CS 2800: Logic and Computation Fall 2010 (Lecture 13)

13 October 2010

1 An Introduction to First-order Logic

In *Propositional*(Boolean) *Logic*, we used large portions of mathematical language, namely those parts that can have a truth value. But this is not powerful enough to express the following simple argument:
`All squares are positive, 16 is a square, therefore 16 is positive`.
We know this is a valid claim, but if we try to formalize it in *Propositional Logic*, we have $p \wedge q \rightarrow r$ and where p stands for *"All squares are positive"*, q stands for *"16 is a square"* and r stands *"16 is positive"*, but it seems we can easily falsify this formula. Whats happening? Note that *Propositional Logic*, henceforth referred to as PL, is not expressive enough to capture the "relationship" between p and q(think about this relationship). Now note that `All squares are positive` is a statement that expresses an infinite number of propositional formulas at once:
1 is positive and 4 is positive and 9 is positive and 16 is positive and ...
which can be formalized in PL as

$$p_1 \wedge p_4 \wedge p_9 \wedge p_{16} \wedge p_{25} \wedge p_{36} \ldots$$

where p_i stands for *"i is positive"*. But wait, formulas can only be finite in size, you cannot have an infinite number of propositional variables appearing in a formula. Bottom line is we have to extend the language, to talk about objects and relations(also called predicates or sometimes properties) among these objects. Particularly we want to be able to talk about *all* objects of the domain of interest, e.g. `all even numbers are a sum of two odd` `primes`. Dually we want to express "there exists an object such that ...", e.g. in `there exists a real number whose square is 2`. This extension of PL , is called *First Order Logic*(or *Predicate Logic*) and henceforth referred to as FOL .

1.1 An informal discussion of $\forall$

Consider a small mathematical statement I might make:

$$2x \geq x$$

Is this true? You might say, yes, it is true, but its truth value depends on what x can be, i.e. the meaning of the symbol `x`. If x can be a negative number, this statement is not true. In this sense, mathematicians are rather sloppy: there are often unwritten assumptions in the statements they make. They can say, "You know what I mean," and they are often right. Computers, however, only understand what we tell them, so we need to be more precise. We will use FOL to make it more precise. First of all, we should write

$$\forall x \quad 2x \geq x$$

This reads as, `For all x, two times x is greater than or equal to x`. More precisely, this means, `For any value assigned to the variable x, two times x is greater than or equal to x`. This $\forall x$ is also known as universal quantification. But this is not quite what we want because for the whole statement to be true, the part inside the "for all" has to be true for any value of x in our domain, which is the whole universe (the meaning of $\forall$ will be mentioned more precisely later). Lets assume the universe contains negative numbers, then,

$$x = -1$$

is a **counterexample** to the above, because it demonstrates how it can be false. To make a true statement, we probably want to restrict this to natural numbers. Well, there is only one kind of universal quantification, but we have studied something in boolean logic that allows us to state that something must be true when something else is true. What we want to say here is, "for all x, *if* x is a natural number, *then* $2x \geq x$". What we need is logical implication:

$$\forall x \quad x \in \mathbb{N} \rightarrow 2x \geq x$$

$x \in \mathbb{N}$ is the way mathematicians say that x is a natural number (more precisely, value of x is in the set of natural numbers). The ACL2 equivalent would be `(natp x)`. Recall that if the hypothesis of an implication (here $x \in \mathbb{N}$) is false, the formula is true. Thus, the statement inside the "for all" is true for all x values that are not natural numbers. For all natural numbers x, the body of the for all is true exactly when $2x \geq x$ is true. (Recall that *false* $\rightarrow p$ is equivalent to true and *true* $\rightarrow p$ is equivalent to p.) To be precise, under standard theory of arithmetic, $2x \geq x$ is true for all natural numbers. That makes this statement a **theorem** in the **theory** of standard arithmetic. But this statement is not a **tautology**, why this is so, will be apparent soon.

In first order logic, we have one other kind of quantifier, the existential quantifier, $\exists$. e.g.

$$\exists x \ (x \in \mathbb{Z}) \wedge (x < 0)$$

$x \in \mathbb{Z}$ means x is an integer, so the whole thing would read as, `There exists an x for which x is in the set of integers and x is less than zero`. Is it true if we can give x some value which is both an integer and less than zero? Yes, we can. $x = -1$ is a **witness** that shows the existential is true in standard arithmetic. Thus, the formula is *true*(a theorem).

1.2 FOL has Relations and Quantifiers

Most mathematical statements are of the form `a has the property P`, or `x and y are in the relation R`, etc. e.g. "n is even", "$3 = 4$", "$7 \leq 11$" , "path(p,A,B)". So we build our language from symbols for *predicates*(also called relations or properties) and individual *objects*. Furthurmore, we add *variables* to range over these objects, and the usual propositional connectives plus the quantifers $\forall$("for all") and $\exists$("there exists").

Kinds of informal mathematical statements we can formalize in FOL :

1. Every natural number is even or odd –
   $\forall x \ x \in \mathbb{N} \rightarrow (evenx) \vee (oddx)$
2. Between any two rational numbers, there is another –
   $(\forall x(\forall y \ x, y \in \mathbb{Q} \wedge x < y \rightarrow (\exists z \ x < z \wedge z < y)))$
3. If there is a path from A to B, there is a path from B to A –
   $(\exists p \ path(p, A, B)) \rightarrow (\exists q \ path(q, B, A))$

We say, $\forall x \exists y (x = 2y)$ as: `for all x, there is a y such that x is two times y`.

Of course, we can use first order logic to model parts of nonmathematical discourse as well, such as the following:

1. `Every politician is corrupt`
2. `Not every politician is corrupt`
3. `Every politician that is corrupt gets caught`
4. `Some corrupt politicians do not get caught`
5. `Any politician is either corrupt and is caught, or is not corrupt`

Every use of the word "all" or "some" ranges over some domain, which can be explicit or implicit. If I say "everything is greater than or equal to 0," I may be referring to the set natural numbers implicitly. If I say "every natural number is greater than or equal to 0," I am explicitly using the word "every" to range over natural numbers. To handle cases where the explicit information is absent, we will assume that for any given statement there is some implicit universe of objects under consideration. As in the case of propositional logic, we would like to focus on the meaning of the logical elements of a given sentence, without worrying about the meaning of the basic terms. For instance, we would like to say that statement 4 is logically equivalent to the negation of statement 5 (interpreting some to mean at least one), independent of how we interpret the terms corrupt or politician, and independent of the worlds political status at any given moment.

So basically the building blocks of the language of FOL are the function and relation symbols. Intuitively, there is some universe or domain that we would like to use with this language: this can be the set of natural numbers, the set of politicians, a set of colored blocks, or whatever. The function symbols are meant to denote functions on the domain of discourse(interest), such as the multiplication function($2x$ is $2 * x$) on the natural numbers, or the biggest political opponent of function on the set of politicians. The relation/predicate symbols are meant to denote properties(relationships) that may or may not hold of members of the universe: e.g. *evenp* or $\geq$ on the set of natural numbers, or "is more corrupt than" on the set of politicians.

The phrase "for all" is called the *universal quantifier* and is denoted symbolically by $\forall$. The phrases "there exists", "there is a", and "for some" all have the same meaning: "there exists" is called the *existential quantifier* and is denoted symbolically by $\exists$. The universal quantifier is like an iterated conjunction and the existential quantifier is like an iterated disjunction. To understand this, suppose that there are only finitely many objects; that is the variable x takes on only values $a_1, a_2, \ldots, a_n$. Then the sentence $\forall x P(x)$ means the same as the sentence $P(a_1) \wedge P(a_2) \wedge \ldots \wedge P(a_n)$ and the sentence $\exists x P(x)$ means the same as the sentence: $P(a_1) \vee P(a_2) \vee \ldots \vee P(a_n)$

Of course, if the number of individuals is infinite, such a meaning of quantifiers is not rigorously possible since infinitely long formulas are not allowed. The similarity between $\forall$ and $\wedge$ and between $\exists$ and $\vee$ suggests many equalities. For example, De Morgans equalities

$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$

$$\neg(p \wedge q) \equiv [\neg p \vee \neg q]$$

have the following "infinite" versions

$$\neg \exists x P(x) \equiv \forall x \neg P(x)$$

and

$$\neg \forall x P(x) \equiv \exists x \neg P(x)$$

.

2 Syntax: What do first-order formulas look like?

Just like we defined the structure of PL formulas, and we said that $\wedge p \to q$ is not a well-formed propositional formula, i.e. it was not formed by using the formation rules that I showed in class, similarily we will define the structure of FOL formulas. Recall that formulas are just a bunch of symbols, wether a sequence of symbols is a formula or not, is determined by the syntax formation rules(similar to data definition). Before we give the data definition of first-order formulas, we define what terms(also called expressions) are, since terms denote objects(from a chosen universe). From now on I will use the word "term" and "expression" interchangeable, but when I relate FOL to ACL2 Logic, I will use "expression" more often. Let A be set of symbols allowed in FOL , i.e. we can only use symbols from this set to construct sequences of symbols(strings) of interest, mainly terms and formulas.

We (inductively) define a term as follows:

- Every Variable symbol in A is a term
- Every Constant symbol in A is a term
- If e_i is a term, and f is a function symbol of arity n in A, then $f(e_1, e_2, \ldots, e_n)$ is also a term.

Here is the same in the familiar data definition format:

$$expr := x \mid c \mid f(e_1, e_2, \ldots, e_n)$$

Note the similarity with expressions in ACL2 language we studied in Lecture 5. Now that we can talk about objects(using *terms*), we want to be able to talk about properties of these objects, we will do that using first-order *formulas*.

Assuming e_i is a term as defined above, we define FOL formulas:

- $e_1 = e_2$ is a (atomic) formula
- If R is a relation symbol in A of arity n, then $R(e_1, \ldots, e_n)$ is a (atomic) formula.
- If ϕ and ψ are formulas, and $\Box$ is one of $\wedge, \vee, \rightarrow, \equiv$, then $(f_1 \Box f_2)$ is also a formula.
- If ϕ is a formula then so are $(\neg\phi)$, $(\forall x\phi)$, $(\exists x\phi)$

ϕ, ψ are greek letters, normally used to represent formulas, you can write their english pronunciations, if you want, which are *phi* and *psi*.

We will introduce some abbreviations. Just like in PL , we drop parantheses when its unambiguous to do so. e.g. we always drop the outermost parantheses for the quantifiers. Here is the precedence order of connectives: $\neg > \forall, \exists > \wedge > \vee > \rightarrow > \equiv$, so the quantifiers bind more strongly that the binary connectives. For better readability, we will sometimes separate the quantifier and the formula by a dot: $\forall x.\phi$.

Note: In sentences of form $\forall x P(x)$ or $\exists x P(x)$ the variable x is called a dummy variable or a **bound variable**. This means that the meaning of the sentence is unchanged if the variable x is replaced everywhere by another variable. Thus

$$\forall x P(x) \equiv \forall w P(w)$$

and

$$\exists x P(x) \equiv \exists z P(z)$$

For example, the sentence "there is an x satisfying $x + 7 = 5$" has exactly the same meaning as the sentence "there is a z satisfying $z + 7 = 5$". We say that the second sentence arises from the first by alphabetic change of a bound variable.

Terminology: In $(\forall x \phi)$, ϕ is the **scope** of the quantifier. If a term, (sub)formula occurs in ϕ, we say it is in the scope of the quantifier in $\forall x \phi$ or $\exists x \phi$. Recall x is called a bound variable. Any other variables occuring in ϕ are called free variables with respect to that scope and are said to **occur freely** in that scope.

- Any variable occuring freely in the scope of the whole formula is one of its **free variables**.
- A first-order formula with no free variables is called a **proposition**
- A proposition that one suspects is true is called a **conjecture**.

3 Semantics: What do the well-formed formulas mean?

Remember Remember the 5th of Nov... Remember that the formal statements(formulas) shown above are just sequences of symbols. We have attached some intuitive meaning to some of them, e.g. arithmetic statements, but we havent explained where this "meaning" came from! The computer is a (very fast) idiot, it has to be told everything.

Also recall that it makes no sense to ask the meaning of an ill-formed sequence of symbols, only after we have determined that a sequence of symbols is a formula, can we ask, what is its meaning!

If I ask you wether a formula $p \vee q \rightarrow r$ was *true*, you would say, "I can answer that question, if you give me the truth values assigned to p, q, r". In the same way, it makes little sense to ask wether a formula $\forall x \exists y R(x, y)$ is

true, without a context(a history). You should ask, what does the symbol R mean, what "universe" of objects are the quantifiers ranging over. Note since there are no free variables, we need not be ask for the variable assignment. But for now we will restrict our attention to propositions only, i.e. formulas with no free variables.
A **theory** constrains the meaning of constants, function symbols and relation/predicate symbols.
e.g. The standard theory of arithmetic(naturals), will say, the symbol "0" means the number `0`, symbol "<" means the less than relation, the universe is $\mathbb{N}$, the function symbol "Succ" means the successor function, the symbol "+" denotes the function `add`, the symbol "*" denotes the `mult` function etc. We normally specify this mapping from relation/function symbols to relations/functions using a (theory) interpretation J. e.g. $J_{arith}(*) = \texttt{mult}$

But the meaning of the logical connectives, the quantifiers and the symbol "=" is given solely by FOL i.e. the **logic** as we see below.

Let U be the Universe, let J be the mapping given by the theory, that is, the meaning of symbol R is given by the predicate $J(R)$, then the meaning of FOL propositions(no free vars) is given by:

- Value of $e_1 = e_2$ is *true* :iff Val(e_1) is same as Val(e_2)
- Value of $R(e_1, \ldots, e_n)$ is *true* :iff Val($J(R)(c_1, \ldots, c_n)$) is *true*
- Meaning of propositional connectives is like in PL
- Value of $\forall x \phi$ is *true* :iff for all $x \in U$, Val($\phi(x)$) is *true*
- Value if $\exists x \phi$ is *true* :iff for some $x \in U$, Val($\phi(x)$) is *true*

Note that Val($\phi(x)$) means find the truth value of ϕ under the assignment made to x. e.g. Value of $\exists x.x < 9$ is *true* since for assignment $x := 7$, the Value of $x < 9$ is the Value of $7 < 9$, which under the standard arithmetic theory is *true*. Let's get back to how this relates to theorems and tautologies.

$$3 = 3 \wedge 4 = 4$$

is a tautology because equality and conjuction ($\wedge$) are built into FOL , so no matter what a particular theory says about the constants 3 and 4, this formula is true. Similarly,

$$\forall y.\ y = y$$

is a tautology, because the basic meaning of quantification, equality, and variable reference are built into the logic. The theory can constrain what values constitute "all values", but in FOL , they are all equal to themselves. However,

$$3 < 4$$

is not a tautology. It is a theorem in standard theories of arithmetic, but one could construct a theories in which this is always false, or even unknown. Thus, it is not a tautology.

Terminology:

- A proposition which has a *true* value under a particular theory is a **theorem**
- A formula which has a *true* value for all theories is a **tautology**

4 Formalizing a simple conjecure in FOL

Let us try to write a logical formula for something we know from mathematics: `There is no largest integer`. How do I write that as a formula? Well, let's rephrase it slightly: `There does not exist an integer that is the largest`. We know how to write `There does not exist ...`:

$$\neg \exists x \ldots$$

Now we need a formula that says x is an integer and x is largest. Well, we've got half of it:

$$\neg \exists x . x \in \mathbb{Z} \ldots$$

For x to be largest, it has to be greater than or equal to every other integer. How do I write a formula that must be true for lots of cases, even an infinite number of cases? (Universal quantification)

$$\neg \exists x . x \in \mathbb{Z} \wedge (\forall y . y \in \mathbb{Z} \rightarrow x \geq y)$$

That says that, `there does not exist an x that is both an integer and, for all y that are integers, x is greater than or equal to y`.

We can use the identities on quantifiers along with our boolean logic identities to change this into equivalent forms:

$$\forall x.\neg(x \in \mathbb{Z} \wedge (\forall y.y \in \mathbb{Z} \rightarrow x \geq y))$$

$\equiv$ {Demorgans Equality}

$$\forall x.\neg x \in \mathbb{Z} \vee \neg(\forall y.y \in \mathbb{Z} \rightarrow x \geq y)$$

$\equiv \{\neg p \vee q \equiv p \rightarrow q\}$

$$\forall x.x \in \mathbb{Z} \rightarrow \neg(\forall y.y \in \mathbb{Z} \rightarrow x \geq y)$$

$\equiv$ {Demorgan version for $\forall$}

$$\forall x.x \in \mathbb{Z} \rightarrow \exists y.\neg(y \in \mathbb{Z} \rightarrow x \geq y)$$

$\equiv$ {Demorgan and $p \rightarrow q \equiv \neg p \vee q$}

$$\forall x.x \in \mathbb{Z} \rightarrow \exists y.y \in \mathbb{Z} \wedge \neg(x \geq y)$$

Now, assuming we are in a standard theory of arithmetic, we can use(instantiate) the theorem that:

$$\neg(x \geq y) \equiv y > x$$

to finally get

$$\forall x.x \in \mathbb{Z} \rightarrow \exists y.y \in \mathbb{Z} \wedge y > x$$

This is formula is rather simple to read and if we think about its meaning, it seems it should be equivalent to our original statement of the proposition: `For every integer, there exists an integer that is larger`.

We aren't yet going into the details of proof, but how would we prove this in mathematics? Well, to prove an existential, we need a witness. In this case, the existential (on y) is inside a universal quantifier (on x). Thus, we need a witness for each x for which the existential must be true. Well, the formula in the universal quantifier is an implication, $x \in \mathbb{Z} \rightarrow \exists y \ldots$. Thus, if x is not an integer, then the body is true regardless of the existential formula. But if x is an integer, then the existential formula must be true for the body to be true. Thus, we need to find witnesses to the existential for each possible integer value of x. If x is an integer, can you name an integer greater than it? How about $x+1$? (Yes, it works.) In fact, I can eliminate the existential quantifier by plugging in $x+1$ for each y under it in the formula:

$$\forall x.x \in \mathbb{Z} \rightarrow (x+1 \in \mathbb{Z} \wedge x+1 > x)$$

Our knowledge of mathematics tells us that this is a theorem in standard theories of arithmetic. (An integer plus one is also an integer and greater than the original integer.)

In the next class we will see the connection to ACL2 Logic. In fact they are very similar.

References: Peter Dillinger's notes, van Dalen(Logic and Structure), and Jeremy Avigad's logic notes.