COSC410 Logic for AI

First-Order Predicate Calculus

Richard A. O'Keefe

17 May 2016

Key Topics

- Things we can't say in propositional calculus
- Predicates
- Functions
- Variables
- Quantifiers
- Semantics
- Resolution
- Unification
- Decidability
- Interesting/Useful results

Things we can't say: splitting atoms

- Suppose we want to say "Abigail's father is Richard" and "Lily's father is Richard".
- If we want to model these statements in propositional calculus, we have to use *different* symbols, because they *mean different things*. (Abigail is in fact my daughter, but Lily is a fox terrier.)
- But atomic sentences in propositional calculus have no parts in common, because they have no parts.
- father(Abigail, Richard) $\wedge$ father(Lily, Richard).

Atomic sentences with parts, BUT

Something like father(Abigail, Richard) does have parts, but it has no parts that are *sentences*, so *as a sentence* it is atomic.

What are these parts?

- There are parts like Abigail, Lily, Richard, that stand for *things* in some *universe of discourse*. They are somewhat like nouns. These parts are called *terms*. (We'll see how terms are made shortly.)
- There are parts like father(_,_) and living(_) that stand for *relations* about things in the universe of discourse. They are somewhat like (stative) verbs. These parts are called *predicates*.
- Applying a predicate to enough terms gives us an *atomic sentence*.

Talking about arithmetic

- We want to be able to talk about $1 + 2$ and 4×5 and the like. 1, 2, 4, 5 are no trouble. They are names for things in our universe of discourse, just like Abigail, Lily, and Richard.
- But what about $\_ + \_$ and $\_ \times \_$? They aren't truth values or predicates and they don't stand for specific numbers. The things that these symbols stand for are *functions* that take numbers as arguments and give us other numbers as results. We'll call the symbols *function symbols* and say that a function symbol can be combined with zero or more terms to give us a term.

Constants and functions

- A *function symbol* f of arity n stands for a function $\hat{f} : \mathcal{U}^n \to \mathcal{U}$
- but we don't know *which* function until someone tells us.
- A *constant symbol* is just a function symbol of arity 0.

Things we can't say: incomplete information

- We can't say "someone is Lily's father" so far, we can only say that some specific dog is her father.
- We can't say "27 has a prime factor" without exhibiting one.
- We can't say "all men are mortal", only that if some specific thing is a man then that thing is mortal.
- We *can* distinguish between saying "¬father(Lily, Richard)" and failing to say that he is.

Mathematical variables

$$\sum_{x \in S} f(x) \qquad \prod_{x \in S} f(x)$$

$$\int_a^b f(x)\, dx \qquad \bigcup_{x \in S} f(x)$$

$$\lim_{x \to 0} \frac{sin(x)}{x} \qquad \lambda x . f(x)$$

Mathematical variables 2

- there are *variables*
- that are *bound* by an operator
- and can be *renamed*
- meaning comes from the operator not the variable

Classical syllogisms

1. **all** men are mortal
2. **some** mortals die
3. ∴ **some** men die

1. **no** fish is human
2. **some** animals are fish
3. ∴ **notall** animals are human

Quantifiers!

- The linguistic name for words like “some”, “all”, “every”, “no”, “many”, “most”, “five”, is *quantifier*.
- So we'll call our binding operators *quantifiers*.
- What should they look like? (doesn't really matter)
- What should they mean? (vital)

An example

1. **all** unicorns are horn-bearers
2. **all** horn-bearers are dangerous
3. $\therefore$ **some** unicorns are dangerous

This is a valid inference in Aristotelian logic, but how can it be when there are no unicorns?

Forall and exists

- $(\forall x)p(x)$
  True iff every $x \in \mathcal{U}$ satisfies $p(x)$.
  False iff some $x \in \mathcal{U}$ falsifies $p(x)$.
- $(\exists x)p(x)$
  True iff some $x \in \mathcal{U}$ satisfies $p(x)$.
  False iff every $x \in \mathcal{U}$ falsifies $p(x)$.
- $(\exists x)p(x) \equiv \neg(\forall x)\neg p(x)$
- $(\forall x)p(x) \equiv \neg(\exists x)\neg p(x)$
- "infinite $\wedge$" and "infinite $\vee$"

Other quantifiers are possible

- $(!x)p(x)$ = there is at most one x making $p(x)$ true
- $(\exists!x)p(x)$ = there is exactly one such x
- $(\forall_1 x)p(x) = ((\forall x)p(x)) \wedge ((\exists x)p(x))$
- Some strictly increase the power of the logic, given equality, these don't.

A semantic view

- Think of $\{x \in \mathcal{U} | p(x)\}$.
- A quantifier makes an assertion about this set.
- $\forall$: it's $\mathcal{U}$
- $\exists$: it isn't empty
- $\not\forall$: it's not $\mathcal{U}$
- $\not\exists$: it is empty
- $\exists!$: it has exactly one element

Syntax of terms

- if x is a variable, x is a term.
- if f is a function symbol of arity n and t_1, ..., t_n are terms, then $f(t_1, \ldots, t_n)$ is a term.
- There are no other terms.
- Writing a function infix makes no difference: $x + 1$ and $+(x, 1)$ are the same term.

Semantics of terms

- We need to know what the *universe* $\mathcal{U}$ is.
- We need an *interpretation* I of the function symbols, such that if f is a function symbol of arity n, $I(f) : \mathcal{U}^n \to \mathcal{U}$.
- We need an *assignment* V mapping variables to $\mathcal{U}$.
- I'll use $\mathcal{M}(\_)$ abbreviating $\mathcal{M}[I, V](\_)$.
- $\mathcal{M}(x) = V(x)$
- $\mathcal{M}(f(t_1, \ldots t_n)) = I(f)(\mathcal{M}(t_1), \ldots, \mathcal{M}(t_n))$

Syntax of sentences

- if p is a predicate symbol of arity n and $t_1, \ldots,$ t_n are terms, then $p(t_1, \ldots, t_n)$ is a(n atomic) sentence. There are no other atomic sentences.
- Infix is OK; $x > 0$ and $>(x, 0)$ are the same.
- if ϕ and χ are sentences, then $\neg\phi$, $\phi \wedge \chi$, $\phi \vee \chi$, $\phi \rightarrow \chi$, $\phi \leftrightarrow \chi$ are sentences.
- We could also allow other propositional connectives.
- if x is a variable and ϕ is a sentence, then $(\forall x)\phi$ and $(\exists x)\phi$ are sentences.
- We could also allow $\forall_1$ and (if $=$ is part of our logic) ! and $\exists!$.

Semantics of sentences

- Our *interpretation* must also interpret the predicate symbols, such that if p is a predicate symbol of arity n, and $B = \{\bot, \top\}$, $I(p) : \mathcal{U}^n \to B$.
- $\mathcal{M}(p(t_1, \ldots, t_n)) = I(p)(\mathcal{M}(t_1), \ldots, \mathcal{M}(t_n))$
- $\mathcal{M}(\neg\phi) = \neg\mathcal{M}(\phi)$
- $\mathcal{M}(\phi \wedge \chi) = \mathcal{M}(\phi) \wedge \mathcal{M}(\chi)$
- $\vee$, $\rightarrow$, $\leftrightarrow$ similar. On the left, the operator is part of the syntax, on the right, it's a truth function.

Semantics of quantified sentences

- Define $(x \mapsto v)f$ to be $\lambda y.\text{if } y = x \text{ then } v \text{ else } f(y)$. That is, $(x \mapsto v)f$ agrees with f except at x, where it's v.
- $\mathcal{M}[I, V]((\forall x)\phi) =$ for all $u \in \mathcal{U}$ $\mathcal{M}[I, (x \mapsto u)V](\phi)$.
- $\mathcal{M}[I, V]((\exists x)\phi) =$ there is some $u \in \mathcal{U}$ such that $\mathcal{M}[I, (x \mapsto u)V](\phi)$

But what about the unicorns?

1. $(\forall x)u(x) \rightarrow h(x)$
2. $(\forall x)h(x) \rightarrow d(x)$
3. $\therefore$ $(\forall x)u(x) \rightarrow d(x)$ does follow, but $(\exists x)u(x) \wedge d(x)$ does not.

Free, bound, open, closed

- An occurrence of a variable is *bound* iff it is in the scope of a quantifier.
- We say the variable occurrence right after $\forall$ or $\exists$ is a *binding* occurrence.
- An occurrence of a variable is *free* iff it is not in the scope of any quantifier.
- $p(x) \wedge (\exists x)q(x)$ contains a free occurrence, a binding occurrence, and a bound one, in that order.
- A sentence is *open* iff it contains at least one free variable occurrence.
- A sentence if *closed* iff every variable occurrence is a binding or bound one.

Satisfaction and validity

- Given an interpretation I, an assignment V *satisfies* ϕ iff $\mathcal{M}(\phi)$ is true.
- ϕ is *satisfiable* under I iff there is some assignment V that satisfies ϕ.
- ϕ is *satisfiable* iff ϕ is satisfiable under some I.
- ϕ is *valid* under I iff every assignment V satisfies ϕ.
- I is then said to be a *model* of ϕ.
- ϕ is *valid* iff it is valid under every I.

Prenex Normal Form

A formula is a *prenex formula* (or *in prenex form*) iff it is of the form $Q_1 x_1 \ldots Q_n x_n B$ where B is a formula with no quantifiers, $n \geq 0$, $x_1, \ldots, x_n$ are variables, and $Q_i \in \{\forall, \exists\}$ for $1 \leq i \leq n$.
It helps to replace $p \leftrightarrow q$ by $(p \rightarrow q) \wedge (q \rightarrow p)$.

Getting there, 1

- An atomic sentence is in prenex form.
- $(\forall x)\phi$ and $(\exists x)\phi$ simply add one more quantifier on the left of the prenex form of ϕ.
- We want to "bubble quantifiers up" over connectives.
- For $\neg\phi$, let $Q_1 x_1 \ldots Q_n x_n B$ be the prenex form of ϕ. Define $\overline{\forall} = \exists$ and $\overline{\exists} = \forall$. Then $\overline{Q}_1 x_1 \ldots \overline{Q}_n x_n \neg B$ is the prenex form of $\neg\phi$.
- $\wedge$ and $\vee$ are left as an exercise.

Getting there, 2

- $((\exists x)A) \to B \equiv (\forall x)(A \to B)$ if x is not free in B; if it is, rename x first.
- $A \to ((\exists x)B) \equiv (\exists x)(A \to B)$ if x is not free in A; if it is, rename x first.
- $((\forall x)A) \to B \equiv (\exists x)(A \to B)$ if x is not free in B; if it is, rename x first.
- $A \to ((\forall x)B) \equiv (\forall x)(A \to B)$ if x is not free in A; if it is, rename x first.

Skolem functions

Warning: this makes sense, and I got used to it pretty quickly, but it's not like the other syntax-driven things we've done.

It turns out to be simpler to only have to deal with one kind of quantifier. If you've heard of the axiom of choice, it's rather like that. Suppose we have $(\forall x)(\exists y)x > y$. If that's true, then for any x, there is at least one y such that $x > y$, and we can let $s(x)$ be any such y. So we can say $(\forall x)x > s(x)$.

The arguments of a Skolem function are the universally quantified variables enclosing the existential quantification it replaces.

Clausal form

- A first-order literal is either an atomic sentence or the negation of an atomic sentence.
- A first-order clause is a set of first-order literals.
- A first-order sentence in clausal form is a set of clauses.
- Convert to prenex form, Skolemise, then convert to clausal form just like propositional calculus.
- All remaining variables are universally quantified.

Propositional resolution

Resolution is the inference rule
from $A \cup \{p\}$ and $B \cup \{\neg p\}$ infer $A \cup B$.
It generalises $(p \to q) \wedge p \to q$.
We could use this on first-order clauses, if it weren't for the variables.

Unification

- Given two atomic sentences, we want to see if we can find values for variables that make them the same.
- Take $p(t_1, \ldots, t_m)$ and $q(u_1, \ldots, u_n)$. If $p \neq q$ or $m \neq n$, those things are not variables, so we can't make them the same, fail. Otherwise, solve $\{t_1 = u_1, \ldots, t_n = u_n\}$.
- The empty set of equations is solved.
- Otherwise, remove any $t = u$ from the set.

Unification 2

- If t and u are both variables, if they are not the same, replace one by the other.
- If t is a variable and u is not, fail if t occurs inside u, otherwise replace t by u everywhere.
- If u is a variable and t isn't, similar.
- If $t = f(v_1, \ldots, v_m)$ and $u = g(w_1, \ldots, w_n)$, fail if $f \neq g$ or $m \neq n$, otherwise add $\{v_1 = w_1, \ldots, v_n = w_n\}$ to the set of equations.

First-order resolution

- From a clause $A \cup \{p, q\}$ infer $A' \cup \{p'\}$ if you can make $p = q = p'$ by unification. (FACTORING)
- From clauses $A \cup \{p\}$, $B \cup \{\neg q\}$ infer $A' \cup B'$ if you can make $p = q$ by unification. (RESOLUTION)
- If you reach an empty clause you have found a contradiction.
- Any first-order problem that has a (dis)proof has a resolution (dis)proof.
- You need a strategy for deciding *which* factorings and resolutions to do, but successful resolution provers exist.

Completeness and Decidability

If a formula in first-order logic is valid, it has a finite proof.
But validity is not decidable. Given a falsifiable formula, a proof procedure may loop forever.

Model existence

A set of formulas Γ is *consistent* if there is a formula A such that $C_1, \ldots, C_m \to A$ is *not* provable for any $C_1, \ldots, C_m \subseteq \Gamma$.
The basic idea here is that you can prove anything from a falsehood, so if you can't prove everything, you're not crazy.
If a set of formulas is consistent, it is satisfiable.
If a set of formulas is satisfiable, it is consistent.

Löwenheim-Skolem theorem

If a first-order sentence is satisfiable in some interpretation, then it is satisfiable in an interpretation where $\mathcal{U}$ is at most countably infinite. That is to say, $\mathcal{U}$ has no more elements than there are natural numbers. In particular, there can be no first-order sentence that is true of the real numbers and false of anything with smaller cardinality.

Herbrand Universe

- If a formula has no constant symbols, add one.
- The Herbrand Universe $\mathcal{H}$ is the set of terms you get by pasting functions together in all possible ways, with no variables.
- If a formula in clausal form has any model, it has a model where $\mathcal{U} = \mathcal{H}$.
- This is exploited by theorem provers, Prolog, Datalog.

What we still cannot say

$(\forall p)(p(0) \wedge (\forall n)p(n) \rightarrow p(n+1)) \rightarrow (\forall n)p(n)$
In first-order predicate calculus, variables may only stand for elements of $\mathcal{U}$, **not** for functions or predicates.
In second-order predicate calculus, variables may also stand for (first-order) functions and predicates.
We need this to talk about the real number system **R**.
In higher-order predicate calculus, the arguments of functions and predicates may themselves be functions and predicates.