\forall x \forall y \forall z(R(x,y) \wedge R(y,z) \rightarrow R(x,z))$
$\forall x \forall y(R(x,y) \rightarrow \exists z(R(x,z) \wedge R(z,y)))$
$\forall x \exists y R(x,y)$
$\forall x \exists y R(y,x)$

These formulae are not *about* anything until we specify

- the **domain** (or **universe**) of quantification, i.e., what class of objects do the variables range over?
- what the predicate R means.

If we specify these things, then we will have an **interpretation** of the formulae, under which each formula is either true or false.

Example Interpretation I

Domain of quantification: all real numbers.
R means 'is less than'.

No real number is less than itself.
Any real number is less than any real number any real number it is less than is less than.
For any two real numbers, such that one is less than the other, there is a third real number which is greater than the first one and less than the second one.
For any real number, there is a real number that it is less than.
For any real number, there is a real number that is less than it.

These are all true.

Example Interpretation II

Domain of quantification: all students at Exeter University.
R means 'is older than'.

No student is older than himself/herself.
Any student is older than any student any student (s)he is older than is older than.
For any two students, such that one is older than the other, there is a third student who is older than the first one and younger than the second.
For any student, there is another student who is younger.
For any student, there is another student who is older.

Only the first two are true.

Models and Satisfaction

An interpretation under which a formula is true is said to **satisfy** that formula. (Otherwise it **falsifies** it.)

An interpretation which satisfies every member of a set of formulae is called a **model** for that set.

A set of formulae is **satisfiable** if it has at least one model. (Otherwise it is **unsatisfiable**.)

Example. The set

$\forall x \neg R(x,x)$
$\forall x \forall y \forall z(R(x,y) \wedge R(y,z) \rightarrow R(x,z))$
$\forall x \forall y(R(x,y) \rightarrow \exists z(R(x,z) \wedge R(z,y)))$
$\forall x \exists y R(x,y)$
$\forall x \exists y R(y,x)$

is satisfied by the first interpretation above, which is therefore a model for the set.

Inference and Validity

A set of formulae Σ **logically implies** (or **entails**) a formula ϕ if and only if every model for Σ satisfies ϕ.

Example. The set

$\forall x \neg R(x,x)$
$\forall x \forall y \forall z(R(x,y) \wedge R(y,z) \rightarrow R(x,z))$

logically implies

$$\forall x \forall y(R(x,y) \rightarrow \neg R(y,x)).$$

This is called a **valid inference**.

Validity and Unsatisfiability

If every model for Σ satisfies ϕ (so the inference from Σ to ϕ is valid) then the set $\Sigma \cup \{\neg\phi\}$ is unsatisfiable.

Example. The set

$\forall x \neg R(x,x)$
$\forall x \forall y \forall z(R(x,y) \wedge R(y,z) \rightarrow R(x,z))$
$\neg\forall x \forall y(R(x,y) \rightarrow \neg R(y,x))$.

is unsatisfiable.

Hence one way to prove that the inference from Σ to ϕ is valid is to show that the set $\Sigma \cup \{\neg\phi\}$ is unsatisfiable.