Logic: The Big Picture

A typical logic is described in terms of

- *syntax*: what are the legitimate formulas
- *semantics*: under what circumstances is a formula true
- *proof theory/ axiomatization*: rules for proving a formula true

Truth and provability are quite different.

- What is provable depends on the axioms and inference rules you use
- Provability is a mechanical, turn-the-crank process
- What is true depends on the semantics

"Hilbert-style" proof systems

Prof. George talked about what are called "natural deduction systems". Here is a slightly different (but related!) approach to proof systems.

An *axiom system* consists of

- *axioms* (special formulas)
- *rules of inference*: ways of getting new formulas from other formulas. These have the form

  $$A_1, \ldots, A_n \vdash B$$

  Read this as "from $A_1, \ldots, A_n$, infer B."

Think of the axioms as tautologies, while the rules of inference give you a way to derive new tautologies from old ones.

Derivations

A *derivation* (or *proof*) in an axiom system AX is a sequence of formulas

$$C_1, \dots, C_N;$$

each formula C_k is either an axiom in AX or follows from previous formulas using an inference rule in AX:

- i.e., there is an inference rule $A_1, \dots, A_n \vdash B$ such that $A_i = C_{j_i}$ for some $j_i < N$ and $B = C_N$.

This is said to be a *derivation* or *proof* of C_N.

A derivation is a syntactic object: it's just a sequence of formulas that satisfy certain constraints.

- Whether a formula is derivable depends on the axiom system
- Different axioms $\rightarrow$ different formulas derivable
- Derivation has nothing to do with truth!
  - How can we connect derivability and truth?
    - In propositional logic, what is true depends on the truth assignment
    - In first-order logic, truth depends on the *interpretation*

Typical axioms of propositional logic:

- $P \Rightarrow \neg\neg P$
- $P \Rightarrow (Q \Rightarrow P)$

What makes an axiom "acceptable"?

- it's a tautology

Typical axioms of propositional logic:

- $P \Rightarrow \neg\neg P$
- $P \Rightarrow (Q \Rightarrow P)$

What makes an axiom "acceptable"?

- it's a tautology

Typical rule of inference is *modus ponens*

$$A \Rightarrow B, A \vdash B$$

What makes an inference rule "acceptable"?

- it preserves validity
- if the formulas on the left-hand side of $\vdash$ are tautologies, then so is the formula on the right-hand side of $\vdash$

Sound and Complete Axiomatizations

Standard question in logic:

> *Can we come up with a nice* sound and complete axiomatization*: a (small, natural) collection of axioms and inference rules from which it is possible to derive all and only the tautologies?*

- *Soundness* says that only tautologies are derivable
- *Completeness* says you can derive all tautologies

If all the axioms are valid and all rules of inference preserve validity, then all formulas that are derivable must be valid.

- Proof: by induction on the length of the derivation

It's not so easy to find a complete axiomatization.

A Sound and Complete Axiomatization for Propositional Logic

Consider the following axiom schemes:

A1. $A \Rightarrow (B \Rightarrow A)$

A2. $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$

A3. $((A \Rightarrow B) \Rightarrow (A \Rightarrow \neg B)) \Rightarrow \neg A$

These are axioms schemes; each one encodes an infinite set of axioms:

- $P \Rightarrow (Q \Rightarrow P)$ and $(P \Rightarrow R) \Rightarrow (Q \Rightarrow (P \Rightarrow R))$ are instances of A1.

A Sound and Complete Axiomatization for Propositional Logic

Consider the following axiom schemes:

A1. $A \Rightarrow (B \Rightarrow A)$

A2. $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$

A3. $((A \Rightarrow B) \Rightarrow (A \Rightarrow \neg B)) \Rightarrow \neg A$

These are axioms schemes; each one encodes an infinite set of axioms:

- $P \Rightarrow (Q \Rightarrow P)$ and $(P \Rightarrow R) \Rightarrow (Q \Rightarrow (P \Rightarrow R))$ are instances of A1.

Theorem: A1, A2, A3 + modus ponens give a sound and complete axiomatization for formulas in propositional logic involving only $\Rightarrow$ and $\neg$.

- Recall: can define $\vee$ and $\wedge$ using $\Rightarrow$ and $\neg$
  - $P \vee Q$ is equivalent to $\neg P \Rightarrow Q$
  - $P \wedge Q$ is equivalent to $\neg(P \Rightarrow \neg Q)$

A Sample Proof

Derivation of $P \Rightarrow P$:

1. $P \Rightarrow ((P \Rightarrow P) \Rightarrow P)$
   [instance of A1: take $A = P$, $B = P \Rightarrow P$]
2. $(P \Rightarrow ((P \Rightarrow P) \Rightarrow P)) \Rightarrow ((P \Rightarrow (P \Rightarrow P)) \Rightarrow (P \Rightarrow P))$
   [instance of A2: take $A = C = P$, $B = P \Rightarrow P$]
3. $(P \Rightarrow (P \Rightarrow P)) \Rightarrow (P \Rightarrow P)$
   [applying modus ponens to 1, 2]
4. $P \Rightarrow (P \Rightarrow P)$ [instance of A1: take $A = B = P$]
5. $P \Rightarrow P$ [applying modus ponens to 3, 4]

Try deriving $P \Rightarrow \neg\neg P$ from these axioms

- it's hard!

It's typically easier to check that a formula is a tautology than it is to prove that it's true, using the axioms

- Just try all truth assignments

Once you prove that an axiom system is sound and complete, you know that if φ is a tautology, then there is a derivation of φ from the axioms (even if it's hard to find)

Syntax of First-Order Logic

We have:

- *constant symbols*: *Alice*, *Bob*
- variables: $x, y, z, \ldots$
- *predicate symbols* of each arity: P, Q, R, ...
  - A *unary* predicate symbol takes one argument: $P(Alice)$, $Q(z)$
  - A *binary* predicate symbol takes two arguments: *Loves(Bob,Alice)*, *Taller(Alice,Bob)*.

An *atomic expression* is a predicate symbol together with the appropriate number of arguments.

- Atomic expressions act like primitive propositions in propositional logic
  - we can apply $\wedge$, $\vee$, $\neg$ to them
  - we can also quantify the variables that appear in them

Typical formula:

$$\forall x \exists y (P(x, y) \Rightarrow \exists z Q(x, z))$$

Semantics of First-Order Logic

Assume we have some domain D.

- The domain could be finite:
  - $\{1, 2, 3, 4, 5\}$
  - the people in this room
- The domain could be infinite
  - N, R, ...

A statement like $\forall x P(x)$ means that $P(d)$ is true for each d in the domain.

- If the domain is N, then $\forall x P(x)$ is equivalent to

$$P(0) \wedge P(1) \wedge P(2) \wedge \ldots$$

Similarly, $\exists x P(x)$ means that $P(d)$ is true for some d in the domain.

- If the domain is N, then $\exists x P(x)$ is equivalent to

$$P(0) \vee P(1) \vee P(2) \vee \ldots$$

Is $\exists x(x^2 = 2)$ true?

Yes if the domain is R; no if the domain is N.

How about $\forall x \forall y((x < y) \Rightarrow \exists z(x < z < y))$?

First-Order Logic: Formal Semantics

How do we decide if a first-order formula is true? Need:

- a domain D (what are you quantifying over)
- an *interpretation* I that interprets the constants and predicate symbols:
  - for each constant symbol c, $I(c) \in D$
    - Which domain element is Alice?
  - for each unary predicate P, $I(P)$ is a predicate on domain D
    - formally, $I(P)(d) \in \{\text{true,false}\}$ for each $d \in D$
    - Is Alice Tall? How about Bob?
  - for each binary predicate Q, $I(Q)$ is a predicate on $D \times D$:
    - formally, $I(Q)(d_1, d_2) \in \{\text{true,false}\}$ for each $d_1, d_2 \in D$
    - Is Alice taller than Bob?
- a valuation V associating with each variable x an element $V(x) \in D$.
  - To figure out if $P(x)$ is true, you need to know what x is.

Defining Truth in First-Order Logic

Now we can define whether a formula A is true, given a domain D, an interpretation I, and a valuation V, written $(I, D, V) \models A$.

- Read this from right to left: A is true at $(\models)$ (I, D, V)

Defining Truth in First-Order Logic

Now we can define whether a formula A is true, given a domain D, an interpretation I, and a valuation V, written $(I, D, V) \models A$.

- Read this from right to left: A is true at ($\models$) (I, D, V)

The definition is by induction:

$(I, D, V) \models P(x)$ if $I(P)(V(x)) = \text{true}$

Defining Truth in First-Order Logic

Now we can define whether a formula A is true, given a domain D, an interpretation I, and a valuation V, written $(I, D, V) \models A$.

- Read this from right to left: A is true at $(\models)$ (I, D, V)

The definition is by induction:

$(I, D, V) \models P(x)$ if $I(P)(V(x)) = \text{true}$

$(I, D, V) \models P(c)$ if $I(P)(I(c)) = \text{true}$

Defining Truth in First-Order Logic

Now we can define whether a formula A is true, given a domain D, an interpretation I, and a valuation V, written $(I, D, V) \models A$.

- Read this from right to left: A is true at $(\models)$ (I, D, V)

The definition is by induction:

$(I, D, V) \models P(x)$ if $I(P)(V(x)) = \text{true}$

$(I, D, V) \models P(c)$ if $I(P)(I(c))) = \text{true}$

$(I, D, V) \models \forall x A$ if $(I, D, V') \models A$ for all valuations V' that agree with V except possibly on x

- $V'(y) = V(y)$ for all $y \neq x$
- $V'(x)$ can be arbitrary

$(I, D, V) \models \exists x A$ if $(I, D, V') \models A$ for some valuation V' that agrees with V except possibly on x.

Defining Truth in First-Order Logic

Now we can define whether a formula A is true, given a domain D, an interpretation I, and a valuation V, written $(I, D, V) \models A$.

- Read this from right to left: A is true at $(\models)$ (I, D, V)

The definition is by induction:

$(I, D, V) \models P(x)$ if $I(P)(V(x)) = \text{true}$

$(I, D, V) \models P(c)$ if $I(P)(I(c))) = \text{true}$

$(I, D, V) \models \forall xA$ if $(I, D, V') \models A$ for all valuations V' that agree with V except possibly on x

- $V'(y) = V(y)$ for all $y \neq x$
- $V'(x)$ can be arbitrary

$(I, D, V) \models \exists xA$ if $(I, D, V') \models A$ for some valuation V' that agrees with V except possibly on x.

Translating from English to First-Order Logic

All men are mortal
Socrates is a man
Therefore Socrates is mortal

There is two unary predicates: *Mortal* and *Man*
There is one constant: *Socrates*
The domain is the set of all people

$$\forall x(Man(x) \Rightarrow Mortal(x))$$
$$Man(Socrates)$$

$$Mortal(Socrates)$$

More on Quantifiers

$\forall x \forall y P(x, y)$ is equivalent to $\forall y \forall x P(x, y)$

- P is true for every choice of x and y

Similarly $\exists x \exists y P(x, y)$ is equivalent to $\exists y \exists x P(x, y)$

- P is true for some choice of (x, y).

What about $\forall x \exists y P(x, y)$? Is it equivalent to $\exists y \forall x P(x, y)$?

- Suppose the domain is the natural numbers. Compare:
  - $\forall x \exists y (y \geq x)$
  - $\exists y \forall x (y \geq x)$

In general, $\exists y \forall x P(x, y) \Rightarrow \forall x \exists y P(x, y)$ is *logically valid*.

- A logically valid formula in first-order logic is the analogue of a tautology in propositional logic.
- A formula is logically valid if it's true in every domain and for every *interpretation* of the predicate symbols.

More valid formulas involving quantifiers:

- $\neg\forall x P(x) \Leftrightarrow \exists x \neg P(x)$
- Replacing P by $\neg P$, we get:

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

- Therefore

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

- Similarly, we have

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

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

Axiomatizing First-Order Logic

Just as in propositional logic, there are axioms and rules of inference that provide a sound and complete axiomatization for first-order logic, independent of the domain.

A typical axiom:

- $\forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall x P(x) \Rightarrow \forall x Q(x))$.

A typical rule of inference is *Universal Generalization*:

$$\varphi(x) \vdash \forall x \varphi(x)$$

Gödel provided a sound and complete axioms system for first-order logic in 1930.

Axiomatizing Arithmetic

Suppose we restrict the domain to the natural numbers, and allow only the standard symbols of arithmetic ($+$, $\times$, $=$, $>$, 0, 1).

Typical true formulas include:

- $\forall x \exists y (x \times y = x)$
- $\forall x \exists y (x = y + y \vee x = y + y + 1)$

Let $Prime(x)$ be an abbreviation of

$$\forall y \forall z ((x = y \times z) \Rightarrow ((y = 1) \vee (y = x)))$$

When is $Prime(x)$ true?

Axiomatizing Arithmetic

Suppose we restrict the domain to the natural numbers, and allow only the standard symbols of arithmetic ($+$, $\times$, $=$, $>$, 0, 1).
Typical true formulas include:

- $\forall x \exists y (x \times y = x)$
- $\forall x \exists y (x = y + y \vee x = y + y + 1)$

Let $Prime(x)$ be an abbreviation of

$$\forall y \forall z ((x = y \times z) \Rightarrow ((y = 1) \vee (y = x)))$$

When is $Prime(x)$ true? If x is prime!

Axiomatizing Arithmetic

Suppose we restrict the domain to the natural numbers, and allow only the standard symbols of arithmetic ($+$, $\times$, $=$, $>$, 0, 1).
Typical true formulas include:

- $\forall x \exists y (x \times y = x)$
- $\forall x \exists y (x = y + y \vee x = y + y + 1)$

Let $Prime(x)$ be an abbreviation of

$$\forall y \forall z ((x = y \times z) \Rightarrow ((y = 1) \vee (y = x)))$$

When is $Prime(x)$ true? If x is prime!

What does the following formula say?

- $\forall x (\exists y (y > 1 \wedge x = y + y) \Rightarrow$
  $\exists z_1 \exists z_2 (Prime(z_1) \wedge Prime(z_2) \wedge x = z_1 + z_2))$

Axiomatizing Arithmetic

Suppose we restrict the domain to the natural numbers, and allow only the standard symbols of arithmetic ($+$, $\times$, $=$, $>$, 0, 1).
Typical true formulas include:

- $\forall x \exists y (x \times y = x)$
- $\forall x \exists y (x = y + y \vee x = y + y + 1)$

Let $Prime(x)$ be an abbreviation of

$$\forall y \forall z ((x = y \times z) \Rightarrow ((y = 1) \vee (y = x)))$$

When is $Prime(x)$ true? If x is prime!

What does the following formula say?

- $\forall x (\exists y (y > 1 \wedge x = y + y) \Rightarrow$
  $\exists z_1 \exists z_2 (Prime(z_1) \wedge Prime(z_2) \wedge x = z_1 + z_2))$
- This is *Goldbach's conjecture*: every even number other than 2 is the sum of two primes.
  - Is it true? We don't know.

Gödel's Incompleteness Theorem

Is there a nice (technically: recursive, so that a program can check whether a formula is an axiom) sound and complete axiomatization for arithmetic?

- *Gödel's Incompleteness Theorem:* NO!

This is arguably the most important result in mathematics of the 20th century.

Connections: Random Graphs

Suppose we have a random graph with n vertices. How likely is it to be connected?

- What is a *random* graph?
  - If it has n vertices, there are $C(n, 2)$ possible edges, and $2^{C(n,2)}$ possible graphs. What fraction of them is connected?
  - One way of thinking about this. Build a graph using a random process, that puts each edge in with probability $1/2$.

- Given three vertices a, b, and c, what's the probability that there is an edge between a and b and between b and c? $1/4$
- What is the probability that there is no path of length 2 between a and c? $(3/4)^{n-2}$
- What is the probability that there is a path of length 2 between a and c? $1-(3/4)^{n-2}$
- What is the probability that there is a path of length 2 between a and every other vertex? $> (1-(3/4)^{n-2})^{n-1}$

Now use the binomial theorem to compute $(1-(3/4)^{n-2})^{n-1}$

$$\begin{aligned}&(1-(3/4)^{n-2})^{n-1}\\ =\;& 1-(n-1)(3/4)^{n-2}+C(n-1,2)(3/4)^{2(n-2)}+\cdots\end{aligned}$$

For sufficiently large n, this will be (just about) 1.

Bottom line: If n is large, then it is almost certain that a random graph will be connected. In fact, with probability approaching 1, all nodes are connected by a path of length at most 2.

This is not a fluke!

Suppose we consider first-order logic with one binary predicate R.

- Interpretation: $R(x, y)$ is true in a graph if there is a directed edge from x to y.

What does this formula say:

$$\forall x \forall y (R(x, y) \vee \exists z (R(x, z) \wedge R(z, y))$$

Theorem: [Fagin, 1976] If P is *any* property expressible in first-order logic using a single binary predicate R, it is either true in almost all graphs, or false in almost all graphs.

This is called a *0-1 law*.
This is an example of a deep connection between logic, probability, and graph theory.

- There are lots of others!

Eight Powerful Ideas

- **Counting**: Count without counting (*combinatorics*)

Eight Powerful Ideas

- **Counting**: Count without counting (*combinatorics*)
- **Induction**: Recognize it in all its guises.

Eight Powerful Ideas

- **Counting**: Count without counting (*combinatorics*)
- **Induction**: Recognize it in all its guises.
- **Exemplification**: Find a sense in which you can try out a problem or solution on small examples.

Eight Powerful Ideas

- **Counting**: Count without counting (*combinatorics*)
- **Induction**: Recognize it in all its guises.
- **Exemplification**: Find a sense in which you can try out a problem or solution on small examples.
- **Abstraction**: Abstract away the inessential features of a problem.
  - represent it as a graph
  - describe it in first-order logic

Eight Powerful Ideas

- **Counting**: Count without counting (*combinatorics*)
- **Induction**: Recognize it in all its guises.
- **Exemplification**: Find a sense in which you can try out a problem or solution on small examples.
- **Abstraction**: Abstract away the inessential features of a problem.
  - represent it as a graph
  - describe it in first-order logic
- **Modularity**: Decompose a complex problem into simpler subproblems.

Eight Powerful Ideas

- **Counting**: Count without counting (*combinatorics*)
- **Induction**: Recognize it in all its guises.
- **Exemplification**: Find a sense in which you can try out a problem or solution on small examples.
- **Abstraction**: Abstract away the inessential features of a problem.
  - represent it as a graph
  - describe it in first-order logic
- **Modularity**: Decompose a complex problem into simpler subproblems.
- **Representation**: Understand the relationships between different representations of the same information or idea.
  - Graphs vs. matrices vs. relations

Eight Powerful Ideas

- **Counting**: Count without counting (*combinatorics*)
- **Induction**: Recognize it in all its guises.
- **Exemplification**: Find a sense in which you can try out a problem or solution on small examples.
- **Abstraction**: Abstract away the inessential features of a problem.
  - represent it as a graph
  - describe it in first-order logic
- **Modularity**: Decompose a complex problem into simpler subproblems.
- **Representation**: Understand the relationships between different representations of the same information or idea.
  - Graphs vs. matrices vs. relations
- **Probabilistic inference:** Drawing inferences from data
  - Bayes' rule
- **Probabilistic methods**: Flipping a coin can be surprisingly helpful!
  - probabilistic primality checking

(A Little Bit on) NP

(No details here; just a rough sketch of the ideas. Take CS 4810/4820 if you want more.)

NP = nondeterministic polynomial time

- a language (set of strings) L is in NP if, for each $x \in L$, you can guess a witness y showing that $x \in L$ and quickly (in polynomial time) verify that it's correct.
- Examples:
  - Does a graph have a Hamiltonian path?
    - guess a Hamiltonian path
  - Is a formula satisfiable?
    - guess a satisfying assignment
  - Is there a schedule that satisfies certain constraints?
  - ...

Formally, L is in NP if there exists a language L' such that

1. $x \in L$ iff there exists a y such that $(x, y) \in L'$, and
2. checking if $(x, y) \in L'$ can be done in polynomial time

NP-completeness

- A problem is NP-hard if every NP problem can be *reduced* to it.

A problem is NP-complete if it is in NP and NP-hard

- Intuitively, if it is one of the hardest problems in NP.

There are *lots* of problems known to be NP-complete

- If any NP complete problem is doable in polynomial time, then they all are.
  - Hamiltonian path
  - satisfiability
  - scheduling
  - ...
- If you can prove P $=$ NP, you'll get a Turing award.