Logic in Computer Science
Chapter 9

Antonina Kolokolova

February 12, 2014

9.1 Predicate logic

Sometimes we encounter sentences that only have a truth value depending on some parameter. For example, $Even(x)$ which states that the number x is even can be true or false depending on the actual value of x. That is, $Even(5)$ is false, and $Even(10)$ is true.

It is convenient to think of predicates as propositions with parameters. Here, parameters can be numbers, items, etc and there can be infinitely many possibilities for a parameter value. For example, $x^2 > x$ is a predicate with an argument x, where we think of x as a number. Another predicate $Parent(x, y)$ could state that x is a parent of y. Here, it makes sense to think of x and y as people, or at least living creatures. Truth values of a predicate are defined for a given assignment of variables. For example, if $x = 2$, then $x^2 > x$ is true, and if $x = 0.5$, then $x^2 > x$ is false. We call a set of possible objects from which the values of a predicate can come from a *domain* of a predicate.

So what is the relation between a predicate, for example $Parent(x, y)$, and a relation $Parent$? A predicate is true iff the corresponding tuple of values is in the relation. For example, $Parent(John, Mary)$ is true if John is a parent of Mary, and the pair (John, Mary) is in the relation $Parent$. Usually we will use the notation $P(x, y, z)$ to mean a predicate, and just P to denote a set (relation); however sometimes I will abuse the notation and mix up these two concepts (especially when talking about databases).

9.2 Quantifiers

Without fixing the values of arguments of a predicate it is not possible to say if the predicate is true or false. That is, unless we want to say that the predicate is false for all possible values of its arguments (in the domain of this predicate). Here, we need to pay careful attention to what we mean by all possible values: $x^2 \geq x$ is true for and it is false for some rational and real numbers such as 0.5.

Quantifiers are the notational device that allows us to talk about all possible values of arguments and make sentences with truth values out of predicates.

Definition 1. *A formula $\forall x\ A(x)$, where $A(x)$ is a formula containing predicates, is true (on the domain of predicates) if is is true on every value of x from the domain. Here, $\forall$ is called a* universal quantifier*, usually pronounced as "for all ...".*

For example, $\forall x\ x^2 \geq x$ states that for every element from the domain the square of that element is greater than the element itself. This formula now has a truth value, provided we know the domain from which x comes from. If the domain is $\mathcal{Z}$, then the formula is true, and if the domain is $\mathcal{Q}$, then it is false. Often the domain is written explicitly: $\forall x \in \mathcal{Z}\ x^2 \geq x$, which is a shortcut for $\forall x\ (x \in \mathcal{Z} \rightarrow x^2 \geq x)$.

When we want to say that something is not true everywhere, all we need to do is to give a counterexample. E.g., to show that for $\mathcal{Q}$ it is not true that $\forall x\ x^2 \geq x$ it is enough to give one value on which $x^2 \geq x$ does not hold such as $x = 0.5$. We denote this with the second type of quantifiers, an *existential* quantifier.

Definition 2. *A formula $\exists x\ A(x)$, where $A(x)$ is a formula containing predicates, is true (on the domain of predicates) if is is true on* some *value of x from the domain. Here, $\exists$ is called a* existential quantifier*, usually pronounced as "exists ...".*

When doing boolean operations on formulas containing quantifiers, always remember that universal and existential quantifiers are opposites of each other. So,

$$\neg(\forall x\ A(x)) \iff \exists x\ \neg A(x) \qquad\qquad \neg(\exists x\ A(x)) \iff \forall x\ \neg A(x)$$

Now that we have this notation we can define what kinds of formulas we can construct using this language, the *first-order formulas.*

Definition 3. *A predicate is a first-order formula (possibly with free variables). A $\wedge, \vee, \neg$ of a first-order formula is a first-order formula. If a formula $A(x)$ has a free variable (that is, a variable x that occurs in some predicates but does not occur under quantifiers such as $\forall x$ or $\exists x$), then $\forall x\ A(x)$ and $\exists x\ A(x)$ are also first-order formulas.*

Note that this definition is very similar to the definition of propositional formulas except here there are predicates instead of propositions and there are quantifiers.

Lets look at some examples of first-order formulas using the $Parent(x, y)$ relation, saying that x is a parent of y.

- $\exists y\ Parent(x, y) \wedge Parent(z, y)$ says x and y have a common child. Here, x and z are free variables, and y is a bound variable.

- $\exists y\ Parent(x, y) \wedge (\exists y\ Parent(z, y))$ says that both x and y have children. Here, y in the first occurrence of $Parent$ is a different variable from y in the second occurrence. We talk about the *scope* of a quantifier to mean part of the formula in which the variable y is the variable mentioned in the quantifier. Usually, a scope of a quantifier starts from the place it occurs in the formula and continues until either there is another quantifier with the same variable name or the parentheses started before the quantifier are closed.

  Although it is technically allowed to reuse the variable names, and it is sometimes useful in practice (think of a variable name, in software, denoting an allocated chunk of memory, and reusing the name as reusing the memory), for readability it is better to use different names for different variables. For example, the formula above has an equivalent but more readable version $\exists y\ Parent(x, y) \wedge \exists u\ Parent(z, u)$. Note that this would allow us to use u and y together in a predicate later in the formula, if we wish to.

- $\exists y\ Parent(x, y) \wedge Parent(y, z)$ says that x is a grandparent of z

- $\forall x \exists y Parent(y, x)$ says that for everybody somebody is his/her parent. This would be true about most of reasonable domains. However, changing the order of quantifiers here we obtain a formula with a very different meaning: $\exists y \forall x Parent(y, x)$ says that y is everybody's parent, which is not likely to be true.

9.3 English and quantifiers

In English, the closest word to the universal quantifier is "all" or "every". The closest word to the existential quantifier is "some" and "exists". But there is one word that can be used as either a universal or an existential quantifier.That is the word *any.* Often we take it to mean a universal quantifier, as in "take any number greater than 1..." (that is, every number greater than 1 would work). But compare the following two sentences:

> "I will be happy if I do well in *every* class".
> "I will be happy if I do well in *any* class".

Here, the word "any" takes the meaning of an existential quantifier: that is, I'll be happy if there exists some class in which I do well. Please keep this in mind when doing the translations.

Puzzle 8. The first formulation of the famous liar's paradox, done by a Cretan philosopher Epimenides, stated "All Cretans are liars". Is this a paradox?

9.4 Multiple quantifiers

As you might have noticed with the previous examples changing the order of quantifiers in a formula completely changes its meaning. Note that it only applies to changing order of quantifiers of different types; changing the order of two $\exists$ quantifiers next to each other would not change the meaning.

For any integer x there is an integer y such that $x + y = 5$. This is not the same as "there exists y such that for all x $x + y = 5$"!

To make it easier to remember, think of the following English phrase:

Everybody loves somebody

There are two ways of reading it, corresponding to two different orders of quantifiers:

> For every person, there is somebody this person loves (e.g, every person loves their mother).

> There exists a person whom everybody loves (e.g., everybody loves Elvis Presley).

So, when you are translating from English a sentence with alternating quantifiers, think: is the meaning "mother" or is the meaning "Elvis?"

We can extend this Valentine's day example to illustrate how the order of quantifiers and quantifier variables changes the meaning of a formula. Suppose that $Loves(x, y)$ means that x loves y. Now, consider the following four formulas with their meaning:

$\forall x \exists y\ Loves(x, y)$	Each person loves somebody (e.g., loves their own mother).
$\exists x \forall y\ Loves(x, y)$	There is somebody who loves everyone in the world (Mother Teresa?)
$\forall y \exists x\ Loves(x, y)$	Every person has someone who loves them (their mother loves them)
$\exists y \forall x\ Loves(x, y)$	There is somebody who is loved by everyone (pick your favourite celebrity)

Chapter 10

10.1 Negating quantified formulas

Recall again that to prove that something is not true "everywhere" we need to give a counterexample. Here we will do a few examples of negating first-order formulas with quantifiers.

Example 1. Here is an example of negating a formula with multiple quantifiers. We change all quantifiers to the opposite, and then negate the formula under the quantifiers as we would a propositional formula.

$$\begin{aligned} &\neg(\exists x \forall y \forall z \exists u(\neg P(x,y) \vee (Q(z,u) \wedge z \neq y)) \\ \iff &\forall x \exists y \exists z \forall u \neg(\neg P(x,y) \vee (Q(z,u) \wedge z \neq y)) \\ \iff &\forall x \exists y \exists z \forall u (P(x,y) \wedge (\neg Q(z,u) \vee z = y)) \end{aligned}$$

Example 2. Consider the formula $\forall x\ (x^2 > x \vee x < 1)$. Suppose we want to prove that this formula is not true when the domain is real numbers $\mathcal{R}$. For that, we need to give a counterexample to the formula, that is, a real number such that $x^2 \not> x$ and $x \not< 1$. A counterexample that works here is $x = 1$, since $1^2 = 1$, not > 1, and $1 < 1$ does not hold either. The way we write it is
$\neg(\forall x\ (x^2 > x \vee x < 1) \iff \exists x\ \neg(x^2 > x \vee x < 1) \iff \exists x\ (x^2 \leq x \wedge x \geq 1)$ Here, we took a simplification one step further than usual, and wrote $\neg(x^2 > x)$ as $x^2 \leq x$, and the same for $x \geq 1$.

Definition 1. *An* instantiation *of a variable is a specific value that this variable is set to.*

For example, in the formula $x^2 > x \vee x < 1$ above we instantiated x to be 1.

Now we can define what it means for one predicate formula to imply another, and for two formulas to be equivalent. When we say that $A(x,\ldots,z) \to B(x,\ldots,z)$ what we mean is that for every instantiation (sometimes called "interpretation" in this context) of free variables, if $A(x,\ldots,z)$ is true on that instantiation then so is $B(x,\ldots,z)$. Similarly, we say that A is equivalent to B (that is, $A(x,\ldots,z) \iff B(x,\ldots,z)$) if for every instantiation of free variables $A \iff B$) for that instantiation.

10.2 Derivations in predicate logic

One of the main tools in proving mathematical statements, and in deductive reasoning in general, is the rule of universal instantiation.

Definition 2. *The rule of* universal instantiation*: if some property is true of everything in a domain, then it is true of any particular thing in the domain.*

So if $\forall x\ x^2 \geq x$, then $5^2 > 5$. Another example says that every number is either even or odd ($\forall x\ (2|x \vee 2|x+1)$). Therefore, if we take some number k, then k is either even or it is odd.

A classical example of reasoning using the rule of universal instantiation is the following:

All men are mortal
Socrates is a man
$\therefore$ Socrates is mortal.

There are several ways to write this argument in predicate logic. The first will make use of the rule of universal instantiation under the assumption that the domain of the quantifier is "men". The second one that explicitly specifies the domain by using an implication, will do the rule of universal instantiation followed by modus ponens. Finally the third one, most closely resembling the original argument, will combine the universal instantiation and modus ponens into one rule, called *universal modus ponens.*

Let us consider predicates $Man(x)$ and $Mortal(x)$, which are true, respectively, on x that are men, and x that are mortal. Let Men be the set of all men (this is the domain of the $\forall$ quantifier in the first example).

$\forall x \in Men\ Mortal(x)$
$\therefore Mortal(Socrates)$

$\forall x(Man(x) \rightarrow Mortal(x)$
$Man(Socrates) \rightarrow Mortal(Socrates)$
$Man(Socrates)$
$\therefore Mortal(Socrates)$

$\forall x(Man(x) \rightarrow Mortal(x)$
$Man(Socrates)$
$\therefore Mortal(Socrates)$

The general rule for the universal modus ponens, the rule used in the original form of the argument and the third translation into logic, is as follows

$\forall x \ P(x) \to Q(x)$
$P(a)$ for a particular a
$\therefore Q(a)$.

Let us look at a more realistic mathematical proof using universal instantiation. Suppose in a piece of a proof goes as follows:

For all x, m, n, $x^m x^n = x^{m+n}$.
For all x, $x^1 = x$.
Therefore, $r^{k+1} r = r^{k+1} r^1 = r^{k+2}$.

Here, we instantiated $x = r, m = k+1, n = 1$. In the first equality in the last line, we used the second premise and in the second equality the first premise. We also used the fact that $1 + 1 = 2$.

Puzzle 1. A man walks into a bar and says to the barman: "pour everybody a drink! when I drink, everyone drinks!". After he finishes the round, he says again: "pour everybody a drink! when I drink, everyone drinks!". The crowd is quite pleased, until he says: "Give me the bill, I'll pay. When I pay, everybody pays!".

What does it have to do with logic, you may ask? Tell me, is there a man such that when he drinks, everybody drinks?

Chapter 11

11.1 Equivalences and normal forms

Recall the formula $\exists y Parent(x, y) \wedge (\exists y Parent(z, y))$. This was our example of a formula illustrating the scope of a quantifier: here, the scope of the second y is the parentheses where it is located, and the scope of the first quantifier is the rest of the formula. When we were discussing this, we said that a variable could be renamed, to avoid the confusion. But note that if all variables in a formula have different names, then it is possible to make the scope of each of them the whole formula. So the formula above is equivalent to $\exists y \exists u Parent(x, y) \wedge Parent(z, u)$. Note that although we moved $\exists u$ to the front of the formula, we did not change its order relative to $\exists y$. Although for quantifiers of the same type the order can be changed, since it cannot be changed between quantifiers of different types it is good to keep the order of quantifiers the same as it was in the formula. It is often convenient to convert formulas into such form with all the quantifiers in front (called "prenex normal form"). It makes it easier to see the order of quantifiers and perform operations such as negation.

So in a first-order formula we can rename a variable (to a variable name which does not occur in the subformula where we are making the change, obviously), and move the quantifiers to the front of the formula. What else are we allowed to do? Just as with propositional formulas, we are allowed to replace a subformula with another logically equivalent formula (preserving variable names). For example, if we have a formula $\exists x \forall y (P(y) \vee Q(y)) \wedge \neg P(x)$. then we cannot rename y to x. However, we can rename y to, say, z. Also, we know from DeMorgan's law that $P(x) \vee Q(x) \iff \neg(\neg P(x) \wedge \neg Q(x))$. By changing the variable to y in this equivalence, we can substitute the new subformula $\neg(\neg P(y) \wedge \neg Q(y))$ into the original quantified formula to obtain $\exists x \forall y \neg(\neg P(y) \wedge \neg Q(y)) \wedge \neg P(x)$

11.2 Predicates with finite domains and resolution

Consider the case when domain of a predicate is small (i.e., finite). In this case, it is possible to represent quantifiers using $\vee$ and $\wedge$, thus reducing a first-order case to a propositional case.

Example 1. Suppose that we consider the relation $Parent(x, y)$ on the domain consisting of 5 people: $\{John, Bob, Mary, George, Alex\}$. Consider a formula $\forall x \exists y Parent(x, y)$, saying now that each one of these 5 people has another of these 5 as a child (which is not possible, for a domain like that). Or, alternatively, consider a relation $\forall x \forall y \exists z Parent(z, x) \wedge Parent(z, y)$, saying that all people in this list are siblings.

Let's look at the first one of these relation. To save space, I will just write $Parent(x, y)$ as $P(x, y)$ here.

What does it mean that $\forall x A(x)$ some formula $A(x)$ is true? In the case of $A(x)$ being $\exists y P(x, y)$, it means that $A(x)$ is true for John and Bob and Mary and George and Alex. So we can write this as

$$(\exists y P(John, y)) \wedge (\exists y P(Bob, y)) \wedge (\exists y P(Mary, y)) \wedge (\exists y P(George, y)) \wedge (\exists y P(Alex, y))$$

Similarly, what does it mean for an existential quantifier to be true? In this case, the formula is true either for John, or for Bob, or for Mary and so on. So the formula becomes

$$\begin{aligned}
&(P(John, John) \vee P(John, Bob) \vee P(John, Mary) \vee P(John, George) \vee P(John, Alex)) \\
\wedge &(P(Bob, John) \vee P(Bob, Bob) \vee P(Bob, Mary) \vee P(Bob, George) \vee P(Bob, Alex)) \\
\wedge &(P(Mary, John) \vee P(Mary, Bob) \vee P(Mary, Mary) \vee P(Mary, George) \vee P(Mary, Alex)) \\
\wedge &(P(George, John) \vee P(George, Bob) \vee P(George, Mary) \vee P(George, George) \vee P(George, Alex)) \\
\wedge &(P(Alex, John) \vee P(Alex, Bob) \vee P(Alex, Mary) \vee P(Alex, George) \vee P(Alex, Alex))
\end{aligned}$$

Now, notice that there are no more free variables in predicates. So in effect they are not predicates anymore, but propositional variables! We can use, say, a variable $p_{m,b}$ to mean $P(Mary, Bob)$ is true, and same for the rest of the occurrences of $P()$. Now, we can write the formula above as a truly propositional formula $((p_{jj} \vee p_{jb} \vee \cdots \vee p_{ja}) \wedge \ldots$ Note that once we got this kind of formula, we can apply resolution to check whether it is a tautology/contradiction.

Example 2. For a more natural example consider the following formula $\exists y,\ 0 \leq y \leq 1\ \forall x,\ 2 \leq x \leq 4\ (y + 1 < x)$ Here, we include the description of the domain into the quantifier. For this example, suppose also that $x, y \in \mathcal{N}$. This formula has the same meaning as $\exists y\ (0 \leq y \leq 1) \wedge (\forall (2 \leq x \leq 4 \rightarrow\ y + 1 < x))$, but here we want to treat these restrictions as restrictions of the domain of the quantifiers.

In this case we have two possible values for y, $y = 0$ and $y = 1$, and three possible values for x, $2, 3$ and 4. After the same transformation as in the previous example, computing $y + 1$ for each y, we obtain the following formula:

$$(1 < 2 \wedge\ 1 < 3 \wedge\ 1 < 4) \vee (2 < 2\ \wedge 2 < 3\ \wedge 2 < 4)$$

Now, this is a propositional formula where we know the meanings of propositions (here, our propositions are $1 < 2$, $2 < 4$ and so on) so we can figure out its truth value. The only unequality here that is false is $2 < 2$, since we took the "strictly greater" relation $<$ here. This makes the subformula $(2 < 2 \ \wedge 2 < 3 \ \wedge 2 < 4)$ false. However, the subformula in the first set of parentheses is true, therefore the whole formula is true.

When we have many formulas $\vee$ or $\wedge$ together, it is often convenient to think about it as just one big $\vee$ or big $\wedge$ operator over many inputs (we can do this because of the associative logic identity). For example, the formula above can be written as $\bigvee_{0 \leq y \leq 1} \bigwedge_{2 \leq x \leq 4} y + 1 < x$. Even more useful this concept becomes when we talk about circuits. In circuits, it is possible to feed more than two wires into an AND gate, or an OR gate, essentially emulating these big $\vee$ and $\wedge$. This is how quantifiers (on finite domains) are represented using circuits.

11.3 Empty set and quantifiers

Recall that an empty set, $\emptyset$, is a set containing no elements. What happens when the empty set is our domain? Then if our quantifier is the universal quantifier, then the formula is always true! For example, $\forall x Parent(x, x)$ is true, as well as $\forall x \forall y Parent(x, y) \wedge Parent(y, x)$. Why would such a strange thing happen?

Remember that one way to talk about domains is to put an implication that if the x is in the domain, then the formula under quantifier is true. That is, $\forall x A(x)$ can be stated as $\forall x(x \in domain \to A(x))$. But note that if $domain = \emptyset$ then the left side of the implication is always false. Therefore, the whole formula is true: for every x, since $x \in domain$ is false, $x \in domain \to A(x)$ is true. Note that from here you can also see that when an existential quantifier has empty domain, the formula is always false: there is no element in the domain that could be used to witness the quantifier. One way of explaining it is to say that an existential quantifier is a negated universal quantifier, so if the universal is always true, then the existential is always false.

So what happens with big $\vee$ and big $\wedge$? Remember the logic identities $T \wedge p \iff p$, $F \vee p \iff p$? You can see this as saying that an "empty $\wedge$" is true, and an "empty $\vee$" is false. This agrees exactly with the fact that a formula with $\forall$ converted to $\wedge$ and $\exists$ converted to $\vee$ will be true when there are no predicates on $\wedge$ and false when there are no predicates on $\vee$.

This brings us to the puzzle from last class, about "is there a person such that when this person drinks, everybody drinks". Yes, and you might know such people: those are the people who never drink. So the domain of the times they are drinking is an empty set.