G52DOA - Derivation of Algorithms
Predicate Logic

Venanzio Capretta

Predicate Logic

So far, we studied *propositional logic*, in which we started with unspecified *propositional variables* $A, B, C, \ldots$ and combined them into complex propositions using connectives. Eventually, we want to use logic to prove properties of objects in some specific domain. Let's call D this domain, it is a set of entities that we leave undefined for the moment. Possible instantiations of D could be:

- The set of all human beings;
- The set of all real numbers;
- The set of all integers;
- The type of arrays of integers.

The *atomic propositions* involving elements of the domain will be obtained by applying to them some predicate, stating a property that an element may or may not satisfy, or relation, stating a connection of some kind between two or more elements. We use capital letters for predicates and relations: $P, Q, R, \ldots$.

For example, if $D =$ *"the set of all human beings"*, we could use the predicate $P(x) =$ *"x has green eyes"* and the binary relation $R(x, y) =$ *"x is a parent of y"*.

If, instead, we take $D =$ *"the set all of integers"*, we could use the predicate $P(x) =$ *"x is even"* and the binary relation $R(x, y) = (x \leq y)$

Predicate logic is about those propositions that can be deduced independently of the meaning of D, P, and R.

We can combine the atomic propositions using connectives as before. For example, the formula $P(y) \to R(x, y)$ means, in the case in which D is the set of all humans, "if y has green eyes, then x is a parent of y"; or, in the case in which D is the set of integers, "if y is even, then x is smaller than or equal to y".

Besides the connectives, we can construct complex sentences using *quantifiers*, which state how many of the elements of the domain satisfy a certain property. We use two quantifiers: the *universal quantifier* $\forall$ (read "for all"), which states that all elements of the domain satisfy a certain property; and the *existential quantifier* $\exists$ (read "there exists"), which states that at least one element of the domain satisfies the property. (This *property* may already be a complex sentence containing connectives and quantifiers.)

Here are some propositions containing quantifiers, with their translations when D is the set of humans:

$\forall x, P(x)$	All humans have green eyes.
$\exists x, P(x)$	There exists a human with green eyes.
$\forall x, \exists y, R(y, x)$	Every human has a parent.
$\forall x, (P(x) \to \exists y, (R(y, x) \wedge P(y))$	Every human with green eyes has a parent with green eyes.

The translation of the last proposition has been simplified a bit to make it more readable in English. The automatic translation would read: "For all humans x, if x has green eyes, then there exists a human y such that y is a parent of x and y has green eyes".

The same propositions have a different translation when we take D to be the set of all integers:

$\forall x, P(x)$	All integers are even.
$\exists x, P(x)$	There exists an even integer.
$\forall x, \exists y, R(y, x)$	For every integer, there exists an integer smaller or equal to it.
$\forall x, (P(x) \to \exists y, (R(y, x) \wedge P(y))$	For every even integer, there exists a smaller or equal even integer.

Note: The rules of precedence state that quantifiers bind stronger than connectives. Therefore, the proposition $\forall x, P(x) \wedge Q(x)$ is parenthesised as $(\forall x, P(x)) \wedge Q(x)$, placing the x argument of Q outside the scope of the universal quantifier. If we want the scope of the quantifier to extend to the whole proposition, we need to put explicit parentheses: $\forall x, (P(x) \wedge Q(x))$.

Similarly $\exists x, P(x) \to Q(x)$ is interpreted as $(\exists x, P(x)) \to Q(x)$. If we want the scope of the quantifier to extend over the whole implication, then we have to write $\exists x, (P(x) \to Q(x))$.

Natural Deduction Rules for Quantifiers

In the following rules, the *eigenvariable* denotes a completely arbitrary element. Therefore, no assumption can be made about it. We must use for it a variable name about which no properties have been assumed or proved. Therefore we require that **the eigenvariable does not occur free outside the box where it is assumed**. On the other hand, the term t can be any element, so we are free to choose one about which some properties have been already proved or assumed.

Universal Quantification

Introduction

	$\vdots$	
n	y	eigenvariable
	$\vdots$	
m	$A[y]$	
	$\vdots$	
k	$\forall x, A[x]$	$\forall_i\ n - m$

Elimination

	$\vdots$	
n	$\forall x, A[x]$	
	$\vdots$	
k	$A[t]$	$\forall_e\ n$

Existential Quantification

Introduction

	$\vdots$	
n	$A[t]$	
	$\vdots$	
k	$\exists x, A[x]$	$\exists_i\ n$

Elimination

	$\vdots$	
h	$\exists x, A[x]$	
	$\vdots$	
n	y	eigenvariable
$n+1$	$A[y]$	assumption
	$\vdots$	
m	B	
	$\vdots$	
k	B	$\exists_e\ h, n - m$

Notice that, in the existential elimination rule, y cannot occur free in B because of the restriction on the eigenvariable. In fact, we required that the eigenvariable does not occur free outside the box where it is assumed; since B occurs outside the box, it cannot contain y.

Syntax of predicate logic

Predicate logic, also called *first order logic*, consists of the following elements.

- A domain of discourse D. This is actually never mentioned in the propositions; simply, every element we talk about is implicitly assumed to belong to this domain.
- Identifiers for predicates and relations: P, Q, R, etc. In our terminology, relations are considered predicates with many arguments, so from now on I will call them all predicates. For every predicate we must specify the number of elements that it takes, its *arity*.
- We may also use some identifiers for *functions*: f, g, etc. These denote operations that map elements of D to elements of D. Also functions should have an arity. We will talk more about function symbols later.

The list of predicates and functions with their arity is called a *signature*. For example, the following is a signature:

	name	arity
relations	P	1
	Q	1
	R	2
functions	f	1
	g	2

The syntax of predicate logic consists of terms and propositions (or *formulas*).

Terms are expressions that denote elements of the domain D. They are constructed inductively starting with a set of variables: $x, y, z, a, b, c, \ldots$; and making more complex terms by applying the function symbols. Some terms of the above signature are the following: x, a, $f(x)$, $g(x, a)$, $f(g(x, f(a)))$, $g(f(z), g(f(a), b))$.

Propositions are formulas denoting statements about the domain D that can be either true or false. They are constructed inductively starting with atomic formulas, combined using connectives and quantifiers. *Atomic formulas* are predicate symbols applied to terms. For example, atomic formulas for the signature above are: $P(x)$, $R(a, b)$, $Q(f(x))$, $R(g(x, b), f(a))$. Finally, here are some examples of complex propositions for the signature above:

$$R(f(a), g(a, b)) \rightarrow Q(b)$$
$$\forall x, (P(f(x)) \rightarrow \exists y, R(f(y), g(x, y)))$$
$$\forall x, \exists y, ((P(x) \vee Q(f(x))) \wedge R(x, g(x, f(y))) \rightarrow P(f(g(x, y))) \vee Q(g(y, y))).$$

Models

A *model* of signature consists of an interpretation of the domain of discourse and of the predicates and relations. Predicates and relations are always written with the arguments inside parentheses following the name of the predicate, and separated by commas. For example, if we have a theory with domain D, a unary predicate $\mathsf{GreenEyes}$, a binary relation $\mathsf{Siblings}$, and a ternary relation $\mathsf{Parents}$; we can interpret it with the following model:

$$\begin{aligned} D &= \text{The set of all human beings;} \\ \mathsf{GreenEyes}(x) &= x \text{ has green eyes;} \\ \mathsf{Siblings}(x, y) &= x \text{ and } y \text{ are siblings;} \\ \mathsf{Parents}(x, y, z) &= x \text{ and } y \text{ are the parents of } z. \end{aligned}$$

Nominalism: The philosophical doctrine of *nominalism* holds that the names of things are conventional and bear no relation to their meanings. We adopt this philosophy: the fact that we give suggestive names to our predicates and relations doesn't mean that their interpretation must be the one that the names suggest. For example, an equally good model for the above is:

$$\begin{aligned} D &= \text{The set of integers;} \\ \mathsf{GreenEyes}(x) &= x \text{ is prime;} \\ \mathsf{Siblings}(x,y) &= x < y; \\ \mathsf{Parents}(x,y,z) &= (x+y=z). \end{aligned}$$

Some propositions are true in a particular model but false in another. For example, the following proposition:

$$\forall x, (\forall y, (\exists u, (\exists v, (\mathsf{Parents}(u,v,x) \wedge \mathsf{Parents}(u,v,y))) \to \mathsf{Siblings}(x,y)))$$

is true in the *human* interpretation (it states that *if x and y have the same parents, then they are siblings*); but it is false in the *integer* interpretation (it states that *if x and y are both the sum of two numbers u and v, then x is smaller than y*.

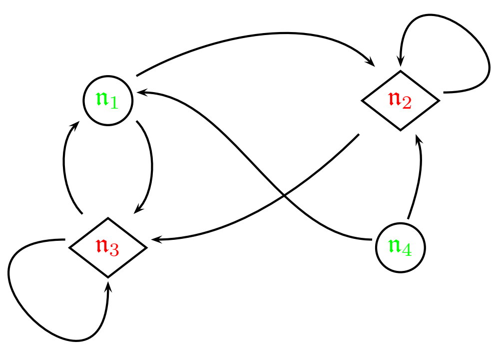
On the other hand, this other proposition:

$$\forall x, (\forall y, (\mathsf{Siblings}(x,y) \to \exists z, \mathsf{Parents}(x,z,y)))$$

is false in the *human* interpretation (it states that *if x and y are siblings, then x is one of y's parents*); but it is true in the *integer* interpretation (it states that *if x is smaller than z, then there exists a number z that added to x gives y*).

Graph models

One particularly simple kind of model is a *graph*. A graph is a set of points, called *nodes*, linked by a set of arrows, called *edges*. In our examples, we colour the nodes either green or red. Here is an example (if you can't see the colours, the green nodes are the ones with a circle around them, the red ones are those with a diamond shape around them):



We use a signature with two unary predicates G and R and a binary predicate (a relation) E. Our interpretation is the following:

- D is the set of nodes of the graph;
- $\mathsf{G}(x)$ means that the node x is green (a circle);

- $\mathsf{R}(x)$ means that the node x is red (a diamond);
- $\mathsf{E}(x, y)$ means that there is an edge from x to y.

For every proposition that we can write down in this signature, we can ask whether it is true in the specific graph model that we have given. Here are a couple of examples.

$$\forall x, (\mathsf{G}(x) \rightarrow \exists y, (\mathsf{E}(x, y) \wedge \mathsf{R}(y)))$$

This proposition states that for every green node, there is at least a red node to which it is connected by an edge. This happens to be true.

$$\exists x, (\mathsf{R}(x) \wedge \forall y, (\mathsf{E}(y, x) \rightarrow \mathsf{G}(y)))$$

This proposition states that there is a red node such that all nodes with edges to it are green. This happens to be false.

We can compute whether a proposition is true or false in a graph model by constructing a truth table. First of all, we determine the values of the atomic propositions for every possible value of the variables:

x	$\mathsf{G}(x)$
$\mathfrak{n}_1$	1
$\mathfrak{n}_2$	0
$\mathfrak{n}_3$	0
$\mathfrak{n}_4$	1

x	$\mathsf{R}(x)$
$\mathfrak{n}_1$	0
$\mathfrak{n}_2$	1
$\mathfrak{n}_3$	1
$\mathfrak{n}_4$	0

x	y	$\mathsf{E}(x, y)$
$\mathfrak{n}_1$	$\mathfrak{n}_1$	0
	$\mathfrak{n}_2$	1
	$\mathfrak{n}_3$	1
	$\mathfrak{n}_4$	0
$\mathfrak{n}_2$	$\mathfrak{n}_1$	0
	$\mathfrak{n}_2$	1
	$\mathfrak{n}_3$	1
	$\mathfrak{n}_4$	0
$\mathfrak{n}_3$	$\mathfrak{n}_1$	1
	$\mathfrak{n}_2$	0
	$\mathfrak{n}_3$	1
	$\mathfrak{n}_4$	0
$\mathfrak{n}_4$	$\mathfrak{n}_1$	1
	$\mathfrak{n}_2$	1
	$\mathfrak{n}_3$	0
	$\mathfrak{n}_4$	0

The value of a quantified proposition is computed by checking its values for all possible interpretation of the quantified variable. In the case of universal quantification, it is true if the formula gives true for all interpretations of the variable, otherwise it is false. In the case of existential quantification, it is true if the formula gives true for at least one interpretaion, otherwise it is false. For example, we will have that $\forall x, \mathsf{G}(x)$ has value 0 because $\mathsf{G}(x)$ doesn't give 1 for all interpretations of x. On the other hand, $\exists x, \mathsf{G}(x)$ has value 1 because there is at least one interpretation of x (for example $\mathfrak{n}_1$) that gives $\mathsf{G}(x)$ the value 1.

Let us consider the proposition $\forall y, \mathsf{E}(x, y)$. In this proposition the variable x is *free*, that is, it is not quantified by either a universal or existential quantifier; whereas the variable y is *bound*, that is, it occurs in the scope of a quantifier. When computing the truth table of a proposition, we must have separate rows for all possible values of the free variables, while we compute the value of the proposition in each row by using the above rules with respect to the bound variable:

x	$\forall y, \mathsf{E}(x, y)$
$\mathfrak{n}_1$	0
$\mathfrak{n}_2$	0
$\mathfrak{n}_3$	0
$\mathfrak{n}_4$	0

This case was not very interesting, since the proposition turns out to be false for every interpretation of x. Let us look at the more interesting proposition: $\exists y, (\mathsf{E}(x, y) \wedge \mathsf{G}(y))$. Here are the truth tables for the sub-formula $\mathsf{E}(x, y) \wedge \mathsf{G}(y)$ and of the proposition obtained by quantifying it:

x	y	$\mathsf{E}(x, y) \wedge \mathsf{G}(y)$
$\mathfrak{n}_1$	$\mathfrak{n}_1$	0
	$\mathfrak{n}_2$	0
	$\mathfrak{n}_3$	0
	$\mathfrak{n}_4$	0
$\mathfrak{n}_2$	$\mathfrak{n}_1$	0
	$\mathfrak{n}_2$	0
	$\mathfrak{n}_3$	0
	$\mathfrak{n}_4$	0
$\mathfrak{n}_3$	$\mathfrak{n}_1$	1
	$\mathfrak{n}_2$	0
	$\mathfrak{n}_3$	0
	$\mathfrak{n}_4$	0
$\mathfrak{n}_4$	$\mathfrak{n}_1$	1
	$\mathfrak{n}_2$	0
	$\mathfrak{n}_3$	0
	$\mathfrak{n}_4$	0

x	$\exists y, (\mathsf{E}(x, y) \wedge \mathsf{G}(y))$
$\mathfrak{n}_1$	0
$\mathfrak{n}_2$	0
$\mathfrak{n}_3$	1
$\mathfrak{n}_4$	1