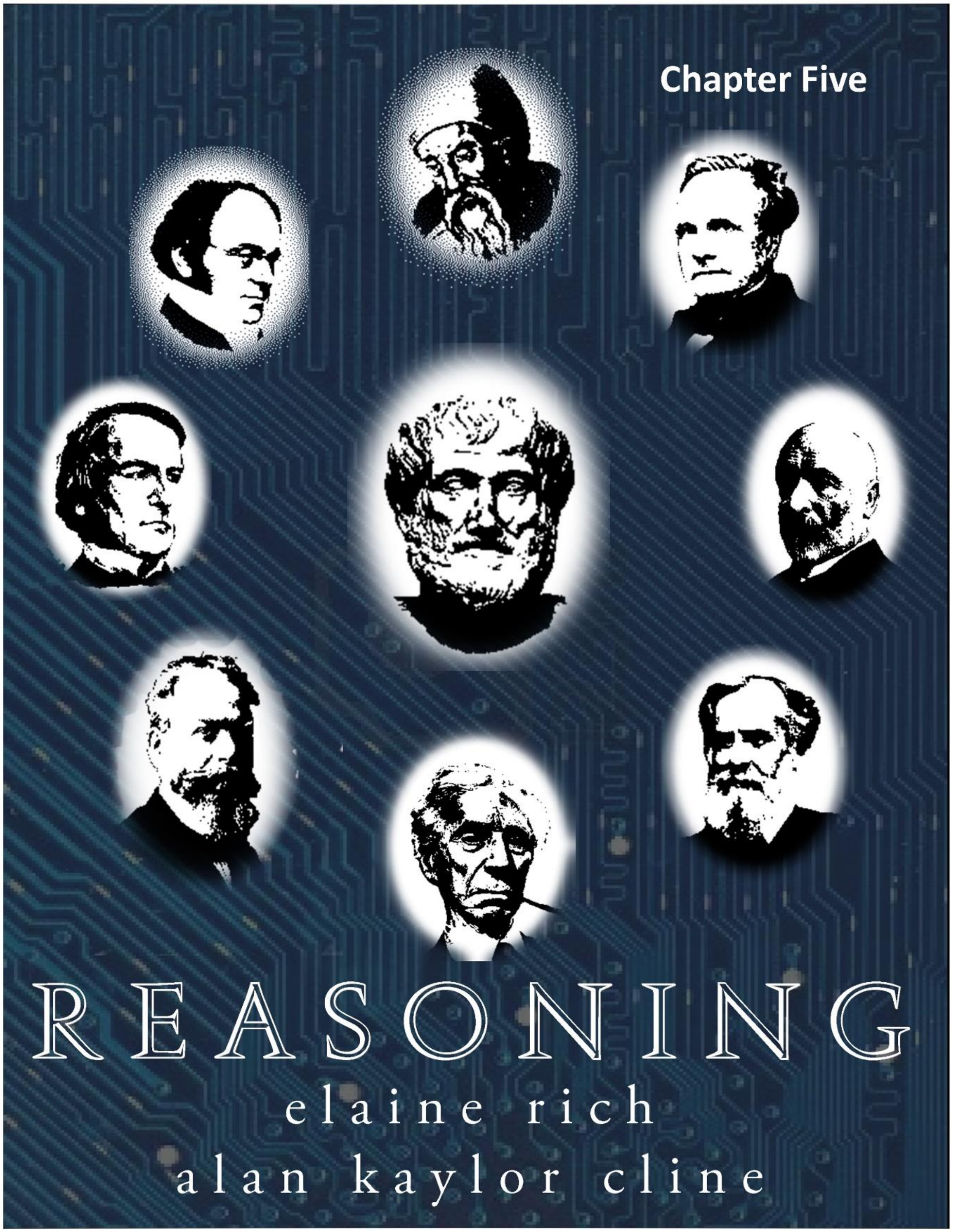

Chapter Five
REASONING
elaine rich
alan kaylor cline

Predicate Logic

The Logicians on our cover are:

Euclid (? - ?)

Augustus De Morgan (1806 – 1871)   Charles Babbage (1791 – 1871)

George Boole (1815 – 1864)   Aristotle (384 BCE – 322 BCE)   George Cantor (1845 – 1918)

Gottlob Frege (1848 – 1925)   John Venn (1834 – 1923)

Bertand Russell (1872 – 1970)

REASONING

AN INTRODUCTION TO LOGIC, SETS, AND FUNCTIONS

CHAPTER 5
PREDICATE LOGIC PROOFS

Elaine Rich
Alan Kaylor Cline

The University of Texas at Austin

Image credits:

Book on plane: *http://hardcoversandheroines.com/2013/06/*

Glass with straw on plane: *http://guestofaguest.com/tag/airplane-drinking/*

Fire-spewing dragon: *http://brightcat13527.deviantart.com/art/The-Fire-Dragon-16859273*

Alan Turing: *http://www.npg.org.uk/collections/search/portrait/mw165875*

http://www.cs.utexas.edu/learnlogic
Library of Congress Cataloging-in-Publication Data

Rich, Elaine, 1950 -
Reasoning—An Introduction to Logic Sets and Functions / Elaine Rich.— 1st ed. p. cm.
ISBN x-xxx-xxxxx-x 1

Table of Contents

Identities and Inference Rules for Predicate Logic I

Moving On From Representation to Proof

Recall that the point of our entire endeavor here is to get at *truth*.

What we've just spent a lot of effort on is simply *representation*. Our goal in doing that was to give ourselves a tool that lets us make claims that are *unambiguous*. We can't ask whether a claim is true if we're not sure what it means. No fighting about meaning allowed.

Approved Electronic Devices

Consider this sign. What does it mean?

Define:

$A(x)$: True if x has been approved.
$ED(x)$: True if x is an electronic device.
$UIF(x)$: True if x may be used in flight.

Here's one possible meaning for the sign:

If x is an electronic device that has been approved, it may be used in flight.

and

If it's not true that x is an electronic device that has been approved, it may not be used in flight.

We can write that as this logical statement:

[1] $\forall x\ (((ED(x) \wedge A(x)) \rightarrow UIF(x)) \quad \wedge \quad ((\neg(ED(x) \wedge A(x))) \rightarrow \neg UIF(x)))$

But is that really what the sign means? It says that anything that isn't an electronic device (whether approved or not) cannot be used in flight.

Surely the sign isn't intended to ban straws and good old fashioned books. So here's another possible meaning for the sign:

If x is an electronic device that has been approved, it may be used in flight.

and

If x is an electronic device that has *not* been approved, it may *not* be used in flight.

In other words, this is a sign *only* about electronic devices. It says nothing about books or straws (which, as it turns out, may be used). Nor does it say anything about guns or knives (which, as it turns out, may not be used).

We can write this second interpretation as this logical statement:

[2] $\forall x\ (((ED(x) \wedge A(x)) \rightarrow UIF(x)) \wedge ((ED(x) \wedge \neg A(x)) \rightarrow \neg UIF(x)))$

The point of our notation is that, while English signs are often ambiguous, logical statements are not.

Now it's time to move on and see how to reason with the logical sentences that we write. In other words, we need to learn how to write proofs.

The job of a proof is to:

- Assure us that some claim is true, and
- (Ideally) give us some insight into why it is true.

We should be on firm ground here – this is exactly what we demanded of our proofs in Boolean logic.

Review – Sound Arguments

Recall that, in our discussion of Boolean logic proofs, we said that an argument (proof) is:

- ***valid*** provided that every one of its steps can be justified by a sound inference rule.
- ***sound*** provided that it is valid *and* that its premises are true (in whatever world we are reasoning about).

We'll consider these same properties of predicate logic proofs.

Note as before the somewhat unfortunate (but conventional) use of the word "sound" to mean one thing when applied to inference rules (i.e., the property that truth is *preserved* by the reasoning process) and another thing when applied to entire arguments (i.e., that truth is both *introduced* by the premises and *preserved* by the argument).

Consider the following argument:

	Lucy is a cat.
	All cats are mammals.
Therefore:	Lucy is a mammal.

This argument is valid. (We'll soon describe the logical inference rules that will let us construct this proof.) It's also sound, since both of the premises are true.

Consider the following argument:

	Lucy is a cat.
	All cats live on Mars.
Therefore:	Lucy lives on Mars.

This argument is also valid. (We can prove it using the same inference rules we used above.) But it isn't sound, since it has a premise that isn't true.

Consider the following argument:

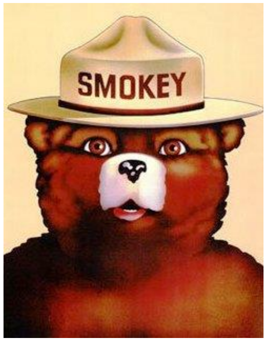
	Lucy lives on Mars.
	All cats live on Mars.
Therefore:	Lucy is a cat.

This argument isn't valid. Our second premise is a quantified version of CAT $\rightarrow$ MARS. We know MARS (in the case of Lucy). But, since Converse isn't a valid inference rule, we can't go from that to CAT (in the case of Lucy).

As in our discussion of Boolean logic, our focus here will be on the construction of valid proofs. Choosing premises (axioms) is another issue, best left to experts in whatever problem domain we want to consider.

Problems

1. Imagine a world that contains:



Bambi

(Part 1) Consider the following argument:

Smokey is a bear.
Smokey has a tail.
Therefore all bears have tails.

Which of the following is true of this argument:

a) It is sound.
b) It is valid but not sound.
c) Its premises are true but its reasoning is invalid.
d) It is total junk.

(Part 2) Consider the following argument:

Bambi is a bear.
All bears have tails.
Therefore Bambi has a tail.

Which of the following is true of this argument:

a) It is sound.
b) It is valid but not sound.
c) Its premises are true but its reasoning is invalid.
d) It is total junk.

(Part 3) Consider the following argument:

Smokey is a bear.
All bears have tails.
Therefore Smokey has a tail.

Which of the following is true of this argument:

a) It is sound.
b) It is valid but not sound.
c) Its premises are true but its reasoning is invalid.
d) It is total junk.

(Part 4) Consider the following argument:

Bambi is a bear.
All bears are brown.
Therefore Bambi isn't black.

Which of the following is true of this argument:

a) It is sound.
b) It is valid but not sound.
c) Its premises are true but its reasoning is invalid.
d) It is total junk.

Review – Natural Deduction Proofs

Boolean logic gives us two quite different-looking ways to build proofs:

- Truth tables
- Natural deduction

Of course, we proved (using truth tables) that our natural deduction rules are sound. So, at a logical level, these two techniques are equivalent. Sometimes one is more convenient than the other.

Unfortunately, we cannot, in general, use truth tables to construct proofs in predicate logic. Why not? Because a truth table is a finite list of possible interpretations (combinations of truth values of the primitive objects). But predicate logic lets us quantify over domains that may be genuinely infinite or ones whose members we don't know enough about to enumerate.

Here's an example where we have an infinite domain:

We can say that every positive integer has a successor:

$\forall x\ (\mathit{PositiveInteger}(x) \rightarrow \mathit{HasSucessor}(x))$

Here are a couple of examples where the domain is formally finite but we can't, as a practical matter, enumerate it:

We can say that every citizen must pay taxes:

$\forall x\ (\mathit{Citizen}(x) \rightarrow \mathit{MustPayTaxes}(x))$

Or we can say that all people have birthdays:

$\forall x\ (\mathit{Person}(x) \rightarrow \mathit{HasBirthday}(x))$

Clearly, if there are infinitely many things for which we would have to consider truth values, we can't enter them into a table of finite size. And, if there are finitely many things but we're unable to make a list of all them, we can't put them into a table either. So truth tables won't work.

But Boolean natural deduction does generalize to predicate logic. Proofs will have the same structure that they did in Boolean logic. Recall that a proof starts with a set of premises (or axioms). Then we apply rules of inference to create new statements. Any statement derived in this way is called a theorem.

To refresh our memory, here's a simple Boolean logic natural deduction proof:

[1]	p	Premise	
[2]	$p \rightarrow q$	Premise	
[3]	$(q \vee s) \rightarrow r$	Premise	
[4]	q	Modus Ponens	[1], [2]
[5]	$q \vee s$	Addition	[4]
[6]	r	Modus Ponens	[3], [5]

We've proven that *r* follows from our three premises.

To generalize natural deduction to predicate logic, we will need to add some new inference rules. As before, we must assure that all of our inference rules are sound (truth-preserving). In other words, if they're applied to a set of premises, they can derive only conclusions that are entailed by those premises. Said another way, they can derive only conclusions that must be true in every interpretation in which the premises are true.

We Inherit All the Rules From Boolean Logic

We've already developed a large collection of tools for working with Boolean logic sentences. Fortunately they generalize to predicate logic in two important ways:

- The identities can be used to transform wffs, *whether or not they are statements*.
- The inference rules can be applied to *fully quantified wffs*.

Let's look first at how we can use the Boolean identities. Recall that the job of the identities is to let us rewrite expressions into forms that we may find more useful.

We may want to do that to a wff that is inside the scope of one or more quantifiers. In that case, the wff itself is not a statement as it doesn't have a truth value. We are allowed to transform such a wff into another one as long as we preserve the property that, once it is fully quantified, it will have the same truth value as the original one did. All of our Boolean identities do that. This shouldn't come as a surprise. All of the operators that we use to build predicate logic wffs are defined in exactly the same way that they are in Boolean logic. So they have all the same properties.

For example, we can exploit the fact that *and* is commutative. Using that fact, we can rewrite the underlined wff on the left as the underlined one on the right:

$\forall x(\underline{P(x) \wedge Q(x)})$ can be rewritten as $\forall x(\underline{Q(x) \wedge P(x)})$.

Recall that one of the Boolean De Morgan's Laws tells us that, if we "push" *not* through an *and*, the *and* becomes an *or*:

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

So suppose we want to say that late sleepers and breakfast eaters don't overlap. We could say that everyone has the property that they don't both sleep late and eat breakfast. We write:

$\forall x\ (\underline{\neg(SleepsLate(x) \wedge EatsBreakfast(x))})$

Then, we could use De Morgan to rewrite the underlined wff above as the underlined wff below (which we can read as everyone either doesn't sleep late or doesn't eat breakfast, or possibly doesn't do either):

$\forall x\ (\underline{\neg SleepsLate(x) \vee \neg EatsBreakfast(x)})$

Now suppose that we are dealing with not just any wff but one that is in fact a statement. All variables in it are bound. It has a truth value. Now we can work with it in all the ways we could work with Boolean statements. We can, of course, apply the identities. And, now, we can also apply inference rules that allow us to combine statements to derive new statements that follow from the statements that we started with.

Let's do an example of the use of a Boolean logic inference rule. Suppose that we have a close but somewhat fractious family whose dynamics can be described with the following premises (assuming that the universe is the family members):

[1] If everyone comes to the party then there will be at least one dispute:

$\underline{(\forall x\ (HasComeToTheParty(x)))} \rightarrow \underline{(\exists y\ (Dispute(y)))}$

[2] Everyone has come to the party:

$\underline{(\forall x\ (HasComeToTheParty(x)))}$

We've underlined three wffs in [1] and [2], above. All three of them are statements. (One way to check this, in the case of [1], is to observe that → doesn't occur inside the scope of any quantifier. There is a fully quantified wff on its left, and another one on its right.)

Recall Modus Ponens: From premises $p \rightarrow q$ and p, infer q.

Observe that [1] has the form $p \rightarrow q$. And [2] is p. So we can conclude that there will be a dispute:

[1]	$(\forall x\ (HasComeToTheParty(x))) \rightarrow (\exists y\ (Dispute(y)))$	Premise	
[2]	$(\forall x\ (HasComeToTheParty(x)))$	Premise	
[3]	$(\exists y\ (Dispute(y)))$	Modus Ponens	[1], [2]

The one thing that we're still stuck on is the use of inference rules *within* the scope of quantifiers. We'll soon see how to do that.

Problems

1. Consider the following dialogue:

Brady:	What I think is that people who don't watch tv are clueless.
Drew:	Nah, the real truth is that clueless people don't watch tv.

Is it possible for both Brady and Drew to be right? (Hint, write both of them in predicate logic. You can use predicates like *WatchTV*(*x*) and *Clueless*(*x*).)

a) Yes
b) No

2. Consider the following dialogue:

Chris:	What I think is that dogs that won't play Frisbee are pretentious.
Jody:	Nah, the real truth is that unpretentious dogs think Frisbee is silly and refuse to play.

Is it possible for both Jody and Chris to be right?

a) Yes
b) No

3. Consider the following dialogue:

Bryn:	There aren't any unhappy smart people.
Kelly:	The way I see it, everyone is happy or not at all smart.

Is there any disagreement between Bryn and Kelly?

a) Yes
b) No

4. Consider the following sentence:

$\forall x\ (\exists y\ (\neg(P(x) \wedge Q(y))))$

We can use the Boolean identities to manipulate the wff that is inside the scope of the two quantifiers. Which of the following statements can be derived in that way? (Hint: Another way to ask this question is: Which of the following statements is/are equivalent to the one we started with?)

I. $\forall x\ (\neg\exists y\ (P(x) \wedge \neg Q(y)))$
II. $\forall x\ (\exists y\ (\neg P(x) \vee \neg Q(y)))$
III. $\forall x\ (\exists y\ (P(x) \rightarrow \neg Q(y)))$

a) Just I.
b) Just II.
c) Just III.
d) I and II.
e) II and III.

5. Consider the following sentence:

$\forall x\ ((P(x) \wedge \neg Q(x)) \rightarrow R(x))$

We can use the Boolean identities to manipulate the wff that is inside the scope of the quantifier. Which of the following statements can be derived in that way? (Hint: Another way to ask this question is: Which of the following statements is/are equivalent to the one we started with?)

I. $\forall x\ (\neg(P(x) \wedge \neg Q(x)) \vee R(x))$
II. $\forall x\ ((\neg P(x) \vee \neg\neg Q(x)) \vee R(x))$
III. $\forall x\ ((\neg P(x) \vee Q(x)) \vee R(x))$

a) Just I.
b) Just II.
c) Just III.
d) Two of the three.
e) All three.

6. ***Contagious Disgruntledness***

Suppose that we have a group of people among whom grumpiness is highly contagious. If even one person gets disgruntled, the bad vibes will quickly spread to the whole group. So assume that the universe is our group of people. Then we might write:

[1] $(\exists x\ (Disgruntled(x))) \rightarrow (\forall x\ (Disgruntled(x)))$

(Part 1) Let's first use everyday reasoning. (In other words, we're not limited to the formal inference rules that we've so far described.) Which of the following additional premises would be sufficient, when combined with [1], to allow us to conclude that everyone is disgruntled:

I. *Disgruntled*(*Grouchy*)
II. $\neg$*Disgruntled*(*Sunshine*)
III. $(\exists x\ (Disgruntled(x)))$

a) Just I.
b) Just II.
c) Just III.
d) I or II.
e) I or III.

(Part 2) Now let's use just the formal inference rules that we've described. Which of the following additional premises would be sufficient, when combined with [1], to allow us to conclude that everyone is disgruntled:

I. *Disgruntled*(*Grouchy*)
II. $\neg$*Disgruntled*(*Sunshine*)
III. $(\exists x\ (Disgruntled(x)))$

a) Just I.
b) Just II.
c) Just III.
d) I or II.
e) I or III.

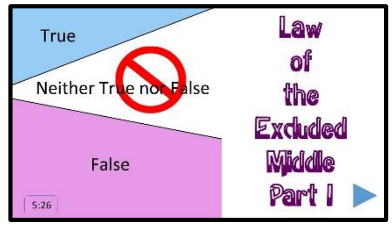
Law of the Excluded Middle

We've just said that we will import all of the identities of Boolean logic into our system of predicate logic. There is one special case that we should say a bit more about.

The ***Law of the Excluded Middle***:

For any statement P: $P \lor \neg P$

The Law of the Excluded Middle (LEM) is a useful theorem proving tool.



https://www.youtube.com/watch?v=oaSLa1Ya5-M

> Watch the video for a good example. In it, we prove that there exist irrational numbers x and y such that:
>
> x^y is rational.

Classical logic, following Aristotle, assumes both LEM and the ***Principle of Noncontradiction***:

For any statement P: $\neg(P \land \neg P)$

In Boolean logic, both of these rules are tautologies:

		LEM		Noncontradiction
P	$\neg P$	$P \lor \neg P$	$P \land \neg P$	$\neg(P \land \neg P)$
T	F	T	F	T
F	T	T	F	T

In predicate logic, they are not. They (or some variant) must be asserted as premises. The Principle of Noncontradiction is not controversial, since it's not possible to build a useful logical system without it.

LEM, however, is slightly controversial. There are philosophers and mathematicians (such as constructivists) who don't assume it.

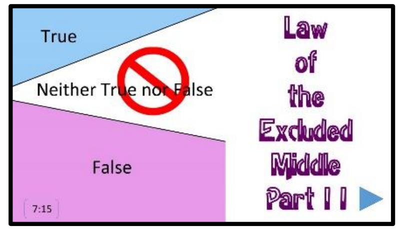
But consider the following derivation:

[1]	$\neg(P \land \neg P)$	Principle of Noncontradiction
[2]	$\neg P \lor \neg\neg P$	De Morgan
[3]	$\neg P \lor P$	Double Negation
[4]	$P \lor \neg P$	Commutativity

Line [4] is exactly LEM. We have derived it from the (noncontroversial) Principle of Noncontradiction. Why then, must LEM be assumed as a separate premise if we want to use it? The answer is that, in predicate logic, Double Negation is not a tautology. It too is rejected by the constructivists.

Arguments for rejecting LEM (and Double Negation) fall into at least two categories:

- Philosophical and mathematical ones. For example, mathematicians from the constructivist school do not want to accept, as a proof of the existence of some object *x* with property *P*, anything short of the exhibition of a specific such *x* or a concrete procedure for finding it.



- Essentially linguistic ones. There are English sentences that, on first analysis, seem to contradict LEM in the sense that they do not appear to be either true or false. Further analysis often shows that the issue is the mapping from English into logic, rather than a logical problem with the LEM.

https://www.youtube.com/watch?v=r_KG3EZuJmw

We will follow Aristotle and assume the Law of the Excluded Middle. Then we can prove Double Negation by running the derivation we just did, but backwards. (Alternatively, we could assume Double Negation and then prove LEM.)

Problems

1. Assume standard definitions of the words used here. Mark each argument as sound (correct) or not.

(Part I)

[1]	*x* is an integer	Premise	
[2]	*x* is not negative	Premise	
[3]	*x* is a positive integer	LEM	[1], [2]

(Part II)

[1]	*x* is an integer	Premise	
[2]	*x* is not even	Premise	
[3]	*x* is an odd integer	LEM	[1], [2]

Quantifier Exchange

We've already seen that we can get a lot of mileage out of our Boolean reasoning techniques. But not quite enough. To reason with quantified statements, we need one new pair of identities and four new inference rules.

First, we'll introduce the two identities. They are variously called:

- Quantifier exchange, or
- Pushing *not*s through quantifiers.

Here are they are:

[Quantifier Exchange A] $\neg(\forall x\ (P(x))) \equiv \exists x\ (\neg P(x))$

[Quantifier Exchange B] $\neg(\exists x\ (P(x))) \equiv \forall x\ (\neg P(x))$

Let's read these to see what they say:

[A] Suppose that we have that it's not the case that $P(x)$ is true for all values of x. An equivalent way of saying the same thing is that there must be at least one value of x for which it's false.

[B] Suppose we have that there is no value of x for which P is true. An equivalent way of saying the same thing is that $P(x)$ is false for all values of x.

As we work with logical expressions, we often want to "reduce the scope of *not*s." By this we mean that we want *not*s to apply to simple subexpressions rather than complex ones. It's often easier to work with the overall expression once it's in that form. So one way that describes how these rules are often used is:

You can push a *not* rightward across a quantifier by doing two things:

1. Flip the quantifier. In other words, ∀ becomes ∃ and ∃ becomes ∀. Then:
2. Apply the *not* to the entire scope of the original quantifier.

For example, we might start with the claim that it's not true that everyone is the mother of someone. Then we can reason as follows:

[1]	$\neg\forall x\ (\exists y\ (MotherOf(x, y)))$	Premise	
[2]	$\exists x\ (\neg\exists y\ (MotherOf(x, y)))$	Quantifier Exchange [A]	[1]
[3]	$\exists x\ (\forall y\ (\neg MotherOf(x, y)))$	Quantifier Exchange [B]	[2]

To get [2], we pushed the *not* through ∀, which changed it to ∃ and landed the ¬ just inside the outermost parentheses. To get [3], we did a second quantifier exchange. This time we pushed the *not* through ∃, which changed it to ∀ and landed the ¬ just inside one more level of parenthesization.

Now we have the equivalent claim that there exists someone who fails to be the mother of every single person (assuming people is our domain).

Recall that we've already observed that:

- The quantifier ∀ is a shorthand for a large conjunction. When we say $\forall x\ (P(x))$, what we're really saying is:

  [1] $P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge \ldots \wedge P(x_n)$, where n is the number of objects in our domain.

- The quantifier ∃ is a shorthand for a large disjunction. When we say $\exists x\ (P(x))$, what we're really saying is:

  [2] $P(x_1) \vee P(x_2) \vee P(x_3) \vee \ldots \vee P(x_n)$, where n is the number of objects in our domain.

When we look at the quantifiers in this way, what we see is that our quantifier exchange rules are simply giant versions of the Boolean De Morgan's laws. Recall that those laws are:

[De Morgan 1]	$(\neg(p \wedge q)) \equiv (\neg p \vee \neg q)$	Push *not* through *and*.
[De Morgan 2]	$(\neg(p \vee q)) \equiv (\neg p \wedge \neg q)$	Push *not* through *or*.

As given here, each of De Morgan's laws applies in the case of a conjunction or disjunction of exactly two terms. Suppose that the generalization to n (for any value of n) terms were true:

[Generalized De Morgan 1] $(\neg(p_1 \wedge p_2 \wedge p_3 \wedge \ldots \wedge p_n)) \equiv (\neg p_1 \vee \neg p_2 \vee \neg p_3 \vee \ldots \wedge \neg p_n)$

[Generalized De Morgan 2] $(\neg(p_1 \vee p_2 \vee p_3 \vee \ldots \vee p_n)) \equiv (\neg p_1 \wedge \neg p_2 \wedge \neg p_3 \wedge \ldots \wedge \neg p_n)$

(Note that we've left out a lot of parentheses here. They're not necessary since both *and* and *or* are associative. So we've used this more readable form rather than our usual, fully parenthesized one.) It turns out that both of these generalized claims are true. This can be proved straightforwardly using a proof technique called induction that we'll consider later in this course. For now, assume that they're true.

Now let's rewrite our new Generalized De Morgan's laws using quantifiers. Assume that our domain contains n objects and that $P(x)$ is equivalent to the Boolean expression p_x. (For example, $P(1)$ and p_1 are just two ways of saying the same thing.)

Look at the left hand side of Generalized De Morgan 1. It's the *not* of a large conjunction. Since the universal quantifier $\forall$ is a shorthand for a large conjunction, we can use it here. Our left hand side becomes:

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

And the right hand side is simply a large disjunction (each term of which happens to be negated). Our shorthand for that is the existential quantifier $\exists$. So our right hand side becomes:

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

Putting the two sides together and then doing a similar thing for Generalized De Morgan 2, we get:

[Quantified Generalized De Morgan 1] $\neg\forall x\ (P(x)) \equiv \exists x\ (\neg P(x))$

[Quantified Generalized De Morgan 2] $\neg\exists x\ (P(x)) \equiv \forall x\ (\neg P(x))$

Compare these two equivalences to the two Quantifier Exchange rules at the top of this slide. They look the same. There actually is one difference: we derived our Generalized De Morgan rules assuming some finite sized set (we called the size n). The Quantifier Exchange rules at the top of this slide, however, don't make that assumption. They work for any size set, finite or infinite. That's why they're *new* rules for reasoning in predicate logic. They're not just another way of writing something that we could have done in our Boolean framework.

So what we see now is that the Quantifier Exchange rules are generalizations to any size set, including infinite ones, of De Morgan's laws.

Since Quantifier Exchange and De Morgan's laws do the same thing, we'll often see that it's useful to combine them. In particular, if we apply Quantifier Exchange and are able to produce a wff that contains no more quantifiers, we may continue to simplify by treating the wff as a Boolean expression and applying De Morgan's laws.

For example, suppose that we start with the (obviously true) statement that there's no one who both hates chocolate and is trustworthy. Then we can reason as follows:

[1]	$\neg\exists x\ (Hates(x, Chocolate) \wedge Trustworthy(x))$	Premise	
[2]	$\forall x\ (\neg(Hates(x, Chocolate) \wedge Trustworthy(x)))$	Quantifier Exchange B	[1]
[3]	$\forall x\ (\neg Hates(x, Chocolate) \vee \neg Trustworthy(x))$	(Boolean) De Morgan	[2]

Now we've reduced the scope of *not* to individual predicates. We have derived a new universal rule: One doesn't hate chocolate or one isn't trustworthy.

From now on, we'll lump Quantifier Exchange A and Quantifier Exchange B together as a single identity called simply Quantifier Exchange.

Problems

1. Consider the plight of a poor talent scout for a musical. She moaned, "It's hopeless. Everyone who tried out sounds like a cat in heat or has two left feet." We'll simplify a bit. Define:

$S(x)$:	True if x can sing.	Alternatively: True if x does not sound like a cat in heat.
$D(x)$:	True if x can dance.	Alternatively: True if x does not have two left feet.

Assume that x ranges over the set of people who tried out. Which one or more of the following expressions correspond(s) to the scout's lament? (Hint: Write out one expression that captures the meaning. Then use the Quantifier Exchange rules to see what other expression(s) are equivalent to yours.)

I. $\forall x\ (\neg S(x) \vee \neg D(x))$
II. $\forall x\ (\neg(S(x) \vee D(x)))$
III. $\neg\exists x\ (S(x) \wedge D(x))$

2. Consider this sign (often seen on the doors of greedy movie theaters). Exactly what does it mean? First, let's assume that the adjective "outside" applies to both food and drink.

NO OUTSIDE FOOD OR DRINK

English Aside

Yet another way in which English suffers from a lack of explicit parentheses is that it's ambiguous whether modifiers attach just to the thing they're closest too or whether they have wider scope. In the case of our sign, we're assuming (since this is consistent with maximum greediness) that the intent is:

- No outside (food or drink).

But suppose that we'd said, "No smelly food or drink." Now (given the lack of smell of most drinks) the most likely interpretation is:

- No (smelly food) or drink.

Define:

Outside(*x*): True if *x* is from outside.
Food(*x*): True if *x* is food.
Drink(*x*): True if *x* is a drink.

Assume that the job of the sign is to make a claim that must be true of objects that are brought into the theater.

Which (one or more) of the following expressions correspond(s) to the intended meaning of this sign? (Hint: Write out one expression that captures the meaning. Then use the Quantifier Exchange rules to see what other expression(s) are equivalent to yours.)

I. $\forall x\, (\neg(Outside(x) \wedge (Food(x) \vee Drink(x)))$
II. $\forall x\, (\neg Outside(x) \vee \neg(Food(x) \vee Drink(x)))$
III. $\neg\exists x\, (Outside(x) \wedge (Food(x) \vee Drink(x)))$
IV. $\forall x\, (\neg Outside(x) \vee (\neg Food(x) \wedge \neg Drink(x)))$

3. Let the universe of discourse be the natural numbers. Let $P(x, y)$ correspond to the claim that $x > y$. What are the truth values of each of the following expressions? Justify your answer.

a) $\forall x\, (\forall y\, (P(x, y)))$
b) $\forall x\, (\exists y\, (P(x, y)))$
c) $\exists x,y\, (P(x, y))$
d) $\neg\exists x\, (\forall y\, (P(x, y)))$

4. Consider the following expression:

$$\neg\exists x\ (\forall y\ (\neg(P(x) \vee Q(x)) \wedge \neg R(x, y)))$$

Using the Quantifier Exchange rules and the rules of Boolean logic, we can construct an equivalent expression that contains no instances of $\neg$.

Which of the following is such an expression:

a) $\forall x\ (\exists y\ ((P(x) \wedge Q(x)) \vee R(x, y)))$
b) $\forall x\ (\exists y\ ((P(x) \vee Q(x)) \vee R(x, y)))$
c) $\exists x\ (\forall y\ ((P(x) \wedge Q(x)) \vee R(x, y)))$
d) $\exists x\ (\forall y\ ((P(x) \vee Q(x)) \vee R(x, y)))$
e) $\forall x\ (\exists y\ (P(x) \wedge (Q(x) \vee R(x, y))))$

5. Prove that these two expressions are equivalent by using Quantifier Exchange, plus Boolean identities, to transform the first into the second:

[1] $\neg\exists x\ (\forall y\ (\exists z\ (P(x, y) \wedge P(y, z))))$

[2] $\forall x\ (\exists y\ (\forall z\ (P(x, y) \rightarrow \neg P(y, z))))$

Identities and Inference Rules for Predicate Logic I

New Rules for Instantiating and Generalizing Quantifiers

So now our proof toolkit includes all of the Boolean identities and inference rules. And we have just added two new identities, the two Quantifier Exchange rules.

The last thing we need is some new inference rules that will allow us to reason with quantified expressions. To see why, consider the following argument that we should surely be allowed to make:

Breathes

[1]	$\forall x\ (Student(x) \rightarrow Person(x))$	All students are people.
[2]	$\forall x\ (Person(x) \rightarrow Breathes(x))$	All people breathe.

Therefore:

[3]	$\forall x\ (Student(x) \rightarrow Breathes(x))$	All students breathe.

An argument with this structure is called a ***syllogism***. Syllogisms have formed the core of classical logic since the days of Aristotle. We have no problem writing syllogisms in Boolean logic. In fact, the inference rule that lets us write them is called Hypothetical Syllogism:

From $p \rightarrow q$ and $q \rightarrow r$, infer $p \rightarrow r$.

We've already seen that we can treat a predicate logic statement (i.e., an entire expression that has a truth value) as a single proposition and then reason with it just as we would have done in Boolean logic.

But this won't solve the problem of writing syllogisms like the ***Breathes*** one. To do that, we need to reach inside quantified expressions. We want to match the (nonstatement) wff $Person(x)$ on the right hand side of the implication in [1] with the same (nonstatement) wff on the left hand side of the implication in [2] so that we can chain the two statements together. We don't yet have a way to do that. We'll present one soon.

But, first, let's see why we can't just wing it.

NONMYTHOLOGICAL STUDENTS
Assume the following premises:

[1] All students are creatures.
[2] All zebras are creatures.
[3] All creatures are mortal unless they are mythological.
[4] All mythological creatures have stripes.
[5] There are no striped students.
[6] Not all zebras have stripes.

Can we conclude:

[7] All students are mortal.
[8] All zebras are mortal.

The answer is yes for [7] and no for [8]. And this time, it's much less obvious how we can reason correctly and not make mistakes.

To guarantee that we continue to use only inference rules that we know are sound, we need to formalize the ways that we can work with quantified statements.

Problems

1. For each of the following arguments, indicate whether or not it is a syllogism:

(a) All great desserts are chocolate.
Brownies are chocolate.
∴ Brownies are great desserts.

(b) All unicorns have stripes.
All striped things are happy.
∴ All unicorns are happy.

(c) All wooly things are warm.
All sheep are wooly.
∴ All sheep are warm.

Our Approach – Back and Forth to Boolean Logic

So we need a way to reach inside quantified expressions to reason with them.

To solve this problem, we'll propose a bigger picture plan:

- We observe that we have very flexible inference rules for Boolean logic.

- So, to work in predicate logic, we will define a set of rules that allow us to:

  1. Transform predicate logic expressions into Boolean ones,
  2. Reason in the Boolean world, and finally
  3. Transform the Boolean expressions back into predicate logic.

Of course, we must do that in a way that is sound.

Here's the key idea we'll use:

- In Boolean logic we reason about individuals one at a time.

- But quantified statements, like $\forall x\ (P(x))$ or $\exists x\ (Q(x))$, let us reason about entire groups of individuals all at once.

- So what we need is a way to transform quantified statements into statements about particular individuals. Then we can reason with them.

- And finally we need a way to transform back into generalized (quantified) statements.

With this idea in mind, let's return to the *Nonmythological Students* example:

[1] All students are creatures.
[2] All creatures are mortal unless they are mythological.
[3] All mythological creatures have stripes.
[4] There are no striped students.

We'll reason as follows. Consider any student. (Now we're not talking about all students. We're talking about just one student but (s)he's an arbitrary one. We don't know anything special about him/her.) We do know (s)he must be a creature (from [1]) and that (s)he is not striped (from [4]). Thus (s)he cannot be mythological (from [3]). So (s)he must be mortal (from [2]). And so we can conclude, that if an arbitrary student must be mortal, all students are mortal. We've thus proved:

[5] All students are mortal.

The set of steps that we just did is what we now must formalize.

Working with Universal Quantifiers: Arbitrary Elements

To make this work in the case of *universally* quantified expressions, we'll introduce the idea of an ***arbitrary element***. What we mean by "arbitrary" is that the element has no additional characteristics other than being an element of the universe.

To see how this helps, let's return to the ***Breathes*** syllogism problem. Recall that we have:

[1]	$\forall x\ (Student(x) \rightarrow Person(x))$	All students are people.
[2]	$\forall x\ (Person(x) \rightarrow Breathes(x))$	All people breathe.

And we want to prove:

[3]	$\forall x\ (Student(x) \rightarrow Breathes(x))$	All students breathe.

We'll assume a universe of living things. Let's let *c* be a name for some "arbitrary living thing". We could call our arbitrary living thing anything we want. We could call him/her/it *supercalifragilisticone*. The only thing that matters is that we don't pick a name that we're using anywhere else in our system. We don't want to be able to pick up any extra information about our "arbitrary living thing" that we wouldn't know of absolutely every living thing.

We know, from [1] above, that *anyone* who is a student is a person.

There ought to be an inference rule (and soon we'll define one) that lets us apply this claim to the particular case of our arbitrary living thing *c*. That would give us:

[1a] $Student(c) \rightarrow Person(c)$

Similarly, from [2] above, we have that *anyone* who is a person breathes.

The same rule that let us go from [1] to [1a] should let us go from [2] to [2a], which makes a particular claim about our arbitrary living thing *c*:

[2a] $Person(c) \rightarrow Breathes(c)$

Now we've got two quantifier-free expressions. The Boolean hypothetical syllogism rule can be used to chain [1a] and [2a] together to produce another claim about our particular living thing *c*:

[3a] $Student(c) \rightarrow Breathes(c)$

Now comes the biggie: Since *c* was an arbitrary living thing about whom we knew nothing except what we could derive from general statements about all living things, anything we know about *c* must generalize to the entire domain of living things. So we need a second new inference rule that will allow us to conclude that "*any* student breathes":

[3] $\forall x\ (Student(x) \rightarrow Breathes(x))$

The details will come soon, but let's review the big picture of what we just did because this is the key:

- We had a universal statement (actually two of them).
- Using new-rule-to-come-1, we gave a name to an arbitrary element.
- We argued (using Boolean logic) about the arbitrary element and came to some conclusion about it.
- And then, since it was arbitrary, we expressed the conclusion, using new-rule-to-come-2, as a new universal statement.

Problems

1. Assume the following premises:

[1]	$\forall x\ (Phlobber(x) \rightarrow Crazy(x))$	All phlobbers are crazy.
[2]	$\forall x\ ((Green(x) \wedge Zamzow(x)) \rightarrow Crazy(x))$	All green zamzows are crazy.
[3]	$\forall x\ (Crazy(x) \rightarrow Funny(x))$	All crazy things are funny.

Let's use the reasoning process that we just described. Let *c* be an arbitrary element of the universe (as we did above). Consider the following statements that we might like to derive:

[I]	$Phlobber(c) \rightarrow Crazy(c)$	If *c* is a pholbber, *c* is crazy.
[II]	$Zamzow(c) \rightarrow Crazy(c)$	If *c* is a zamzow, *c* is crazy.
[III]	$Crazy(c) \rightarrow Funny(c)$	If *c* is crazy, *c* is funny.

Which of the following statements is true:

a) Exactly one of these can be derived using the idea of an arbitrary element.
b) Just I and II can be derived using the idea of an arbitrary element.
c) Just I and III can be derived using the idea of an arbitrary element.
d) Just II and III can be derived using the idea of an arbitrary element.
e) All of them can be derived using the idea of an arbitrary element.

Working with Existentially Quantified Statements: "The One"

Now we need a similar process for working with *existentially* quantified statements. But there is a crucial difference. An existential quantifier guarantees the existence of (possibly only) one element that satisfies its predicate. So we can't pick an arbitrary element and assume that it's "the one". What we can do is to give a name to "the one". We'll soon describe exactly how to do this with new-rule-to-come-3.

As before, the key is that we must pick a name that doesn't match any other elements that we're reasoning about. We know nothing special about "the one" except that it exists and satisfies one particular predicate. So, in particular, if we are working with multiple existentially quantified expressions, we must not assume that "the one" for one of them is necessarily the same as "the one" for any others.

> Suppose that we have:
>
> [1] ∃x (*Likes*(x, *Rootbeer*))
> [2] ∃x (*Likes*(x, *Beets*))
>
> Assuming that we have new-rule-to-come-3, which will let us name "the one", we could write:
>
> [1a] *Likes*(*d**, *Rootbeer*)
> [2a] *Likes*(*e**, *Beets*)
>
> What's critical is that we not use the same symbol in both [1a] and [2a]. It's possible that *d** = *e** (in other words that it's the same person who likes both rootbeer and beets), but we have no basis for concluding that.

And now for the key difference between the arbitrary elements that arise when we're working with universally quantified expressions and the existential elements ("the ones") that arise from existentially quantified expressions:

- Dealing with arbitrary elements: We saw that if we have proved a claim about an *arbitrary* element, then we can (using new-rule-to-come-2) generalize that claim so that we create a new *universally* quantified statement.

- Dealing with existential elements: After proving a claim about "the one", all we know is that our claim is true of that one and, thus, of at least one element. So we can generalize our claim but only to an *existentially* quantified statement.

So the final rule that we're about to describe (for now we'll call it new-rule-to-come-4) will let us assert that existentially quantified statement.

As a bookkeeping matter: When we're writing a proof, we must clearly distinguish between arbitrary elements (about which we've made no assumptions) and "the ones" (about which we've made the specific assumption that they are "the ones"). To make it easy to keep these two kinds of objects distinct, every time we introduce a name for "the one", we'll end the name with an asterisk (as we've just done).

While it's critical that we not confuse arbitrary elements and "the ones", it's worth pointing out that people do sometimes fail to make that distinction. When that happens, logical nonsense results.

Suppose I say, "Some kittens are fluffy":

[1] $\exists x\ (Kitten(x) \wedge Fluffy(x))$

Since this is an existentially quantified statement, we are allowed to write:

[1a] $Kitten(k^*) \wedge Fluffy(k^*)$

In this case, *k** is "the one", about whom we know absolutely nothing except that it's a fluffy kitten.

Now suppose that someone else says, "My kitten Lucy isn't fluffy" (as if that were a counterexample to my claim). We'd write that specific statement as:

[2] $Kitten(Lucy) \wedge \neg Fluffy(Lucy)$

Now compare [1a] and [2]. It is possible that Lucy and *k** are aliases for the same individual. But nothing requires that they be. So we can't conclude any contradiction between them. Our guy *k** may be a big ball of fluff while his third cousin Lucy could be as sleek as can be.

So I should respond, "I did not make a universal statement about all kittens. I said there is at least one fluffy kitten. Lucy might not be that one fluffy kitten".

Problems

1. Assume:

[1] $\exists x\ (Mathematician(x) \wedge Funny(x))$
[2] $Philosopher(Frege)$
[2] $Philosopher(Russell)$

Which of these can be proved using some combination of the new rule that we just described, plus the identities and inference rules of Boolean logic?

I. $Mathematician(Frege) \wedge Funny(Frege)$
II. $Mathematician(e^*) \wedge Funny(e^*)$
III. $Mathematician(Russell)$
IV. $Mathematician(c^*)$

Substituting One Variable for Another

Notice that, both when we introduce arbitrary elements and when we introduce names for "the one", we end up rewriting logical expressions and substituting one name for another.

For example, suppose we start with:

[1] $\exists x\ (Boy(x) \wedge Stupid(x))$

Then we can name "the one" b^* and write:

[1a] $Boy(b^*) \wedge Stupid(b^*)$

What we've done is to substitute b^* for x. Notice that, when we do this, we must do it consistently within the entire scope of the quantifier that binds x. Within that scope, every instance of x (and nothing else) must be replaced by b^*.

We need a concise notation for describing such substitutions. Let P be any wff and let x and y be any variables. Then we'll write:

$P(x/y)$

Read this as, "P, with x (whatever it is) substituted in every place that a free (unbound) instance of y (whatever it is) appeared.

So, continuing with our example, let P be the wff:

[3] $Boy(x) \wedge Stupid(x)$

Then $P(b^*/x)$ yields:

[1a] $Boy(b^*) \wedge Stupid(b^*)$

You may now be saying, "Wait. I thought that you said that substitution happens to free (unbound) variables. Yet, in [1], the instances of x are bound by the existential quantifier $\exists$." True. But we're letting P be [3], the (nonstatement) wff that is *inside the scope* of $\exists$. So the instances of x are free and we'll substitute for them. This will be important since, as in [1a], we want to derive statements that are quantifier-free.

We are now ready to see the new rules. Two deal with $\forall$ and two deal with $\exists$. Two (one for $\forall$ and one for $\exists$) remove the quantifiers (in favor of arbitrary or existential elements). The process of introducing the arbitrary or existential element is called *instantiation* (i.e., defining an instance). Two (again one each for $\forall$ and $\exists$) reapply the quantifiers. This process is called *generalization* (i.e., making a general statement from an instance).

Problems

1. Let P be the wff: $Hungry(x) \wedge Thirsty(x)$.

What is $P(Gerry/x)$?

a) $Hungry(x, Gerry) \wedge Thirsty(x, Gerry)$
b) $Hungry(Gerry) \wedge Thirsty(Gerry)$
c) $Hungry(Gerry) \wedge Thirsty(x)$

Explanation: We substitute *Gerry* for every occurrence of *x*.

2. Let P be the wff: $Friendly(x) \wedge Likes(x, y)$.

What is $P(Sunny/x)$?

a) $Friendly(Sunny) \wedge Likes(y, Sunny)$
b) $Friendly(Sunny) \wedge Likes(x, y)$
c) $Friendly(Sunny) \wedge Likes(Sunny, y)$
d) $Friendly(Sunny) \wedge Likes(Sunny, Sunny)$

Universal Instantiation

Universal Instantiation:

$$\frac{\forall x\,(P(x))}{\therefore\ P(c/x)}$$

Note: The substitution of *c* for *x* must be for all free occurrences of *x* and for no other variables.

Thus, from the fact that *P* is true for all values of *x*, conclude that it must be true for any particular object *c*.

This is what we have been calling new-rule-to-come-1.

Uses of Universal Instantiation usually occur early in proofs for which the premises have universally quantified statements.

There are two main ways in which we use this rule. The first is the simplest: we want to prove a claim about some particular individual. We reason that, if *P* must be true of everyone, it must, in particular be true of Smokey or Fred or 275 or whatever.

For example, we can use Universal Instantiation to make the following argument:

No cats are birds. Lucy is a cat. Therefore, Lucy is not a bird.

Define:

Bird(x): True if x is a bird.
Cat(x): True if x is a cat.

Proof:

[1]	∀x (*Cat*(x) → ¬*Bird*(x))	Premise	
[2]	*Cat*(*Lucy*)	Premise	
[3]	*Cat*(*Lucy*) → ¬*Bird*(*Lucy*)	Universal Instantiation	[1]
[4]	¬*Bird*(*Lucy*)	Modus Ponens	[2] , [3]

And now for the second use of Universal Instantiation: We can use it to make claims about *arbitrary elements* (about which we know nothing else). To do this, we must use names about which we have no other premises (unlike what we did in the Lucy case above). If we introduce arbitrary elements, then we can prove new general claims by doing what we've already done in our proof of the <ex Breathes> syllogism. We introduce arbitrary elements with Universal Instantiation, reason about them, and then conclude by using what we've been calling new-rule-to-come-2 (next) to derive a new universally quantified claim. We'll hold off giving an example of this until we have new-rule-to-come-2 to work with.

Problems

1. Assume that we are willing to use Universal Instantiation, along with some number of Boolean identities and inference rules. Consider the following premises. There's a way to exploit exactly four of them (together) to prove that *ActsStrange*(*Kelly*). Which one isn't necessary?

a) $\forall x\ ((SleepDeprived(x) \vee SimplyCrazy(x)) \rightarrow ActsStrange\ (x))$
b) $\forall x\ ((AttendsUT(x) \wedge SimplyCrazy(x)) \rightarrow ActsStrange(x))$
c) $\forall x\ (Student\ (x) \rightarrow SleepDeprived(x))$
d) $\forall x\ (AttendsUT\ (x) \rightarrow Student\ (x))$
e) $AttendsUT\ (Kelly)$

2. Suppose that we start with:

[1] $\neg\forall x\ (Happy(x) \vee Sad(x))$

Which of these claims is true:

a) We can apply Universal Instantiation to [1] and derive $\neg(Happy(Pogo) \vee Sad(Pogo))$.
b) We can apply Universal Instantiation to [1] and derive $(\neg Happy(Pogo) \vee \neg Sad(Pogo))$.
c) We cannot apply Universal Instantiation to [1].

Universal Generalization

Universal Generalization: $P(c/x)$ — P is true for the specific value c (which appears everywhere a more general value x might appear).

$\therefore \ \forall x\ (P(x))$ — P is true for any value x.

Restrictions:

- c must not appear as a free variable in $P(x)$.
- c is not mentioned in any hypothesis or undischarged assumption. (Recall that, to use the Conditionalization inference rule, we introduce one or more assumptions, reason with them, and then *discharge* them when we capture their effect by writing that they imply the conclusion. What we're saying here is that we cannot generalize any variable that was arbitrarily introduced for that purpose but that has not yet been discharged.) In other words, c is an *arbitrary* element of the universe.

So, if $P(c)$ holds for some arbitrary element c of the universe, we can conclude that it must in fact hold for any element. Thus we can conclude, $\forall x\ (P(x))$.

But we must be careful. If c represents some specific element of the universe that may have properties that other elements don't have, then the generalization is not valid.

For example, recall our proof that Lucy is not a bird:

[1]	$\forall x\ (Cat(x) \rightarrow \neg Bird(x))$	Premise	
[2]	$Cat(Lucy)$	Premise	
[3]	$Cat(Lucy) \rightarrow \neg Bird(Lucy)$	Universal Instantiation	[1]
[4]	$\neg Bird(Lucy)$	Modus Ponens	[2] , [3]

Notice that, on line [2], we had *Cat*(*Lucy*). But *Lucy* wasn't an arbitrary element. It was a specific element about which we had a premise. So we cannot generalize [2] to conclude that everything in the universe is a cat.

This is what we have been calling new-rule-to-come-2.

Use of Universal Generalization usually occurs at the end of proofs for which the conclusion has a universally quantified statement. Before we can apply it, we must go back through our proof to make sure that the value that we are generalizing is in fact an arbitrarily chosen one.

We can now write out a complete proof, in our standard notation, of the ***Breathes*** syllogism. Note that the Hypothetical Syllogism rule that we use here is the same one we've been using since we started writing Boolean proofs.

[1]	$\forall x\ (Student(x) \rightarrow Person(x))$	Premise	
[2]	$\forall x\ (Person(x) \rightarrow Breathes(x))$	Premise	
[1a]	$Student(c) \rightarrow Person(c)$	Universal Instantiation	[1]
[2a]	$Person(c) \rightarrow Breathes(c)$	Universal Instantiation	[2]
[3a]	$Student(c) \rightarrow Breathes(c)$	Hypothetical Syllogism	[1a], [2a]
[3]	$\forall x\ (Student(x) \rightarrow Breathes(x))$	Universal Generalization	[3a]

Notice that when we apply Universal Instantiation, we can pick any value we like (after all, the claim is *universal*. So, in particular, if we apply it twice, we can choose the same value both times. That's what we've done here. And that's often what we want to do.

So now we've seen that Universal Instantiation and Universal Generalization, taken together, give us a way to prove traditional syllogisms. Alternatively, we could have introduced a special syllogism rule. Many logicians have done that. We chose to go our way, however, because it does more than solve the syllogism problem. It's more general. The idea that we start a predicate logic proof by first moving into the Boolean world, doing what we need to do (however many steps that takes) and finally moving back to quantified statements is very powerful.

Suppose that we are given two premises:

[1] $\forall x\ ((P(x) \wedge Q(x)) \rightarrow R(x))$
[2] $\forall x\ (P(x))$

We want to prove: $\forall x\ (Q(x) \rightarrow R(x))$

This seems as though it ought to be true. If $P(x)$ is true of everything anyway, we shouldn't have to worry about it. But is that so? Let's see if we can prove it.

[1]	$\forall x\ ((P(x) \wedge Q(x)) \rightarrow R(x))$	Premise	
[2]	$\forall x\ (P(x))$	Premise	
[3]	$(P(c) \wedge Q(c)) \rightarrow R(c)$	Universal Instantiation	[1]

We've chosen c as an arbitrary element of the universe. This is allowed. We know nothing else about it.

While we're at it, let's also instantiate our other premise, [2]. This time, we won't instantiate to a new arbitrary element. We'll instantiate to the specific element c that we already have. This is allowed. If $P(x)$ is true for all x, it must, in particular, be true of c, regardless of what c is. Thus c is still an arbitrary element.

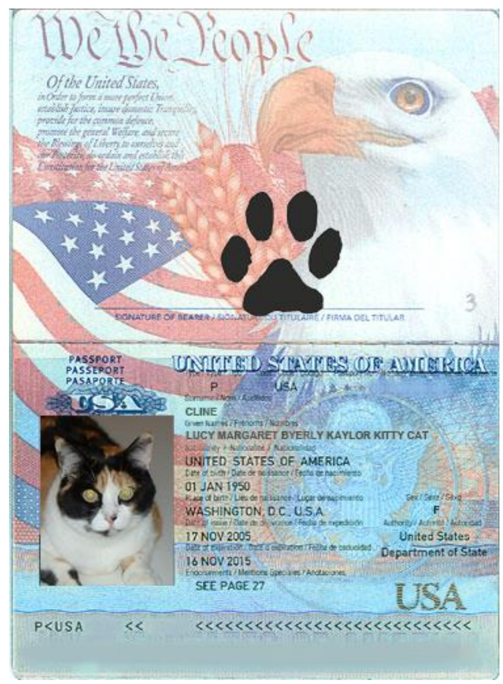
[4]	$P(c)$	Universal Instantiation	[2]

Now we can work with [3] and [4] completely in the Boolean world, applying as many identities and rules as we need.

[5]	$\neg(P(c) \wedge Q(c)) \vee R(c)$	Conditional Disjunction	[3]
[6]	$(\neg P(c) \vee \neg Q(c)) \vee R(c)$	De Morgan	[5]
[7]	$\neg P(c) \vee (\neg Q(c) \vee R(c))$	Associativity of *or*	[6]
[8]	$(\neg Q(c) \vee R(c)) \vee \neg P(c)$	Commutativity of *or*	[7]
[9]	$\neg Q(c) \vee R(c)$	Disjunctive Syllogism	[8], [4]
[10]	$Q(c) \rightarrow R(c)$	Conditional Disjunction	[9]

Now we're close to the claim we're trying to prove. The only issue is that we know that it's true of the arbitrary element c. We want to show that it must be true for any element. But that's easy; Universal Generalization will do that for us (precisely because c is arbitrary). So we have:

[11]	$\forall x\ (Q(x) \rightarrow R(x))$	Universal Generalization	[10]



We just proved our claim for arbitrary predicates *P*, *Q*, and *R*. But we can see why it might be useful if we give meanings to them. Define:

P(*x*): True if *x* is a person.
Q(*x*): True if *x* was born in the United States.
R(*x*): True if *x* is a citizen of the United States.

Then our original claim is that if you're a person and you were born in the U.S., you're a citizen of the U.S. Given an arbitrary universe, we need the restriction that you must be a person since cats and mice, although born in the U.S., are not citizens. But if we assume a universe of people (thus assuming that everything in question is a person), we can drop the explicit person requirement and say simply that if you're born in the U.S., you're a U.S. citizen.

Problems

1. Define: $P(x)$: x is Popular
$L(x, y)$: x Likes y

Fill in the ten blanks in the following proof:

[1]	$\forall x\ (\forall y\ (P(x) \rightarrow L(y, x)))$	Premise	
[2]	$P(Riley)$	Premise	
[3]	$P(Tracy)$	Premise	
[4]	$\forall y\ (P(Riley) \rightarrow L(y,$ ________{1}))	________________{2}	[1]
[5]	$P(Riley) \rightarrow L($________{3}, __________{4})	________________{5}	[4]
[6]	$L($________{6}, __________{7})	Modus Ponens	[2], [5]
[7]	$\forall y\ (L($________{8}, __________{9}))	________________{10}	[6]

Existential Instantiation

Existential Instantiation

$$\frac{\exists x\,(P(x))}{\therefore\; P(c^*/x)}$$

Restrictions:

- c^* must be a symbol that has not previously been used.
- If the quantified expression $\exists x\,(P(x))$ originally occurred inside the scope of one or more universal quantifiers that have already been instantiated then:

  If any of those universals have been instantiated to arbitrary elements, then we must describe c^* as depending on the values of those arbitrary elements. (See example below to make this clearer.)

This rule says that if P holds for some element of the universe, then we can give that element a name such as c^*. When selecting symbols, we must be careful: the symbols must be selected one at a time and must not duplicate any symbol that has already been selected within the same proof.

Thus, if $\exists x\,(P(x)) \wedge \exists y\,(Q(y))$ is true, then, to use this rule, we select a name, let's say c^*, for the object of which P must be true. Then we select a *different name*, let's say d^*, for the object of which Q, must be true.

This is what we have been calling new-rule-to-come-3.

To simplify our bookkeeping and to guarantee that we don't confuse the names we choose here for "the ones" with the names we choose for arbitrary elements when we apply *Universal* Instantiation, we'll always choose names (like c^* and d^*) that end with an asterisk.

Voters

Consider the following argument: If you are 18 years old then you can vote. Someone, call the person c, is 18 years old. Therefore c can vote.

Define:

$E(x)$: True if x is 18 years old.
$V(x)$: True if x can vote.

Then the proof proceeds as follows:

[1]	$\forall x\,(E(x) \rightarrow V(x))$	Premise	
[2]	$\exists x\,(E(x))$	Premise	
[3]	$E(c^*)$	Existential Instantiation	[2]
[4]	$E(c^*) \rightarrow V(c^*)$	Universal Instantiation	[1]
[5]	$V(c^*)$	Modus Ponens	[3], [4]

In step 3 above, a specific person with property *E* was given the name *c**. For that same person *c**, the statement $E(c^*) \rightarrow V(c^*)$ holds by Universal Instantiation. Note that the order of steps 3 and 4 cannot be reversed. We can do Existential Instantiation first and give the name *c** to "the one". Then we can use Universal Instantiation since, when we apply it, we may make the claim about *any* element and thus, in particular *c**. But, if we do Universal Instantiation first and choose *c** as the name of some arbitrary element, then that name has already been used. The best we'd be able to do, if we apply Existential Instantiation at that point, would be to choose a second name for "the one". But then we'd not be able to conclude anything about either *c** or (say) *d** because they'd be different.

Notice, in this example, that we've ended by concluding something (i. e., can vote) of some object *c** about which we know nothing else. But we do in fact know one more thing. We now know that there exists *someone* who can vote. We need one more rule, to be described next. It will let us take that step.

Problems

1. Consider the following beginning of a proof:

[1]	*Student*(*Blake*)	Premise	
[2]	*Cat*(*Lucy*)	Premise	
[3]	$\exists x\ (\exists y\ (Likes(x, y)))$	Premise	

Which of the following is/are legal possibilities for step [4]:

I.	[4]	$\exists y\ (Likes(c^*, y))$	Existential Instantiation	[3]
II.	[4]	$\exists y\ (Likes(Blake, y))$	Existential Instantiation	[3]
III.	[4]	$\exists x\ (Likes(x, Lucy))$	Existential Instantiation	[3]

2. Consider the following beginning of a proof:

[1]	*Student*(*Blake*)	Premise	
[2]	*Cat*(*Lucy*)	Premise	
[3]	$\exists x\ (\exists y\ (Likes(x, y)))$	Premise	
[4]	$\exists y\ (Likes(c^*, y))$	Existential Instantiation	[3]

Which of the following is/are legal possibilities for step [5]:

I.	[5]	$Likes(c^*, Lucy)$	Existential Instantiation	[4]
II.	[5]	$Likes(c^*, d^*)$	Existential Instantiation	[4]
III.	[5]	$Likes(c^*, c^*)$	Existential Instantiation	[4]

Skolem Functions

But first, before we introduce our last new rule, let's explain the second restriction on the use of this instantiation rule. Suppose that we began with the following expression, with respect to the universe of bears:

[1] $\forall x\ (\exists y\ (MotherOf(y, x)))$

If we wish to use the instantiation rules, we must begin at the outside (in other words, with the entire statement). So we first instantiate x. If we do that for a specific value, say Smokey, then we get:

[2] $\exists y\ (MotherOf(y, Smokey))$

Now we can give a name to the particular individual who is Smokey's mother. We can write:

[3] $MotherOf(c^*, Smokey)$

We haven't gotten into any trouble (yet).

But now suppose that we begin by instantiating x with an *arbitrary* element. Recall that we do this when we want to reason about that arbitrary element, typically with the intent of generalizing it later. When we do this, we can make no specific claims about the value that we choose. So, in particular, we can't claim that there's any one c^* who is the mother of any arbitrary bear. Clearly different bears can have different mothers.

We can still apply the Existential Instantiation rule. But we must make clear that the value we're asserting the existence of depends on the value that is chosen for x. To do that we'll use the notation:

$f(x)$,

which we'll read as, "f of x", meaning "a value that depends on x".

So, suppose that we'd used Universal Instantiation on [1] to produce:

[4] $\exists y\ (MotherOf(y, a))$

Then we can use Existential Instantiation to produce:

[5] $MotherOf(f(a), a)$

So we have that, no matter who a is, there's something, whose value depends on a, who is a's mother.

What if the original existentially quantified expression occurred inside the scope of two or more universal quantifiers? Easy. We simply make the instantiated value depend on all of them.

Suppose for example, that we want to claim that, for every pair of Facebook users, there's a definition of their Facebook relationship. (This definition includes whether they are friends, whether one has hidden the other, etc.) Then (assuming a universe of Facebook users) we could write:

[6]	$\forall x\ (\forall y\ (\exists z\ (FBRelDef(z, x, y))))$

After applying Universal Instantiation twice, we will have:

[7]	$z\ (FBRelDef(z, a, b))$

Then we apply Existential Instantiation and we get the fact that the Facebook relationship that exists for *a* and *b* depends on who *a* and *b* are:

[8]	$FBRelDef(f(a, b), a, b)$

https://www.youtube.com/watch?v=MFw3yIk_KCQ

Nifty Aside

Objects such as *c**, are sometimes called ***Skolem constants***. Objects such as *f*(*a*) and *f*(*a*, *b*) are sometimes called ***Skolem functions***. While it's true that these objects are clunky for us to reason with, they play a very important role in some computational logic systems (i.e., programs that produce proofs).

Problems

1. Given:

[1] $\forall x\ (\exists y\ (\forall z\ (P(x, y, z))))$

Which of the following statements could result from a correct application of the Instantiation rules that we have presented? Assume that all uses of Universal Instantiation derive arbitrary elements.

a) $P(a, b^*, c)$
b) $P(a, f(a), c)$
c) $P(a, f^*(a,), c)$
d) $P(a, b^*, a)$

Existential Generalization

Existential Generalization	$P(c/x)$	*P* is true for the specific value *c* (which appears everywhere a more general value *x* might appear).
	$\therefore\ \exists x\,(P(x))$	There is some value of *x* for which *P* is true.

Restriction: *x* does not appear free in $P(c)$.

This is perhaps the simplest of all of these rules. What it says is that if there is some element *c* in the universe that has the property *P*, then we can say that there exists *something* in the universe that has the property *P*.

This is what we have been calling new-rule-to-come-4.

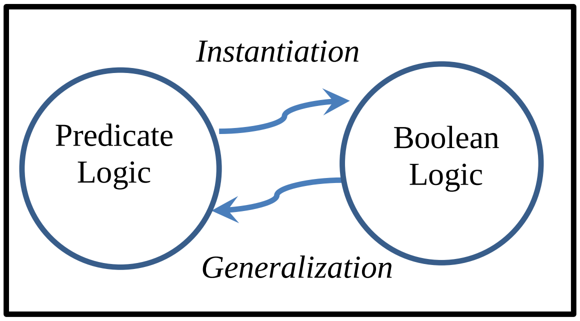
Note that *c* must be a specific value. It cannot be a function (as described on the previous slide) of some other value.

Now let's continue the Voting example. We'd like to show that there is *someone* who can vote. All we need to do is to add line [6]:

[1]	$\forall x\,(E(x) \to V(x))$	Premise	
[2]	$\exists x\,(E(x))$	Premise	
[3]	$E(c^*)$	Existential Instantiation	[2]
[4]	$E(c^*) \to V(c^*)$	Universal Instantiation	[1]
[5]	$V(c^*)$	Modus Ponens	[3], [4]
[6]	$\exists x\,(V(x))$	Existential Generalization	[5]

Summary of the New Rules

Here's one way to think about how we use these instantiation and generalization rules:



https://www.youtube.com/watch?v=d8XbF8LxGgI

We instantiate. Then we work in the simpler world of Boolean logic. Then we generalize at the end.

Here are some important ideas to keep in mind as you're doing this:

- Quantifier Exchange is an equivalence rule. So it can be used anywhere in an expression. (In other words, it's fine to apply it to an entire expression, but application to smaller subexpressions is also allowed.)

> Suppose that we are given:
>
> $\forall x\ (\neg\exists y\ (P(x, y)))$
>
> We can apply Quantifier Exchange to the existential quantifier that occurs inside the larger, universally quantified expression. If we do that, we get:
>
> $\forall x\ (\forall y\ (\neg P(x, y)))$

- Inference rules (unlike equivalence rules) can be applied only to entire statements, not to subexpressions within statements. The instantiation and generalization rules are inference rules.

- So, if we want to instantiate (i.e., remove a quantifier), we must work from the outside in.

Suppose that we are given:

$\forall x\ (\exists y\ (P(x, y)))$

We can apply Universal Instantiation to the entire expression. We'll get:

$\exists y\ (P(c, y))$

But we may not apply Existential Instantiation to the inner subexpression without getting rid of the outer quantifier first.

- On the other hand, if we want to generalize (i.e., add a quantifier), we must work from the inside out.

Suppose that we had again started with:

[1] $\forall x\ (\exists y\ (P(x, y)))$

Then we applied Universal Instantiation to the entire expression, getting rid of x and yielding:

[2] $\exists y\ (P(c, y))$

Then we applied Existential Instantiation to that, getting rid of *y* and yielding:

[3] $P(c, f(c))$

(Recall that since the existentially quantified *y* occurs inside the scope of the universally quantified x, we must make our instantiated *y* depend on particular values of x.)

Now suppose that we want to generalize back. (In a real proof, we'd do some work first, but we're simplifying here.)

We can apply Universal Generalization to this, (replacing *c* with x) giving us:

[4] $\forall x\ (P(x, f(x))$

But now we're stuck. We cannot reach inside this expression to do Existential Generalization. If we want to do that, we must do it first, starting from [3], and writing:

[5] $\exists y\ (P(c, y))$

And then we can apply Universal Generalization and get [1] back.

- Because of the requirement that Existential Instantiation must use a previously undefined element (but Universal Instantiation does not have that requirement), it generally makes sense to do Existential Instantiations first.

Let's return to the <ex Voters> example. Recall that we had:

[1]	$\forall x\ (E(x) \rightarrow V(x))$	Premise	
[2]	$\exists x\ (E(x))$	Premise	
[3]	$E(c^*)$	Existential Instantiation	[2]
[4]	$E(c^*) \rightarrow V(c^*)$	Universal Instantiation	[1]
[5]	$V(c^*)$	Modus Ponens	[3], [4]
[6]	$\exists x\ (V(x))$	Existential Generalization	[5]

We did Existential Instantiation first and gave a name, c*, to the object that must exist. Then, since a universal claim is true of everything, we asserted that it must, in particular, be true of c*.

Suppose, on the other hand, we'd tried to use Universal Instantiation first. Then we'd have:

[1]	$\forall x\ (E(x) \rightarrow V(x))$	Premise	
[2]	$\exists x\ (E(x))$	Premise	
[3]	$E(c) \rightarrow V(c)$	Universal Instantiation	[1]

At this point, we've picked someone, *c*, to make the claim about. If there were additional universally quantified statements, we could also assert them of *c*. But there aren't. There is, however, an existentially quantified one. But to use it, we'll have to pick a new name for the object that we know must exist:

[4]	$E(d^*)$	Existential Instantiation	[2]

Oops. We can't combine [3] and [4] to conclude anything.

- Some of our new predicate logic rules are natural generalizations, to a possibly infinite domain, of Boolean logic rules that we already had:

  - The quantifier exchange rules generalize the Boolean De Morgan's laws. We've already seen why that's so.

  - Universal Instantiation generalizes Boolean Simplification. Recall that the quantifier ∀ can be thought of as another way of writing a large (possibly even infinite) conjunction. So we have this generalization of Simplification:

$$\frac{P(x_1) \wedge P(x_2) \wedge P(x_3) \wedge \ldots}{\therefore\ P(x_k)} \qquad \text{written as: } \forall x\ (P(x))$$

    If I know that some claim is true for all values, then I can conclude that it must be true for any individual one.

  - Universal Generalization generalizes Boolean Conjunction: Again, recall that ∀ can be thought of as another way of writing a large (possibly even infinite) conjunction. So we have this generalization of Conjunction:

$$\begin{array}{lll} P(c) \quad \rightarrow & & P(x_1) \\ & & P(x_2) \\ & & \ldots \\ \hline & \therefore & P(x_1) \wedge P(x_2) \wedge \ldots \end{array} \qquad \text{written as: } \forall x\,(P(x))$$

If I know that some claim is true for some arbitrary individual *c*, then I know that it must be true for individual_1 and individual_2 and so forth. Thus it is true of all individuals. Note the sense in which Universal Instantiation and Universal Generalization are inverses of each other.

- Existential Generalization generalizes Boolean Addition: Recall that the quantifier ∃ can be thought of as another way of writing a large (possibly even infinite) disjunction So we have this generalization of Addition:

$$\begin{array}{ll} & P(x_1) \\ \hline \therefore & P(x_1) \vee P(x_2) \vee P(x_3) \ldots \end{array} \qquad \text{written as: } \exists x\,(P(x))$$

If I know that some claim is true for one individual then I know that it is true of at least one element out of some possibly infinite set.

- Existential Instantiation doesn't generalize any of our Boolean rules, but it's interesting nevertheless to write it out in an analogous way:

$$\begin{array}{ll} & P(x_1) \vee P(x_2) \vee P(x_3) \ldots \\ \hline \therefore & P(x_k) \text{ for } \textit{some } x_k \end{array} \qquad \text{written as: } \exists x\,(P(x))$$

If I know that there exists some individual of whom some claim is true, then there must be at least one specific one of whom it's true. Note the sense in which Existential Instantiation and Existential Generalization are inverses of each other.

Big Idea

In the appendix, you'll find a Predicate logic "cheat sheet". You may want to keep it handy while working proofs.

Checking Predicate Logic Proofs

StepWise (the proof checking tool that we used for Boolean logic) also works for predicate logic proofs.

https://www.youtube.com/watch?v=4RU9po49OcE

We'll soon see how to design nontrivial proofs using all of the rules at our disposal. But, before we do that, let's practice using the instantiation and generalization rules that we've just learned.

Problems

1. Assume the following premises:

[1] $\forall x\, (Dragon(x) \rightarrow SpewsFire(x))$ All dragons spew fire.
[2] $\exists x\, (Dragon(x))$ Dragons exist.

Prove:

$\exists x\, (SpewsFire(x))$ Fire spewing creatures exist.

2. Assume the following premises:

[1] $\neg\exists x\, (Dragon(x) \wedge PetOwner(x))$ No dragons own pets.
[2] $\forall x\, (\neg PetOwner(x) \rightarrow Grumpy(x))$ Anyone without a pet is grumpy.

Prove:

$\forall x\, (Dragon(x) \rightarrow Grumpy(x))$ All dragons are grumpy.

3. Let the universe over which $\forall x$ quantifies be the set of dragons. Assume the following premise:

$\forall x\, (\exists y\, (Tail(y) \wedge HasPart(x, y)))$ Every dragon has a tail.

Here's a proposed proof that all dragons share the same tail:

[1]	$\forall x\, (y\, (Tail(y) \wedge HasPart(x, y)))$	Premise	
[2]	$\exists y\, (Tail(y) \wedge HasPart(d, y))$	Universal Instantiation	[1]
[3]	$Tail(t^*) \wedge HasPart(d, t^*)$	Existential Instantiation	[2]
[4]	$\forall x\, (Tail(t^*) \wedge HasPart(x, t^*))$	Universal Generalization	[3]
[5]	$\exists t\, (\forall x\, (Tail(t) \wedge HasPart(x, t)))$	Existential Generalization	[4]

Clearly it does not follow from our premise that all dragons share the same tail. So there's at least one thing wrong with this proof. In which line did we make our first mistake? You should be able to explain exactly what we did wrong.

a) Line [2]
b) Line [3]
c) Line [4]
d) Line [5]

Creating Predicate Logic Proofs I

Overview

Just as we did in Boolean logic, we're now going to walk through the process of constructing natural deduction proofs for a collection of representative examples. There's a video for each of them.

For each of these problems, we suggest that you first try to do the proof yourself. Then you can watch a video in which we walk through the construction of a proof.

Universal Generalization Proof Problem: The Barber

The Barber

Assume that whenever we have a barber and a person who needs a haircut then the barber can cut the person's hair. Let's show that any barber who needs a haircut could cut his/her own hair. (But notice that, in the real world, self barbering is actually hard. Maybe what we really should do is to change our premise to prevent drawing our conclusion.)

Assign the following names to basic statements:

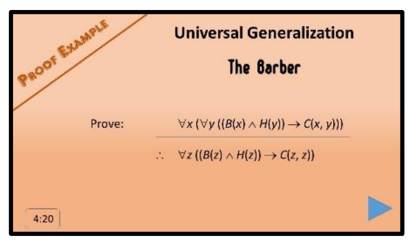
$B(x)$: True if x is a barber.
$H(x)$: True if x needs a haircut.
$C(x, y)$: True if x can cut y's hair.

Prove: $\forall x\ (\forall y\ ((B(x) \wedge H(y)) \rightarrow C(x, y)))$ — Whenever we have a barber and a person who needs a haircut then the barber can cut the person's hair.

$\therefore \forall z\ ((B(z) \wedge H(z)) \rightarrow C(z, z))$ — Any barber who needs a haircut can cut his or her own hair.

You should do this proof yourself.

You can also watch our video, which will outline our strategy for doing this.



https://www.youtube.com/watch?v=lxNkUWim2I8

Problems

1. ***Amazed Teacher*** : Assume the following premise:

[1] $(\forall x\ (Student(x) \rightarrow CompletedHomework(x))) \rightarrow Amazed(Teacher)$

Assume that we want to prove:

$Amazed(Teacher)$

Which of the following statements, would if combined with [1], be sufficient to enable us to prove *Amazed*(*Teacher*)?

I. $Student(Flopsy)$
II. $CompletedHomeworks(Flopsy)$
III. $\forall x\ (CompletedHomework(x))$

a) I and II are sufficient.
b) III is sufficient.
c) All of them are required.
d) No combination is sufficient.

2. Let's continue the ***Amazed Teacher*** example. We'll abbreviate *CompletedHomework*(*x*) as *Done*(*x*). Assume the following premises:

[1] $(\forall x\ (Student(x) \rightarrow Done(x))) \rightarrow Amazed(Teacher)$
[2] $\forall x\ (Done(x))$

Prove *Amazed*(*Teacher*).

3. ***NotMotherOfSelf*** **:** Assume the following premise:

[1] $\forall x\ (\forall y\ (MotherOf(x, y) \rightarrow \neg MotherOf(y, x)))$

Prove: $\forall x\ (\neg MotherOf(x, x))$

3. Assume the following premises:

[1] $\forall x\ (P(x) \vee R(x))$
[2] $\forall x\ ((P(x) \vee (Q(x) \vee R(x))) \rightarrow W(x))$

Prove: $\forall x\ (W(x) \vee S(x))$

4. Assume: [1] $\forall x\ (P(x) \vee R(x))$
[2] $\forall x\ ((P(x) \vee (Q(x) \vee R(x))) \rightarrow W(x))$

Prove: $\forall x\ (W(x) \vee S(x))$

Existential Instantiation Proof Problem: Absolute Value

Absolute Values

Suppose our universe is the set of integers and we assume that every integer has both an absolute value and another number that is its successor. Thus every integer has an absolute value.

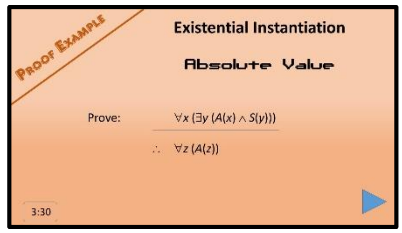
Assign the following names to basic statements:

$A(x)$: True if x has an absolute value.
$S(x)$: True if x has a successor.

Prove: $\forall x\ (\exists y\ (A(x) \wedge S(y)))$ Every integer has an absolute value and a successor.

$\therefore\ \forall z\ (A(z))$ Every integer has an absolute value.

You should do this proof yourself. (Hint: Be careful. Notice that the existentially quantified variable occurs within the scope of a universally quantified one. You will need to instantiate y as a function of the variable you used when you instantiated x.)



You can also watch our video, which will outline our strategy for doing this.

https://www.youtube.com/watch?v=gogJe8bMGj8

Problems

1. Assume the following premise:

[1] $\exists x\ (\forall y\ (R(x, y)))$ Premise

Prove: $\forall y\ (\exists x\ (R(x, y)))$

Universal Instantiation Proof Problem: Fathers and Sons

FathersAndSons

Assume the universe of male people. We assume that if anyone is the father of some person then that father cannot be the son of his own son. Thus no one is his own father.

Assign the following names to basic statements:

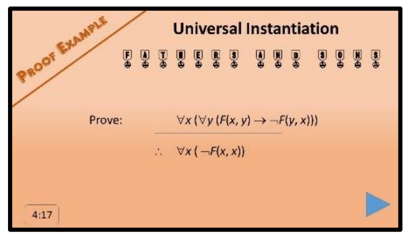
$F(x, y)$: True if x is the father of y.

Prove: $\forall x\ (\forall y\ (F(x, y) \rightarrow \neg F(y, x)))$ If anyone is the father of some person then that father cannot be the son of his own son.

$\therefore\ \forall x\ (\neg F(x, x))$ No one is his own father.

You should do this proof yourself.

You can also watch our video, which will outline our strategy for doing this.



https://www.youtube.com/watch?v=uLu8ihQz510

Quantifier Exchange Proof Problem: First Class

FirstClass

Assume that if anyone has a coach class ticket but there is no coach seat available then that passenger gets a first class ticket. Alice has a coach class ticket. But every coach seat is one that cannot be occupied by Alice. Thus, Alice gets a first class ticket.

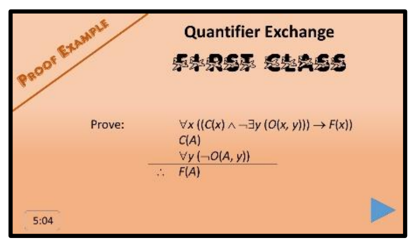
Assign the following names to basic objects and statements:

A:	**A**lice
C(*x*):	True if *x* has a **C**oach class ticket.
O(*x*, *y*)	True if Person *x* may **O**ccupy seat *y*.
F(*x*)	True if *x* gets a **F**irst class ticket.

Prove: $\forall x\ ((C(x) \wedge \neg\exists y\ (O(x, y))) \rightarrow F(x))$	If anyone has a coach class ticket but there is no coach seat then that passenger gets a first class ticket.
$C(A)$	Alice has a coach class ticket.
$\forall y\ (\neg O(A, y))$	Every coach seat is one that cannot be occupied by Alice.
$\therefore\ F(A)$	Alice gets a first class ticket.

You should do this proof yourself.

You can also watch our video, which will outline our strategy for doing this.



https://www.youtube.com/watch?v=G1TlZ7QNoak

Problems

1. ***Neighborhood Gangs*** : Assume the following premise:

[1] $(\exists x\ (MemberOfGangWeHate(x) \wedge InOurNeighborhood(x)\ )) \rightarrow HighAlert$

(By the way, notice the predicate *HighAlert*. It looks like a Boolean proposition. It is. There's no reason that we can't, in predicate logic, use Boolean propositions. Think of them as predicates that don't take any arguments.)

Suppose that we want to prove:

HighAlert

(Part I) Which of the following statements, would if combined with [1], be sufficient to enable us to prove *HighAlert*?

I. $MemberOfGangWeHate(OneEye)$
II. $MemberOfGangWeHate(Bugsy)$
III. $InNeighborhood(Lefty)$
IV. $InNeighborhood(Bugsy)$
V. $\forall x\ (InNeighborhood(x))$

a) I and III together is the only sufficient combination.
b) II and IV together is the only sufficient combination.
c) I and V together is the only sufficient combination.
d) I and V together is a sufficient combination but there are other sufficient combinations of these premises.
e) No combination of these premises is sufficient.

2. Let's return to the ***Neighborhood Gangs*** problem. We'll abbreviate *MemberOfGangWeHate* as *Member* and *InOurNeighborhood* as *In*. Prove *HighAlert* assuming the following premises:

[1]	$(\exists x\ (Member(x) \wedge In(x)\ )) \rightarrow HighAlert$	Premise
[2]	$Member(Bugsy)$	Premise
[3]	$In(Bugsy)$	Premise

3. Back to the ***Neighborhood Gangs*** problem again. This time, prove *HighAlert* using the following premises:

[1]	$(\exists x\ (Member(x) \wedge In(x)\ )) \rightarrow HighAlert$	Premise
[2]	$Member(OneEye)$	Premise
[3]	$\forall x\ (In(x))$	Premise

Creating Predicate Logic Proofs II

This section contains some additional examples, including some with videos.

Existential Instantiations, Then Universal Ones

Recall that, whenever we use Existential Instantiation, we must instantiate to some previously undefined element. Yes, we know that some element exists. But we know nothing else about it. But this isn't so for Universal Instantiation. When we have a universal statement, its claim applies to all elements, and so, in particular, to any specific element we may be working with. Because of this difference, it generally makes sense, when we have a choice, to do Existential Instantiations first.

White Mammals

Suppose that we want to prove that there exists a white mammal. We have two premises:

$\forall x\ (Bear(x) \rightarrow Mammal(x))$	All bears are mammals.
$\exists x\ (Bear(x) \wedge White(x))$	There exists a white bear.

We can prove our claim as follows:

[1] $\forall x\ (Bear(x) \rightarrow Mammal(x))$	Premise	
[2] $\exists x\ (Bear(x) \wedge White(x))$	Premise	
[3] $Bear(c^*) \wedge White(c^*)$	Existential Instantiation [2]	
[4] $Bear(c^*)$	Simplification	[3]
[5] $White(c^*)$	Simplification	[3]
[6] $Bear(c^*) \rightarrow Mammal(c^*)$	Universal Instantiation	[1]
[7] $Mammal(c^*)$	Modus Ponens	[4], [6]
[8] $Mammal(c^*) \wedge White(c^*)$	Conjunction	[5], [7]
[9] $\exists x\ (Mammal(x) \wedge White(x))$	Existential Generalization	[8]

We used Existential Instantiation in step [3], chose the name c*, and then used Universal instantiation to state the general claim in [1] as a specific claim about c*. If we'd tried to do those two instantiations in the opposite order, we'd have gotten stuck:

[1] $\forall x\ (Bear(x) \rightarrow Mammal(x))$	Premise	
[2] $\exists x\ (Bear(x) \wedge White(x))$	Premise	
[3] $Bear(d) \rightarrow Mammal(d)$	Universal Instantiation	[1]
[4] $Bear(c^*) \wedge White(c^*)$	Existential Instantiation	[2]

Oops. We can't combine what we know about *d* with what we know about c*.

Existential Instantiation and Generalization Proof Problem: Drug Test

DrugTest

Assume that if there is any woman who uses any medication then that woman may participate in our drug test. There is a woman who uses some medication. Thus, someone may participate in our drug test.

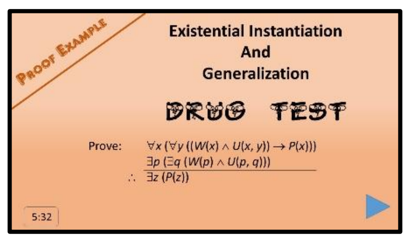
Assign the following names to basic statements:

$W(x)$:	True if x is a **W**oman.
$U(x, y)$	True if person x **U**ses medication y.
$P(x)$	True if x may **P**articipate in our drug test.

Prove:	$\forall x\ (\forall y\ ((W(x) \wedge U(x, y)) \rightarrow P(x)))$	If there is any woman who uses any medication then that woman may participate in our drug test.
	$\exists p\ (\exists q\ (W(p) \wedge U(p, q)))$	There is a woman who uses some medication.
	$\therefore\ \exists z\ (P(z))$	Someone may participate in our drug test.

You should do this proof yourself.

You can also watch our video, which will outline a strategy for creating a proof.



https://www.youtube.com/watch?v=BfZ1EU3daxs

Contradictory Premises Proof Problem: Funny

Assume that anyone who is funny is not sad. There is someone who is sad while everyone is funny. Thus, someone is richer than everyone.

Assign the following names to basic statements:

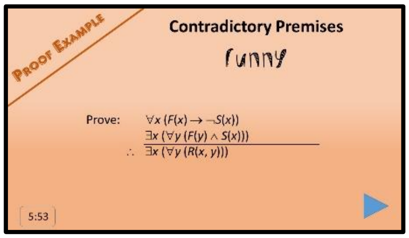
$F(x)$: True if x is **F**unny.
$S(x)$ True if x is **S**ad.
$R(x, y)$ True if x is **R**icher than y.

Prove:	$\forall x\,(F(x) \rightarrow \neg S(x))$	Anyone who is funny is not sad.
	$\exists x\,(\forall y\,(F(y) \wedge S(x)))$	There is someone who is sad while everyone is funny.
	$\therefore\ \exists x\,(\forall y\,(R(x, y)))$	Someone is richer than everyone.

Yes, you are reading this correctly. The premises are about being funny and being sad, and the conclusion is about what appears to be something completely unrelated: wealth. Recall what we learned in Boolean logic about proving something that appears to come out of the blue.

You should do this proof yourself.

You can also watch our video, which will outline a strategy for creating a proof.



https://www.youtube.com/watch?v=3gabk6libRA

Quantifier Exchange Proof Problem: Asleep in Class

Assume that Alice is in class. If Tom is asleep then it is not true that Alice is in class and Bob did his homework. If anyone is in class then everyone did the homework. Thus, it isn't true that everyone is asleep.

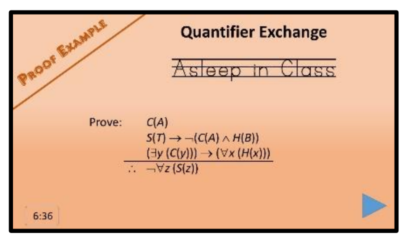
Assign the following names to basic objects and statements:

A, B, T:	**A**lice, **B**ob, **T**om, respectively.
$C(x)$:	True if x is in **C**lass.
$H(x)$	True if x did his or her **H**omework.
$S(x)$	True if x is a**S**leep.

Prove:	$C(A)$	Alice is in class.
	$S(T) \rightarrow \neg(C(A) \wedge H(B))$	If Tom is asleep then it is not true that Alice is in class and Bob did his homework.
	$(\exists y\ (C(y))) \rightarrow (\forall x\ (H(x)))$	If anyone is in class then everyone did the homework.
	$\therefore\ \neg\forall z\ (S(z))$	It isn't true that everyone is asleep.

You should do this proof yourself. You will notice that, since T stands for true in that system, the problem uses M in place of T.

You can also watch our video, which will outline a strategy for creating a proof.



https://www.youtube.com/watch?v=OWWGmMwFL24

Problems

1. Prove that this claim holds with no premises required:

$$(\exists x\ (\forall y\ (R(x, y)))) \rightarrow\ (\forall u\ (\exists v\ (R(v, u))))$$

Hint: Recall the Conditionalization inference rule that we defined in Boolean Logic:

$$\begin{array}{ll} & A\text{, a set of premises} \\ & (A \wedge p) \text{ entails } q \\ \hline \therefore & p \rightarrow q \end{array}$$

To use the rule, we assume whatever we want. Call it p. We prove that, with that assumption, along with whatever other premises we have (call them A), q must follow. We can then assert:

$$p \rightarrow q$$

This rule is useful in proving claims (as in this problem) of the form: $p \rightarrow q$.

(Here, we'll just let A be the empty set of premises.) Remember that, to use this rule correctly, we begin by assuming p. We must then later (often at the end), "discharge" p by making it the antecedent of an implication.

2. Assume the following premises:

[1] $\exists y\ (P(y)) \rightarrow \forall x\ (Q(x))$
[2] $P(a)$
[3] $R(c) \rightarrow \neg(P(a) \wedge Q(b))$

Prove: $\neg\forall z\ (R(z))$.

3. Assume the following premises:

[1] $\forall x\ (\forall y\ ((G(x) \vee N(x, y)) \rightarrow L(y)))$
[2] $\exists u\ (\exists v\ (G(u) \wedge N(u, v)))$

Prove: $\exists z\ (L(z))$

Lemmas

Recall that a lemma is really just a theorem. But we typically call a theorem a lemma when the reason that we've proved it is that it's a useful intermediate step. Once it's proved, we can use it as often as we like in other proofs. The idea of proving lemmas so that we can save a lot of repeated work can be very useful. We saw several examples of this in our discussion of Boolean logic. Lemmas can be even more useful when we're working in predicate logic because there may be many more steps involved in our proofs.

Wine and Chocolate

Define:

Wine(x):	True if x is a wine lover.
Choc(x):	True if x is a chocolate lover.
Food(x):	True if x loves food.
Drink(x):	True if x loves drinks.
Paris(x):	True if x loves Paris.

Suppose that we're given the following premises:

$\forall x\ (Choc(x) \rightarrow Food(x))$	Chocolate lovers are food lovers.
$\forall x\ (Wine(x) \rightarrow Drink(x))$	Wine lovers are drink lovers.
$\forall x\ ((Drink(x) \wedge Food(x)) \rightarrow Paris(x))$	Anyone who loves food and drink loves Paris.

We're also told about a lot of people who love wine and chocolate. We'd like to be able to peg all of them as Paris lovers. The easiest way to do that will be to prove a lemma:

$\forall x\ ((Wine(x) \wedge Choc(x)) \rightarrow Paris(x))$

Then we can go directly to Paris-loving for everyone who's a lover of wine and chocolate. This is a particularly good thing to do because proving the lemma, while not hard, is a bit annoying. We don't want to have to do that reasoning more than once. So here it is (once):

[1]	$\forall x\ (Choc(x) \rightarrow Food(x))$	Premise
[2]	$\forall x\ (Wine(x) \rightarrow Drink(x))$	Premise
[3]	$\forall x\ ((Drink(x) \wedge Food(x)) \rightarrow Paris(x))$	Premise

We'll use the Conditionalization inference rule. So we'll assume the antecedent of our goal. More specifically, we'll assume that it's true of some arbitrary individual c. At the end of the proof, we can generalize it to all individuals.

[4]	$(Wine(c) \wedge Choc(c))$	(Conditional) Premise	
[5]	$Choc(c)$	Simplification	[4]
[6]	$Choc(c) \rightarrow Food(c)$	Universal Instantiation	[1]
[7]	$Food(c)$	Modus Ponens	[5], [6]
[8]	$Wine(c)$	Simplification	[4]
[9]	$Wine(c) \rightarrow Drink(c)$	Universal Instantiation	[2]
[10]	$Drink(c)$	Modus Ponens	[8], [9]
[11]	$(Drink(c) \wedge Food(c)) \rightarrow Paris(c)$	Universal Instantiation	[3]
[12]	$(Drink(c) \wedge Food(c))$	Conjunction	[10], [7]
[13]	$Paris(c)$	Modus Ponens	[12], [11]

[14]	$(Wine(c) \wedge Choc(c)) \rightarrow Paris(c)$	Conditional Discharge	[4], [13]
[15]	$\forall x\ ((Wine(x) \wedge Choc(x)) \rightarrow Paris(x))$	Universal Generalization	[14]

Whew. Done. Now we can identify a lot of Paris lovers very easily.

Soundness/Completeness/Decidability

As in Boolean logic, we will say that an inference rule is ***sound*** if, whenever it is applied to a set *P* of statements (premises), any conclusion that it produces is entailed by *P* (i.e., it must be true whenever *P* is). Clearly we are willing to accept only rules that are sound (and all of ours are). We haven't worked this hard to set up a system that lets us prove claims that aren't true. Of course, we could make that happen by having very weak rules. Why go out on any limbs?

But that won't do. We need something else as well. We'll say that a set of inference rules *R* is ***complete*** if (and only if), given any set *P* of premises (axioms), all statements that are entailed by *P* can be proved by applying the rules in *R*. In other words, if some statement *S* has to be true whenever all the premises in *P* are true, then it must be possible, using the rules in *R*, to prove it. We want a set of rules powerful enough to let us prove every theorem that follows from our premises.

In fact, what we'd hope for is all three of the following things (assume that *P* is a set of premises, *S* is a statement, and *R* is a set of inference rules):

(1) A guarantee that, if *S* is a theorem, given premises *P*, then we have the tools we need to prove it. In other words, we have a complete set *R* of inference rules. And, of course, *R* can't "overprove": every rule in *R* must be sound.

(2) A guarantee that, if *S* is actually *true* in some world that we're trying to describe, then we can come up with a set *P* of premises such that *S* is a theorem given *P*. In other words, all true things are theorems. Also, of course, all false things aren't theorems.

(3) An effective procedure (think computer program or algorithm) that can look at *S* and be guaranteed to answer correctly the question, "Is *S* a theorem given *P*?"

Okay, good news first:

- We have (1). We can't just use truth tables, the way we did in Boolean logic. But there are rule sets that work. This was proved by the Austrian/American mathematician Kurt Gödel in 1929. (So this isn't obvious; for example it wasn't something that the ancient Greeks knew.) This result is called Gödel's Completeness Theorem.

And now for the bad news:

- We don't have (2). Kurt Gödel (the very man who proved the completeness result that we just mentioned), went on to prove aa pair of Incompleteness Theorems that tell us that there are interesting systems (think arithmetic) that have the property that there's no set of axioms that will make all true statements theorems while at the same time avoiding inconsistencies.

  Visit *http://en.wikipedia.org/wiki/G%C3%B6del%27s_incompleteness_theorems* for more on this fundamental theorem that shook the mathematical world in 1931.

- We don't have (3) either. We know (from the Completeness Theorem) that, if *S* is a theorem, there's a way to find a proof of it. But what if it isn't? Can we assert that we tried really hard to find a proof but we failed? Does that mean that *S* isn't a theorem? No it doesn't. In some cases (in other words, given some sets of axioms) it's possible to do this. But in some other cases, it isn't. This claim, that there's no effective procedure for deciding whether *S* is a theorem is called the ***undecidability*** of predicate logic. This claim was proved in another fundamental, ground-shaking theorem in 1936, this time by the British mathematician Alan Turing. And, by the way, Turing's proof of this particular undecidability result laid the groundwork for a much broader set of undecidability proofs. There's a whole class of problems that will not *ever* be able to be solved by any sort of computer. Speed doesn't matter. Memory size doesn't matter. The existence of a solution to any one of these problems creates a logical contradiction.

Big Idea

There are provable limits to what predicate logic can do for us. And there are provable limits to what we can compute. Sometimes, in trying to solve real problems, those limits bite us. Nevertheless, the logical system that we've just defined can serve us well when we understand it.

Nifty Aside

Alan Turing is a hero to computer scientists. He worked on the earliest computers. He developed a machine that cracked German U-boat codes during the World War II. He proved key theoretical results (like the one we just mentioned). He wrote one of the first computer chess programs. And he had a sad death when he was way too young. Visit http://www.turing.org.uk/ for more on him and his work. Also, search YouTube for videos about him.

One last thing: Do not be confused by the fact that there exists both a Completeness Theorem and an Incompleteness Theorem. The terminology is unfortunate since it is based on two different notions of completeness. The Completeness Theorem tells us that there exist *rule* sets that are complete (in the sense we defined at the top of this page). The Incompleteness Theorem tells us that we may always be stuck with a *theorem* set that is incomplete in the sense that it doesn't completely cover the set of true statements.

Problems

1. A set of inference rules that allows us to derive *all and only* claims that must be true whenever the premises are true is said to be:

a) sound
b) complete
c) both
d) something else but not either of these

2. Alan Turing is famous for many seminal contributions to computer science. One is his attempt to define what it would mean for a computer to be intelligent. A version of his proposed "Imitiation Game" is now called the Turing Test. You can read about it here: http://en.wikipedia.org/wiki/Turing_test In his 1950 paper (http://orium.pw/paper/turingai.pdf), Turing famously predicted, "I believe that in about fifty years' time it will be possible, to programme computers, with a storage capacity of about 10^9, to make them play the imitation game so well that an average interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning."

Was Turing right? There are many modern chatbots that play the imitation game. One interesting one is Alice (http://alice.pandorabots.com/). Try having a conversation with Alice.

Appendices

Boolean Identities

Double Negation:	p	$\equiv$	$\neg(\neg p)$
Equivalence:	$(p \equiv q)$	$\equiv$	$(p \rightarrow q) \wedge (q \rightarrow p)$
Idempotence:	$(p \wedge p)$	$\equiv$	p
	$(p \vee p)$	$\equiv$	p
DeMorgan1:	$(\neg(p \wedge q))$	$\equiv$	$(\neg p \vee \neg q)$
DeMorgan2:	$\neg(p \vee q)$	$\equiv$	$(\neg p \wedge \neg q)$
Commutativity of *or*:	$(p \vee q)$	$\equiv$	$(q \vee p)$
Commutativity of *and*:	$(p \wedge q)$	$\equiv$	$(q \wedge p)$
Associativity of *or*:	$(p \vee (q \vee r))$	$\equiv$	$((p \vee q) \vee r)$
Associativity of *and*:	$(p \wedge (q \wedge r))$	$\equiv$	$((p \wedge q) \wedge r)$
Distributivity of *and* over *or*:	$(p \wedge (q \vee r))$	$\equiv$	$((p \wedge q) \vee (p \wedge r))$
Distributivity of *or* over *and*:	$(p \vee (q \wedge r))$	$\equiv$	$((p \vee q) \wedge (p \vee r))$
Conditional Disjunction:	$(p \rightarrow q)$	$\equiv$	$(\neg p \vee q)$
Contrapositive:	$(p \rightarrow q)$	$\equiv$	$(\neg q \rightarrow \neg p)$

Boolean Inference Rules

Modus Ponens:	From p and $p \rightarrow q$,	infer q
Modus Tollens:	From $p \rightarrow q$ and $\neg q$,	infer $\neg p$ . . .
Disjunctive Syllogism:	From $p \vee q$ and $\neg q$,	infer p . . .
Simplification:	From $p \wedge q$,	infer p . . .
Addition:	From p,	infer $p \vee q$ . . .
Conjunction:	From p and q,	infer $p \wedge q$
Hypothetical Syllogism:	From $p \rightarrow q$ and $q \rightarrow r$,	infer $p \rightarrow r$
Contradictory Premises:	From p and $\neg p$,	infer q
Resolution:	From $p \vee q$ and $\neg p \vee r$,	infer $q \vee r$ . . .
Conditionalization:	Assume premises A.	
	Then, if $(A \wedge p)$ entails q,	infer $p \rightarrow q$

A Useful Axiom

Law of the Excluded Middle: $p \vee \neg p$

Quantifier Exchange

Quantifier Exchange A:	$\neg(\forall x(P(x)))$	$\equiv$	$\exists x(\neg P(x))$
Quantifier Exchange B:	$\forall x(\neg P(x))$	$\equiv$	$\neg(\exists x(P(x))$

Instantiation and Generalization

Universal Instantiation:	From $\forall x\ (P(x))$,	infer $P(c/x)$
Universal Generalization:	From $P(c/x)$,	infer $\forall x\ (P(x))$
Existential Instantiation:	From $\exists x\ (P(x))$,	infer $P(c^*/x)$
Existential Generalization:	From $P(c/x)$,	infer $\exists x\ (P(x))$