First Order Logic II

David L. Dill
Department of Computer Science
Stanford University

Announcements

- Office hours are finalized and posted.
- Dill office hours: 3:30 PM - 5:30 PM on Wednesdays (e.g., TODAY).
- We have a Piazza web page for discussion.
  - Try not to give away homework problems.
  - Discussion of certain homework problems on Piazza may be explicitly banned.
- Notes from the (very well attended) section will be posted shortly.
- HW1 In problem 5, conversion to CNF using intermediate variables is a two-phase process. Phase 1 is to convert an arbitrary formula to a conjunction of small formulas of the general form $X_i \leftrightarrow (P_j \oplus P_k)$. Phase 2 is to convert each of the small formulas to a small CNF formula. The first part of the probem asks you to figure out how to do Phase 2 for formulas of the form $X_i \leftrightarrow (P \land Q)$. We assume that, if you can figure out how to do that, you can figure out the other cases. For the last two parts, you can base your answer on Phase 1, only. For the last part, estimate the size to within a small constant factor.

Outline

- Multiple quantifers
- Identities involving quantified formulas
- Formal proofs
  - Simple propositional rules

Multiple Quantifiers

Formulas get more interesting when there are multiple quantifiers.

What does the following mean?

$\exists x\ (\mathrm{Cube}(x) \land \forall y\ (\mathrm{Tet}(y) \rightarrow \mathrm{LeftOf}(x, y)))$

"There is a cube that is to the left of every tetrahedron."

What about:

$\forall x\ (\mathrm{Cube}(x) \rightarrow \forall y\ (\mathrm{Tet}(y) \rightarrow \mathrm{LeftOf}(x, y)))$

"Every cube is to the left of every tetrahedron."

$\forall x\ \forall y\ ((x < y) \lor (y < x))$ is not valid. (Why?)

Consider $x = y$.

DANGER: Do not assume that distinct variables represent distinct individuals.

Sequences of Quantifiers

Swapping two adjacent quantifiers *of the same kind* does not change the meaning of the sentence:

$\forall x\ \forall y\ R(x, y) \equiv \forall y\ \forall x\ R(x, y)$

$\exists x\ \exists y\ R(x, y) \equiv \exists y\ \exists x\ R(x, y)$

However, the order of quantifiers of different kinds is *critical* to the meaning. (These are called *alternating quantifiers*.)

$\forall x\ \exists y\ (y > x)$ "For every number, there is a larger number."

$\exists y\ \forall x\ (y > x)$ "There is a number that is larger than all other numbers."

$\forall x\ \exists y\ (\text{Married}(x, y))$ "Everyone has a spouse."

$\exists y\ \forall x\ (\text{Married}(x, y))$ "There is one person to whom everyone is married."

Statements about numbers of objects

With what we have now, we can make some statements about numbers of objects.

There is at least one cube $\exists x\ \text{Cube}(x)$

There is at most one cube $\exists x\ \forall y\ (\text{Cube}(y) \rightarrow y = x)$ (What if there are no cubes?)

There are at least two cubes $\exists x\ \exists y\ (\text{Cube}(x) \wedge \text{Cube}(y)) \wedge x \neq y$

There are at most two cubes $\exists x\ \exists y\ \forall z\ (\text{Cube}(z) \rightarrow (z = x \vee z = y))$

DeMorgan's Laws for Quantifiers

Recall DeMorgan's Laws

$\neg(a \wedge b) \equiv \neg a \vee \neg b$ $\qquad \neg(a \vee b) \equiv \neg a \wedge \neg b$

And the intuition:

$\forall x\ P(x) = P(0) \wedge P(1) \wedge \ldots \wedge P(n) \wedge \ldots$

$\exists x\ P(x) = P(0) \vee P(1) \vee \ldots \vee P(n) \vee \ldots$

DeMorgan's Law applies to quantifiers as well:

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

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

Aristotelian Forms Revisited

As an example of the previous rules, and other identities you know, here is a nifty relationship among the Aristotelian forms:

$$\begin{aligned} \neg\forall x\ (P(x) \rightarrow Q(x)) &\equiv \exists x\ \neg(P(x) \rightarrow Q(x)) && \text{Quant DeMorgan} \\ &\equiv \exists x\ \neg(\neg P(x) \vee Q(x)) && \text{implies-or} \\ &\equiv \exists x\ (P(x) \wedge \neg Q(x)) && \text{DeMorgan} \end{aligned}$$

A similar relationship holds between

$\neg\exists x\ (P(x) \wedge Q(x)) \equiv \forall x\ (P(x) \rightarrow \neg Q(x))$

Other Identities

There are some very useful distributive laws.

$\forall x\ (P(x) \land Q(x)) \equiv (\forall x\ P(x)) \land (\forall x\ Q(x))$

$\exists x\ (P(x) \lor Q(x)) \equiv (\exists x\ P(x)) \lor (\exists x\ Q(x))$

In general, $\forall$ does *not* distribute over $\lor$ and $\exists$ does *not* distribute over $\land$.

But, if x *does not occur free in* P,

$\forall x\ (P \lor Q(x)) \equiv P \lor (\forall x\ Q(x))$

$\exists x\ (P \land Q(x)) \equiv P \land (\exists x\ Q(x))$

Proofs

Why proofs?

Help prevent errors. Over millenia, we have learned that it is too easy to believe mathematical non-facts, and to make mistakes.

Proving things can help you know whether you can count on them to be true.

Convince others. Proofs are arguments for the correctness of a result. Mathematicians have learned not to believe claims until they see a proof.

Unproved claims are properly called "conjectures." In some cases, a conjecture has stood for over 400 years without being proved or disproved.

Goldbach's conjecture: Every even number greater than 2 is a sum of two primes.

Proofs vs. Boolean algebra

We already have two ways to prove *any* theorem in propositional logic: Truth tables and Boolean algebra.

We're going to learn yet another way of proving propositional theorems: step-by-step deduction. Reasons:

- Sometimes it's more efficient, especially using pencil and paper.
- Sometimes it's more persuasive.
- It generalizes to first-order and other logics.
- It enables the mathematical study of proofs.

Deductive reasoning

Most theorems are of the form Premises $\rightarrow$ Conclusion

The premises are assumptions and is a logical consequence of the premises.

A deductive proof goes from the premises to the conclusion by a series of steps that follow strict rules.

Each rule is guaranteed to produce a logical consequence of previously proved logical consequences of the premises.

A successful proof will eventually produce the desired conclusion of the theorem. Because of the way it was derived, it will be guaranteed to be a logical consequence of the premises.

Formal vs. informal proof

Mathematicians have been writing informal proofs for thousands of years.

In the late 19th and early 20th century, there was an effort to define logic and proofs *precisely*, so there could be no disagreement about the truth of theorems proved with these systems.

The results were systems for *formal proof*. Proofs *about* mathematics became a subject of mathematical study themselves.

Formal proof has been important to computer science, but being able to do formal proofs is not going to be an important skill for most of you.

Therefore, we will talk about formal proof as a way of understanding the requirements of informal proofs.

Fitch – a deductive system

Many systems of formal proof have been invented.

"Natural deduction" is a system that bears a closer resemblance to informal mathematical proof than some systems. We will talk about a "Fitch-style" system.

We will study a system for doing formal proofs, called *System F*.

There is a program called "Fitch" that checks proofs in System F, which you will use to learn it.

The format of a Fitch proof is

premise1	
premise2	
conclusion1	*justification1*
conclusion2	*justification2*

Each conclusion must be justified using rules that are precisely defined as part of System F.

Some simple Boolean Proof Rules

System F is organized into "introduction" and "elimination" rules for each connective.

There are some simple rules in System F.

In an *informal* proof, these would generally go without saying.

- $=$ elimination/introduction
- Conjunction elimination/introduction
- Disjunction elimination
- Negation elimination
- $\bot$ introduction
- Conditional elimination
- Biconditional elimination

Proofs involving =

a = b (identity) is a special predicate that always means that terms a and b name the same individuals.

Principle of undiscernability of identicals: If a statement referring to a holds, the statement continues to hold when b is substituted for some or all occurrences of a.

System F has two rules for identity.

The first comes from reflexivity.

Identity introduction:

⋮	
n = n	*= Intro*

Note that n can be an arbitrary term: E.g., a + b = a + b.

= Elimination

If we know n = m and P(n) (some sentence that may have n)

we can conclude P(m) (P with some or all occurrences of n replaced with m).

Technical detail: In System F, *n* and *m* below are not variables, they are uninterpreted constants ("names"). Names and variables should have different symbols.

$$
\begin{array}{|ll}
n = m & \\
\hline
\vdots & \\
P(n) & \\
\vdots & \\
P(m) & = \textit{Elim} \\
\vdots &
\end{array}
$$

Conjunction Elimination

This rule captures a simple idea: If we know $P \wedge Q$ to be true, we (obviously) know P to be true.

In Fitch:

$\wedge$ *Elimination*

$$
\begin{array}{r|l}
 & \vdots \\
 & P_1 \wedge \ldots \wedge P_i \wedge \ldots \wedge P_n \\
 & \vdots \\
\triangleright & P_i
\end{array}
$$

Conjunction Introduction

This rule is also obvious: If you have proved a bunch of things is true, their logical AND is true.

$\wedge$ *Introduction*

$$
\begin{array}{r|l}
 & \vdots \\
 & P_1 \\
 & \vdots \\
 & P_i \\
 & \vdots \\
 & P_n \\
 & \vdots \\
\triangleright & P_1 \wedge \ldots \wedge P_i \wedge \ldots \wedge P_n
\end{array}
$$

Disjunction Introduction

If a sentence is true, the OR of that sentence with anything else is true.

$\vee$ *Introduction*

$$
\begin{array}{r|l}
 & \vdots \\
 & P_i \\
 & \vdots \\
\triangleright & P_1 \vee \ldots \vee P_i \vee \ldots \vee P_n
\end{array}
$$

Negation Elimination

Another simple rule deals with double negation.

¬ Elimination

$$\begin{array}{l|l} & \vdots \\ & \neg\neg P \\ & \vdots \\ \triangleright & P \end{array}$$

⊥ Introduction

This rule allows you to mark a contradiction ($\bot$).

⊥ Introduction

$$\begin{array}{l|l} & \vdots \\ & P \\ & \vdots \\ & \neg P \\ & \vdots \\ \triangleright & \bot \end{array}$$

Conditional Elimination

Another name for implication is *the conditional*.

If P holds and $P \rightarrow Q$ holds, then Q holds.

→ Elimination

$$\begin{array}{l|l} & P \rightarrow Q \\ & \vdots \\ & P \\ & \vdots \\ \triangleright & Q \end{array}$$

This rule is known as *Modus Ponens*. It is one of the most basic rules of logic.

Biconditional Elimination

This rule is essentially the same as Conditional Elimination, but works in either direction.

↔ Elimination

$$\begin{array}{l|l} & P \leftrightarrow Q \ (\textit{or } Q \leftrightarrow P) \\ & \vdots \\ & P \\ & \vdots \\ \triangleright & Q \end{array}$$

Informal direct proof

The method of direct proof starts with premises and does a chain of *modus ponens* steps to arrive at the conclusion.

Theorem: If n is odd, then n^2 is odd.

Proof: If n is odd, then by definition, $n = 2k + 1$. Therefore, $n^2 = 4k^2 + 4k + 1$. But then this can be written $2(2k^2 + 2k) + 1$, so n^2 is odd. □

Written in pseudo-Fitch style (and fudging quantifiers):

→ *Elimination*

	odd(n)	
	odd(n) $\leftrightarrow$ n = 2k + 1	*def of* odd
	n = 2k + 1	*bicond elim*
	n = 2k + 1 $\rightarrow$ $n^2 = 4k^2 + 4k + 1$	*algebra*
	$n^2 = 4k^2 + 4k + 1$	→ *elim*
▷	odd(n^2)	*def of* odd