Logic: The Big Picture

Logic is a tool for formalizing reasoning. There are lots of different logics:

- probabilistic logic: for reasoning about probability
- temporal logic: for reasoning about time (and programs)
- epistemic logic: for reasoning about knowledge

The simplest logic (on which all the rest are based) is *propositional logic*. It is intended to capture features of arguments such as the following:

> *Borogroves are mimsy whenever it is brillig. It is now brillig and this thing is a borogrove. Hence this thing is mimsy.*

Propositional logic is good for reasoning about

- conjunction, negation, implication ("if ... then ...")

Amazingly enough, it is also useful for

- circuit design
- program verification

Propositional Logic: Syntax

To formalize the reasoning process, we need to restrict the kinds of things we can say. Propositional logic is particularly restrictive.

The *syntax* of propositional logic tells us what are legitimate formulas. We've seen this already:

We start with *primitive propositions*, basic statements like

- It is now brillig
- This thing is mimsy
- It's raining in San Francisco
- n is even

We can then form more complicated *compound propositions* using connectives like:

- $\neg$: not
- $\wedge$: and
- $\vee$: or
- $\Rightarrow$: implies

MCS uses English (NOT, AND, OR, IMPLIES). I'll stick to the standard mathematical notation.

Examples:

- $\neg P$: it is not the case that P
- $P \wedge Q$: P and Q
- $P \vee Q$: P or Q
- $P \Rightarrow Q$: P implies Q (if P then Q)

Typical formula:

$$P \wedge (\neg P \Rightarrow (Q \Rightarrow (R \vee P)))$$

Wffs

Formally, we define *well-formed formulas* (*wffs* or just *formulas*) inductively:

1. Every primitive proposition $P, Q, R, \ldots$ is a wff
2. If A is a wff, so is $\neg A$
3. If A and B are wffs, so are $(A \wedge B)$, $(A \vee B)$, and $(A \Rightarrow B)$
    - note that I added parentheses for disambiguation, just as in regular expressions
    - it's worth stressing: formulas are syntactic objects, just like regular expresesions

Wffs

Formally, we define *well-formed formulas* (*wffs* or just *formulas*) inductively:

1. Every primitive proposition $P, Q, R, \ldots$ is a wff
2. If A is a wff, so is $\neg A$
3. If A and B are wffs, so are $(A \wedge B)$, $(A \vee B)$, and $(A \Rightarrow B)$
   - note that I added parentheses for disambiguation, just as in regular expressions
   - it's worth stressing: formulas are syntactic objects, just like regular expresesions

More precisely:

- $\Phi_0 = \{\mathit{primitive\ propositions}\}$
- $\Phi_{n+1} = \Phi_n \cup \{\neg A, (A \wedge B), (A \vee B),)A \Rightarrow B) : A, B \in \Phi_n\}$

$\Phi^* = \cup_{n=0}^{\infty} \Phi_n$

Φ^* is the smallest set that contains Φ_0 and is closed under $\neg$, $\wedge$, $\vee$, and $\Rightarrow$.

Semantics

Given a formula, we want to decide if it is true or false.

We've seen this for propositional logic: use truth tables.

Tautologies

- Recall: A formula φ is *valid* (also known as a *tautology*) if every truth assignment makes φ true.
- φ is *satisfiable* if some truth assignment makes φ true.

Tautologies

- Recall: A formula φ is *valid* (also known as a *tautology*) if every truth assignment makes φ true.
- φ is *satisfiable* if some truth assignment makes φ true.

- How hard is it to check if a formula is true under a given truth assignment?
- Easy: just plug it in and evaluate.
  - Time linear in the length of the formula

Tautologies

- Recall: A formula φ is *valid* (also known as a *tautology*) if every truth assignment makes φ true.
- φ is *satisfiable* if some truth assignment makes φ true.

- How hard is it to check if a formula is true under a given truth assignment?
- Easy: just plug it in and evaluate.
  - Time linear in the length of the formula

- How hard is it to check if a formula is satisfiable/a tautology?
  - How many truth assignments are there for a formula with n primitive propositions?

Tautologies

- Recall: A formula φ is *valid* (also known as a *tautology*) if every truth assignment makes φ true.
- φ is *satisfiable* if some truth assignment makes φ true.

- How hard is it to check if a formula is true under a given truth assignment?
- Easy: just plug it in and evaluate.
  - Time linear in the length of the formula

- How hard is it to check if a formula is satisfiable/a tautology?
  - How many truth assignments are there for a formula with n primitive propositions?

Can we do better than checking every truth assignment?

- In the worst case, it appears not.
  - The problem is co-NP-complete.
  - The *satisfiability* problem—deciding if at least one truth assignment makes the formula true—is NP-complete.

Nevertheless, it often seems that the reasoning is straightforward: Why is this true:

$$((P \Rightarrow Q) \wedge (Q \Rightarrow R)) \Rightarrow (P \Rightarrow R)$$

We want to show that if $P \Rightarrow Q$ and $Q \Rightarrow R$ is true, then $P \Rightarrow R$ is true.

So assume that $P \Rightarrow Q$ and $Q \Rightarrow R$ are both true. To show that $P \Rightarrow R$, assume that P is true. Since $P \Rightarrow Q$ is true, Q must be true. Since $Q \Rightarrow R$ is true, R must be true. Hence, $P \Rightarrow R$ is true.

We want to codify such reasoning.

Formal Deductive Systems

A *formal deductive system* (also known as an *axiom system*) consists of

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

  $$\frac{\begin{array}{l} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}}{B}$$

  Read this as "from $A_1, \ldots, A_n$, infer B."
  - Sometimes written "$A_1, \ldots, A_n \vdash B$"

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?

Typical Axioms

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

What makes an axiom "acceptable"?

- it's a tautology

Typical Rules of Inference

Modus Ponens

$$\frac{\begin{array}{l} A \Rightarrow B \\ A \end{array}}{B}$$

Modus Tollens

$$\frac{\begin{array}{l} A \Rightarrow B \\ \neg B \end{array}}{\neg A}$$

What makes a rule of inference "acceptable"?

- It preserves validity:
  - if the antecedents are valid, so is the conclusion
- Both modus ponens and modus tollens are acceptable

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

Put another way, if AX is an axiom for propositional logic:

- AX is sound if
  $\{\text{valid formulas}\} \supseteq \{\text{formulas provable from AX}\}$
- AX is complete if
  $\{\text{valid formulas}\} \subseteq \{\text{formulas provable from AX}\}$

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

Put another way, if AX is an axiom for propositional logic:

- AX is sound if
  $\{\text{valid formulas}\} \supseteq \{\text{formulas provable from AX}\}$
- AX is complete if
  $\{\text{valid formulas}\} \subseteq \{\text{formulas provable from AX}\}$

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:

- $P \Rightarrow (Q \Rightarrow P)$, $(P \Rightarrow R) \Rightarrow (Q \Rightarrow (P \Rightarrow R))$ 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. $(P \Rightarrow ((P \Rightarrow P) \Rightarrow P)) \Rightarrow ((P \Rightarrow (P \Rightarrow P)) \Rightarrow \quad (P \Rightarrow P))$
   [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!

Algorithm Verification

This is (yet another) hot area of computer science.

- How do you prove that your program is correct?
  - You could test it on a bunch of instances. That runs the risk of not exercising all the features of the program.

In general, this is an intractable problem.

- For small program fragments, formal verification using logic is useful
- It also leads to insights into program design.

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?

(a) Yes
(b) No
(c) It depends

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

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

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

(a) Yes
(b) No
(c) It depends

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

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

We'll skip the formal semantics of first-order logic here.

- If you want to know more, take a logic course!

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

$$\frac{\forall x(Man(x) \Rightarrow Mortal(x)) \quad Man(Socrates)}{Mortal(Socrates)}$$

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

(a) Yes
(b) $\exists y \forall x P(x, y)$ implies $\forall x \exists y P(x, y)$, but the converse isn't true
(c) $\forall x \exists y P(x, y)$ implies $\exists y \forall x P(x, y)$, but the converse isn't true
(d) ???

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

(a) Yes
(b) $\exists y \forall x P(x, y)$ implies $\forall x \exists y P(x, y)$, but the converse isn't true
(c) $\forall x \exists y P(x, y)$ implies $\exists y \forall x P(x, y)$, but the converse isn't true
(d) ???

Suppose the domain is the natural numbers. Compare:

- $\forall x \exists y (y \geq x)$
- $\exists y \forall x (y \geq x)$

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

(a) Yes
(b) $\exists y \forall x P(x, y)$ implies $\forall x \exists y P(x, y)$, but the converse isn't true
(c) $\forall x \exists y P(x, y)$ implies $\exists y \forall x P(x, y)$, but the converse isn't true
(d) ???

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.

Is Everything Provable?

If we ask you to prove something from homework which happens true, is it necessarily provable?

Is Everything Provable?

If we ask you to prove something from homework which happens true, is it necessarily provable?

- Of course, if we ask you to prove it, then it should be provable.
- But what about if a computer scientist is trying to prove a theorem that she is almost certain is true.
  - Can she be confident that it has a proof?
  - Can something be true without being provable?

Is Everything Provable?

If we ask you to prove something from homework which happens true, is it necessarily provable?

- Of course, if we ask you to prove it, then it should be provable.
- But what about if a computer scientist is trying to prove a theorem that she is almost certain is true.
  - Can she be confident that it has a proof?
  - Can something be true without being provable?

Remember, whether something is provable depends on the rules of the game:

- the axioms and inference rules

Obviously, you can't prove much if you don't have a good selection of axioms and inference rules to work with.

- In a remarkable result, Gödel proved that, *no matter what axiom system AX you used*, there were statements that were true about arithmetic that could not be proved in AX.

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

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

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

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

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

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

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

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

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. But it is either true or false.
    - But is it provable?

Gödel's Incompleteness Theorem

Is there an axiom system from which you can prove all and only true statements about arithmetic?

- that is, you want the axiom system to be *sound*
    - The axioms must be valid arithmetic facts, and the rules of inference must preserve validity
    - otherwise you could prove statements that are false

  and *complete*
    - This means that you can prove *all* true statements

This is easy!

- Just take the axioms to consist of all true statements.

Gödel's Incompleteness Theorem

Is there an axiom system from which you can prove all and only true statements about arithmetic?

- that is, you want the axiom system to be *sound*
  - The axioms must be valid arithmetic facts, and the rules of inference must preserve validity
  - otherwise you could prove statements that are false

  and *complete*
  - This means that you can prove *all* true statements

This is easy!

- Just take the axioms to consist of all true statements.

That's cheating! To make this interesting, we need a restriction:

- The set of axioms must be "nice"
  - technically: *recursive*, so that a program can check whether a formula is an axiom

Gödel's Incompleteness Theorem

Is there an axiom system from which you can prove all and only true statements about arithmetic?

- that is, you want the axiom system to be *sound*
  - The axioms must be valid arithmetic facts, and the rules of inference must preserve validity
  - otherwise you could prove statements that are false

  and *complete*
  - This means that you can prove *all* true statements

This is easy!

- Just take the axioms to consist of all true statements.

That's cheating! To make this interesting, we need a restriction:

- The set of axioms must be "nice"
  - technically: *recursive*, so that a program can check whether a formula is an axiom

Gödel's Incompleteness Theorem: There is no sound and complete recursive axiomatization of arithmetic.

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

Key idea of Gödel's proof: Given an axiomatization Ax, we can write a formula S_{Ax} that says "I am true iff I am not provable in Ax."

- Suppose that S_{Ax} is not provable in Ax. We can add S_{Ax} as an axiom to Ax. This gives another axiomatization Ax′. We can find another sentence $S_{Ax'}$ that is true iff it is not provable in Ax′.

Defining Ax involves "arithmetizing" formulas:

- Associating with each formula F a number $[F]$ that encodes the formula F.
- We can also find numbers that encode proofs (which are just sequences of formulas)
  - This uses ideas of number theory!
- S_{A_x} is a formula with one free variable x (just like $Prime(x)$) that is true of x iff the formula represented by the number x is not provable. We then consider $S_{A_x}([S_{A_x}])$.

Key idea of Gödel's proof: Given an axiomatization Ax, we can write a formula S_{Ax} that says "I am true iff I am not provable in Ax."

- Suppose that S_{Ax} is not provable in Ax. We can add S_{Ax} as an axiom to Ax. This gives another axiomatization Ax′. We can find another sentence $S_{Ax'}$ that is true iff it is not provable in Ax′.

Defining Ax involves "arithmetizing" formulas:

- Associating with each formula F a number $[F]$ that encodes the formula F.
- We can also find numbers that encode proofs (which are just sequences of formulas)
  - This uses ideas of number theory!
- S_{A_x} is a formula with one free variable x (just like $Prime(x)$) that is true of x iff the formula represented by the number x is not provable. We then consider $S_{A_x}([S_{A_x}])$.

But wait, there's more ...

The first-order theory of the reals

Instead of interpreting the first-order theory of arithmetic over the natural numbers, we can interpret it over the reals.

- Some formulas hold for both interpretations: $\forall x \forall y (x + y = y + x)$
- Some formulas are true under one interpretation and not the other:
  - $\exists x (x^2 = 2)$
  - $\exists x \exists y (x < y \land \neg \exists z (x < z < y))$

You would think that axiomatizing the real numbers is even harder than axiomatizing the natural numbers, but ...

The first-order theory of the reals

Instead of interpreting the first-order theory of arithmetic over the natural numbers, we can interpret it over the reals.

- Some formulas hold for both interpretations:
  $\forall x \forall y (x + y = y + x)$
- Some formulas are true under one interpretation and not the other:
  - $\exists x (x^2 = 2)$
  - $\exists x \exists y (x < y \wedge \neg \exists z (x < z < y))$

You would think that axiomatizing the real numbers is even harder than axiomatizing the natural numbers, but ...

Theorem: [Tarski] There is an elegant axiomatization of the reals.

The first-order theory of the reals

Instead of interpreting the first-order theory of arithmetic over the natural numbers, we can interpret it over the reals.

- Some formulas hold for both interpretations:
  $\forall x \forall y (x + y = y + x)$
- Some formulas are true under one interpretation and not the other:
  - $\exists x (x^2 = 2)$
  - $\exists x \exists y (x < y \wedge \neg \exists z (x < z < y))$

You would think that axiomatizing the real numbers is even harder than axiomatizing the natural numbers, but ...

Theorem: [Tarski] There is an elegant axiomatization of the reals.

Roughly speaking the axioms say:

- The reals are a field under $+$ and $\times$
- Every odd-degree polynomial has a root

[Canny:] We can decide whether a formula is true or false of the real numbers is exponential time

- Dexter Kozen was a co-author of an earlier paper showing that it was in exponential space

But wait. There's even more . . .

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:

$$\forall x \forall y (R(x, y) \lor \exists z (R(x, z) \land R(z, y))$$

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 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*.

Amazing fact:

- Checking if a formula in the language of graphs is valid (true for every single graphs) is undecidable
  - There is no algorithm that can do it for all formulas
- Checking if a formula is true for *almost* all graphs (i.e., holds with probability 1) can be done in polynomial space.

This is an example of a deep connection between logic, probability, complexity theory, and graph theory.

- There are lots of others!