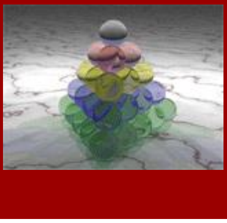
15-251: Great Theoretical Ideas in Computer Science
Lecture 3

Formalization of Proof



What is “proof”?

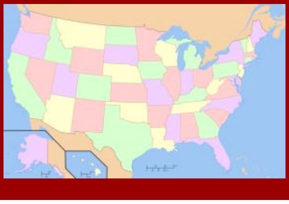
GORM: Good Old Regular Mathematics

- GORM is the math you’ve been doing all your life
- GORM is what we use in the lectures and homeworks
- GORM proofs are written in English (or another human language)
- In GORM, math statements are either true or false
  We try to prove the true ones, disprove the false ones
- GORM proofs are valid if they are:
  rigorous, logical, convincing, complete, precise
- This depends on the audience & assumed background!
- Ultimately, GORM proofs are valid if they are accepted by the community of mathematicians
- That’s OK! But we may also want to try to formalize (within GORM) what it means to be a valid proof.

Four Color Theorem

1852 conjecture:

Any 2-d map of regions can be colored with 4 colors so that no adjacent regions get the same color.



Four Color Theorem

1879: Proved by Kempe in *Amer. J. of Math*

1880: Alternate proof by Tait in
Trans. Roy. Soc. Edinburgh

1890: Heawood finds a bug in Kempe’s proof.

1891: Petersen finds a bug in Tait’s proof.

Kempe’s “proof” was widely acclaimed.

Four Color Theorem

1969: Heesch showed that the theorem could in principle be reduced to checking a large number of cases.

1976:

Appel and Haken wrote a massive amount of code to compute and then check 1936 cases
(1200 hours of computer time).

Claimed this constituted a proof.

Four Color Theorem

Much controversy at the time. Is this a proof??

Arguments against:

No human could ever hand-check the cases.
Perhaps there's a bug in the code.
Perhaps there's a bug in the compiler.
Perhaps there's a bug in the hardware.
No "insight" is derived –
still don't know "why" the theorem is true.

Nevertheless, these days, pretty much everyone credits Appel & Haken with the proof.

Four Color Theorem

A&H found, classified various errors in their work.
Type 1: few minute fix.
Type 3: few day fix.

Ulrich Schmidt 1980 master's thesis:
found thirteen Type 1 errors, one Type 3.

A&H fixed them all (?) in a 1989 book.

1997: "Simpler" computer proof by
Robertson, Sanders, Seymour, Thomas.

Classification of finite simple groups

(the "prime numbers" of group theory)

Theorem: Every finite simple group is either…

Progress started in late 19th century.

100's of papers, 10,000–20,000 pages later…

1983: Gorenstein announces proof is complete.

However, experts knew one piece still missing.

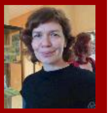
2004: Aschbacher & Smith finish a 1221-page paper, Aschbacher announces proof is complete.

Classification of finite simple groups

Some controversy: Is the theorem proven?

Genuine concern:

Everyone who understands the proof
will die before it's properly collated.



Inna Capdeboscq:
(The youngest person who knows the proof?)

A ~5000 page, 13-volume series of
books describing the proof is underway.

More anecdotes

1993: Wiles announces proof of Fermat's Last Theorem.
Then a bug is found.

1994: Bug fixed, 100-page paper.

1994: Gaoyong Zhang, *Annals of Mathematics*:
proves "n=4 case of Busemann-Petty".

1999: Gaoyong Zhang, *Annals of Mathematics*:
disproves "n=4 case of Busemann-Petty".

Because of such incidents,
many mathematicians became
interested in **formalizing** GORM proofs.

Formal proofs — prehistory

Euclid's *Elements*
(ca. 300 BCE)

Canonized the idea of giving a rigorous, axiomatic deduction for all theorems.

But wasn't **truly** rigorous.

Formal proofs — 19th century

True rigor developed.

Culminated in the understanding that GORM proofs ***can*** be formalized, using tools like First Order Logic, & Deductive Systems.

Bertrand Russell

Alfred Whitehead

Principia Mathematica, ca. 1912

Starting with **axioms of set theory**, developed number theory and some real analysis, in **purely formal logic.**

page 379: "1+1=2"

It became generally agreed that you ***could*** rigorously formalize GORM proofs.

(But nobody wants to!)
(by hand, at least…)

Let's now study the main tool:

First Order Logic

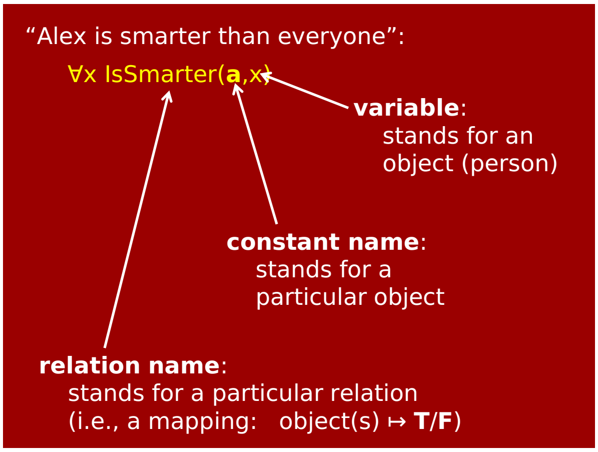
First Order Logic (FOL)

First Order Logic =
Propositional Logic (¬, ∧, ∨, →, ↔)
like from last lecture…

Plus: For All (∀), There Exists (∃), Equals (=)
"constants", "relations", "functions"

Variables like x now represent **objects**, not truth-values.

"Alex is smarter than everyone":

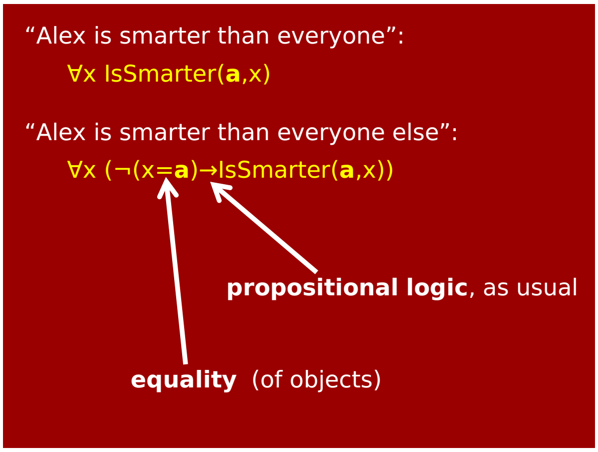


∀x IsSmarter(**a**,x)

variable:
stands for an object (person)

constant name:
stands for a particular object

relation name:
stands for a particular relation
(i.e., a mapping: object(s) ↦ **T/F**)



"Alex is smarter than everyone":

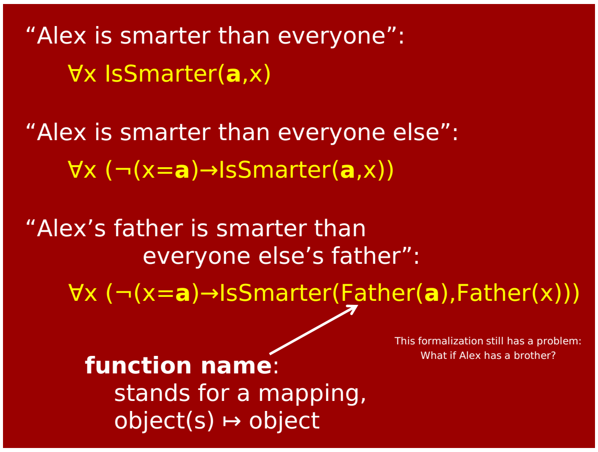
$\forall x\ \mathrm{IsSmarter}(\mathbf{a},x)$

"Alex is smarter than everyone else":

$\forall x\ (\neg(x=\mathbf{a})\rightarrow \mathrm{IsSmarter}(\mathbf{a},x))$

propositional logic, as usual

equality (of objects)



"Alex is smarter than everyone":

$\forall x\ \mathrm{IsSmarter}(\mathbf{a},x)$

"Alex is smarter than everyone else":

$\forall x\ (\neg(x=\mathbf{a})\rightarrow \mathrm{IsSmarter}(\mathbf{a},x))$

"Alex's father is smarter than
everyone else's father":

$\forall x\ (\neg(x=\mathbf{a})\rightarrow \mathrm{IsSmarter}(\mathrm{Father}(\mathbf{a}),\mathrm{Father}(x)))$

function name:
stands for a mapping,
object(s) ↦ object

This formalization still has a problem:
What if Alex has a brother?

Vocabulary: A collection of constant-names,
function-names,
relation-names.

Vocabulary from the previous slide:

one constant-name:	**a**
one function-name:	Father(·)
one relation-name:	IsSmarter(·, ·)

Vocabulary: A collection of constant-names,
function-names,
relation-names.

Another example of a vocabulary:

one constant-name:	**a**
two function-names:	Next(·), Combine(·, ·)
one relation-name:	IsPrior(·, ·)

Example (well-formed) "sentences":

$$\exists x\ (\mathrm{Next}(x)=\mathbf{a})$$

$$\forall x\ \forall y\ (\mathrm{IsPrior}(x,\mathrm{Combine}(\mathbf{a},y)) \rightarrow (\mathrm{Next}(x)=y))$$

$$(\forall x\ \mathrm{IsPrior}(x,\mathrm{Next}(x))) \rightarrow (\mathrm{Next}(\mathbf{a})=\mathrm{Next}(\mathbf{a}))$$

Let's talk about **TRUTH**.

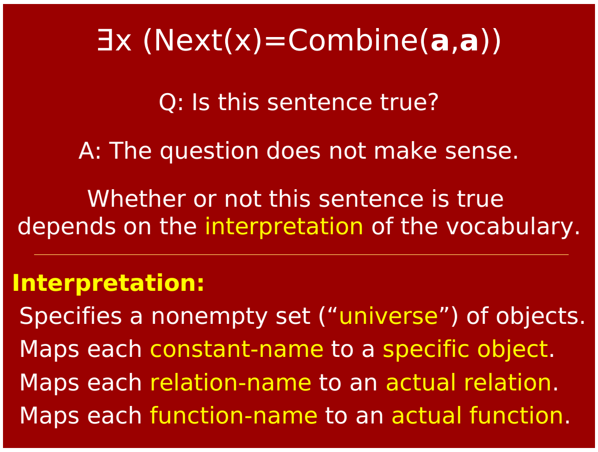
$$\exists x\ (\mathrm{Next}(x)=\mathrm{Combine}(\mathbf{a},\mathbf{a}))$$

Q: Is this sentence true?

A: The question does not make sense.

Whether or not this sentence is true
depends on the interpretation of the vocabulary.

Interpretation:
Informally, says what objects are
and what the vocabulary items stand for.



∃x (Next(x)=Combine(**a**,**a**))

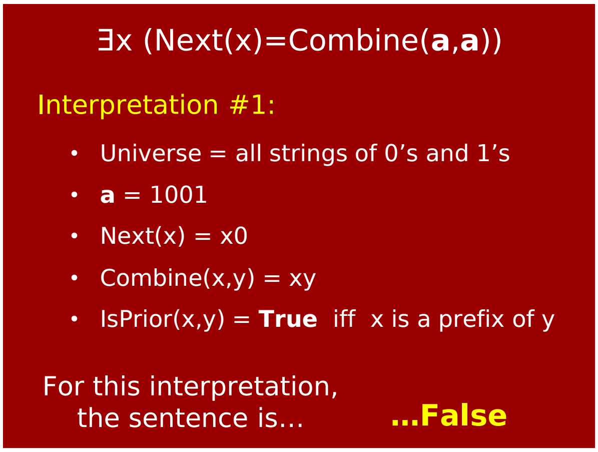
Q: Is this sentence true?

A: The question does not make sense.

Whether or not this sentence is true
depends on the interpretation of the vocabulary.

Interpretation:

Specifies a nonempty set (“universe”) of objects.
Maps each constant-name to a specific object.
Maps each relation-name to an actual relation.
Maps each function-name to an actual function.

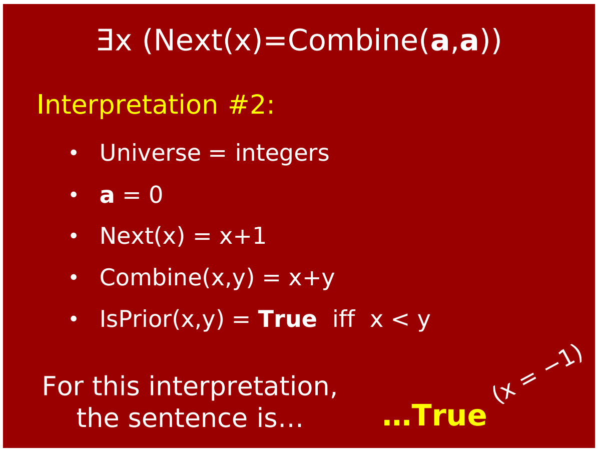


∃x (Next(x)=Combine(**a**,**a**))

Interpretation #1:

- Universe = all strings of 0’s and 1’s
- **a** = 1001
- Next(x) = x0
- Combine(x,y) = xy
- IsPrior(x,y) = **True** iff x is a prefix of y

For this interpretation,
the sentence is… **…False**

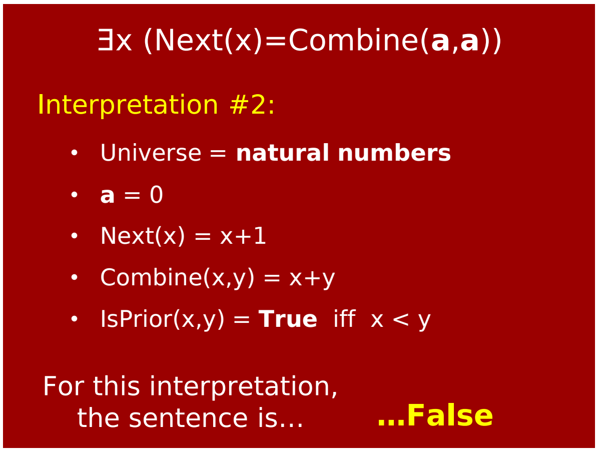


∃x (Next(x)=Combine(**a**,**a**))

Interpretation #2:

- Universe = integers
- **a** = 0
- Next(x) = x+1
- Combine(x,y) = x+y
- IsPrior(x,y) = **True** iff x < y

For this interpretation,
the sentence is… **…True** (x = −1)

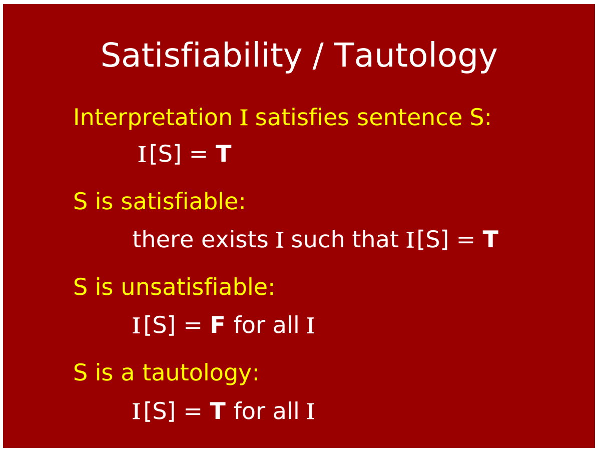


∃x (Next(x)=Combine(**a**,**a**))

Interpretation #2:

- Universe = **natural numbers**
- **a** = 0
- Next(x) = x+1
- Combine(x,y) = x+y
- IsPrior(x,y) = **True** iff x < y

For this interpretation,
the sentence is… **…False**



Satisfiability / Tautology

Interpretation I satisfies sentence S:

I[S] = **T**

S is satisfiable:

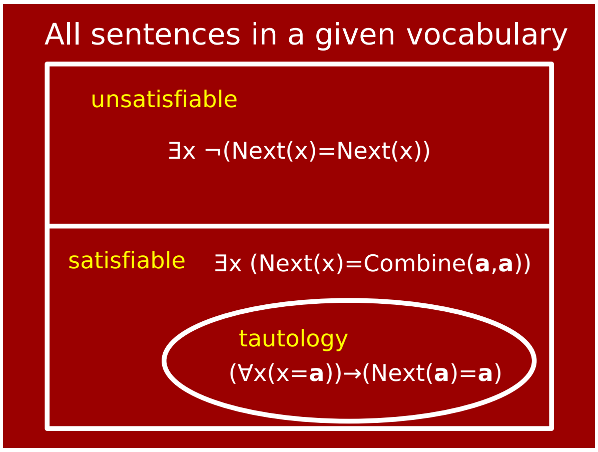
there exists I such that I[S] = **T**

S is unsatisfiable:

I[S] = **F** for all I

S is a tautology:

I[S] = **T** for all I



All sentences in a given vocabulary

unsatisfiable

∃x ¬(Next(x)=Next(x))

satisfiable ∃x (Next(x)=Combine(**a**,**a**))

tautology

(∀x(x=**a**))→(Next(**a**)=**a**)

Tautology: automatically true,
for ‘purely logical’ reasons

Unsatisfiable: automatically false,
for purely logical reasons

Satisfiable (but not a tautology):

truth value depends
on the interpretation
of the vocabulary

$(\exists y\ \forall x\ (x=\text{Next}(y))) \rightarrow (\forall w\ \forall z\ (w=z))$

Problem 1: Show this is satisfiable.

Let’s pick this interpretation:

Universe = integers, Next(y) = y+1.

Now $(\exists y\ \forall x\ (x=\text{Next}(y)))$ means

“there’s an integer y such
that every integer = y+1”.

That’s **False**!
So the whole sentence becomes **True**.
Hence the sentence **is satisfiable.**

$(\exists y\ \forall x\ (x=\text{Next}(y))) \rightarrow (\forall w\ \forall z\ (w=z))$

Problem 2: Is it a tautology?

There is no “truth table method”. ☹

You can’t enumerate all possible interpretations!

It **seems like** you have to use some cleverness…

$(\exists y\ \forall x\ (x=\text{Next}(y))) \rightarrow (\forall w\ \forall z\ (w=z))$

Problem 2: Is it a tautology?

Solution: Yes, it is a tautology!

Proof: Let I be any interpretation.

If $\mathrm{I}[\exists y\ \forall x\ (x=\text{Next}(y))] = \mathbf{F}$,
then the sentence is **True**.

If $\mathrm{I}[\exists y\ \forall x\ (x=\text{Next}(y))] = \mathbf{T}$,
then every object equals Next(y).

In that case, $\mathrm{I}[\forall w\ \forall z\ (w=z)] = \mathbf{T}$.

So no matter what, $\mathrm{I}[\text{the sentence}] = \mathbf{T}$.

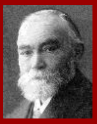
$(\exists y\ \forall x\ (x=\text{Next}(y))) \rightarrow (\forall w\ \forall z\ (w=z))$

Problem 2: Is it a tautology?

Hmm… It’s really a shame
that there’s no truth table method.

Is there **any** “mechanical method”??

Checking tautologies



&

Gottlob Frege David Hilbert

invented the idea of
FOL Deductive Calculus
(usually called a “Hilbert System”)

This means a **Deductive System** for generating
tautologies in First Order Logic.

Checking tautologies

Open almost any textbook on logic.
Chapter 1 will describe some kind of
FOL Deductive Calculus, like:

Initial tautologies / tautology families:

1. stuff like $A \vee \neg A$ for any sentence A
2. stuff like $\forall x\ \forall y\ ((x=\mathbf{a} \wedge y=\mathbf{b}) \rightarrow (\mathrm{Func}(x,y)=\mathrm{Func}(\mathbf{a},\mathbf{b})))$
3. stuff like $\mathrm{IsR}(\mathbf{a}) \rightarrow (\exists x\ \mathrm{IsR}(x))$
4. blah blah blah, a bunch more obviously tautological kinds of sentences

Deduction rule: From A and A→B can deduce B

Easy claim: Anything deducible is a tautology.
(This is "by design".)

Question: is every tautology deducible?

Kurt Gödel

His 1929 PhD thesis: Yes!

"Gödel's COMPLETENESS Theorem"

Checking tautologies

Consequence:

There **is** a purely mechanical algorithm to verify that a given tautology S really is a tautology.

Brute-force search for the shortest deduction in FOL Deductive Calculus!

Logical entailment

"Is S a tautology of First Order Logic?"

moderately interesting

"Assuming sentences $A_1, \ldots, A_m$ ('axioms')
is S a logical consequence ('theorem')?"

more typical kind of
thing to be interested in

Logical entailment

Definition:

Formulas $A_1, \ldots, A_m$ **entail** formula S,

written $A_1, \ldots, A_m \vDash S$,

if every interpretation $\mathcal{I}$ which makes

$A_1, \ldots, A_m$ equal **T** also makes S equal **T**.

Equivalently, $(A_1 \wedge \cdots \wedge A_m) \rightarrow S$ is a tautology.

Formalizing GORM proofs

1. Think of some universe you want to reason about.
2. Invent an appropriate vocabulary
   (constant, function, relation names).
3. Start with some axioms which are true under the interpretation you have in mind.
4. See what theorems these axioms **entail**.

(By Gödel's theorem, equivalent to what you can
deduce from the axioms with FOL Deductive Calculus.)

Example 1: Euclidean geometry

Euclid's Axioms:

1. To draw a straight line from any point to any point.
2. To produce a finite straight line continuously in a straight line.
3. To describe a circle with any center and radius.
4. That all right angles are equal to one another.
5. If a straight line falling on two straight lines make the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which are the angles less than the two right angles.

Euclid

WTF?

Example 1: Euclidean geometry

Alfred Tarski

He did it right.

In his interpretation, universe is the points of $\mathbf{R}^2$.

Example 1: Euclidean geometry

constant-names, function-names: none

relation-names: IsBetween(x,y,z)
IsSameLength(x_1,x_2,y_1,y_2)

axioms:

$\forall x_1 \forall x_2$ IsSameLength(x_1,x_2,x_2,x_1)
$\forall x \forall y \forall z$ IsSameLength(x,y,z,z)→(x=y)
$\forall x \forall y$ IsBetween(x,y,x)→(y=x)
"Segment Extension": $\forall x_1,x_2,y_1,y_2$
$\exists z$ IsBetween(x_1,x_2,z)$\wedge$IsSameLength(x_2,z,y_1,y_2)
… 7 more …

Example 1: Euclidean geometry

Cool fact proved by Tarski (using GORM):

These 11 axioms are **"complete"** for Euclidean geometry.

For every true statement S in Euclidean geometry,

$\{A_1, \ldots, A_{11}\} \vDash S$.

Example 2: Arithmetic of ℕ

constant-name: **0**

function-names: Successor(x)
Plus(x,y)
Times(x,y)

Giuseppe Peano

axioms:

$\forall x$ ¬(Successor(x)=**0**)
$\forall x \forall y$ (Successor(x)=Successor(y))→(x=y)
$\forall x$ Plus(x,**0**)=x
$\forall x \forall y$ Plus(x,Successor(y))=Successor(Plus(x,y))
$\forall x$ Times(x,**0**)=**0**
$\forall x \forall y$ Times(x,Successor(y))=Plus(Times(x,y),x)
"Induction:" For any parameterized formula F(x),
(F(**0**)$\wedge$($\forall x$ F(x)→F(Successor(x)))) → $\forall x$ F(x)

Example 2: Arithmetic of ℕ

Question:
How 'complete' are those 7 axioms?

Answer based on 125 years of experience:
Pretty darn complete.
'Almost all' true statements about arithmetic can be deduced from them.

Four-Square Theorem, Weak Goldbach Conjecture, Prime Number Theorem, Fermat's Last Theorem…

*There are a **few** intentionally-designed, not-too-crazy arithmetical theorems not entailed by Peano's axioms.*

Getting ambitious: **All** of GORM??

In early 20th c., mathematicians sought a simple subject that could capture all GORM topics.

They came up with **Set Theory**.

It's extremely hacky and kludgy, but you **can** seemingly express all GORM concepts with sets.

Getting ambitious: **All** of GORM??

Gross details (don't study these!)…

Define ordered pairs $(x,y) = \{\{x\},\{x,y\}\}$, & tuples, relations, functions…
Define $0 = \varnothing$, $1 = \{\varnothing\}$, $2 = \{\varnothing, \{\varnothing\}\}$, $3 = \{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}$, …
Define induction based on the natural numbers.
Define addition and multiplication by induction.
Define integers from the natural numbers.
Define rational numbers in terms of pairs of integers.
Define real numbers in terms of sequences of rationals.
Now you can start defining calculus concepts, geometry concepts, algebra concepts, …

It's like programming in COBOL or FORTRAN.
(**Type Theory** is the superior modern approach.)

Example 3: Set theory

constant-names, function-names: none

relation-name:

IsElementOf(x,y)
[“$x \in y$”]

Ernst Zermelo++

axioms, catchily known as “**ZFC**”:

$\forall x\, \forall y\, (\, (\forall z\;\; z \in x \leftrightarrow z \in y) \;\rightarrow\; x = y\,)$

$\forall x\, \forall y\, \exists z\, (x \in z \wedge y \in z)$

… 7 more axiom/axiom families …

Example 3: Set theory

Question:
How ‘complete’ are those 9 axioms?

Answer based on 100 years of experience:
Amazingly complete!
Almost all true statements about **math** (GORM) can be deduced from them.

In particular, everything we will prove in 15-251!

So you **can** formalize all of GORM using ZFC + FOL Deductive Calculus.

However, it's super-painful to do by hand.
(Remember 1+1=2 on page **379**?!)

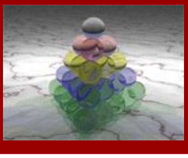
But we have computers now, you know…

Lord Wacker von Wackenfels
(1550−1619)

Kepler Conjecture

Kepler, 1611: As a New Year's
present (???) for his patron,
Lord Wacker von Wackenfels,
wrote a paper with this conjecture:

The densest way to pack spheres is like this:



Kepler Conjecture

2005:

Pittsburgher Tom Hales:
120 page proof in
Annals of Mathematics

Plus code to solve 100,000 distinct optimization problems, taking 2000 hours computer time.

Annals recruited a team of 20 referees.
They worked for 4 years.
Some quit. Some retired. One died.
In the end, they gave up.
But said they were "99% sure" it was a proof.

Kepler Conjecture

Hales: "I will code up a completely formal axiomatic deductive proof, checkable by computer."

Open source "Project Flyspeck": 2004—

Just finished last August!!

Computer-assisted proof

Proof assistant software like
HOL Light, Mizar, Coq, Isabelle, Agda
does two things:

1. Checks that a proof deduced in FOL Deductive Calculus (or typed lambda calculus theory) is valid.
2. Helps user code up such proofs.

Developing proof assistants is an active area of CS theory research, especially at CMU!

Computer-assisted proof

Suppose, e.g., HOL Light certifies a formal proof.
Can you trust it?

- You don't need to trust the million-line proof.
- You don't need to trust the process used to generate that proof.
- You just need to trust HOL Light's 430-line program for verifying FOL deductions.

Computer-formalized proofs

Fundamental Theorem of Calculus *(Harrison)*

Fundamental Theorem of Algebra *(Milewski)*

Prime Number Theorem *(Avigad @ CMU, et al.)*

Gödel's Incompleteness Theorem *(Shankar)*

Jordan Curve Theorem *(Hales)*

Brouwer Fixed Point Theorem *(Harrison)*

Four Color Theorem *(Gonthier)*

Feit-Thompson Theorem *(Gonthier)*

Kepler Conjecture *(Hales++)*

Proof of the Four Color Theorem

```
Variable R : real_model.
Theorem four_color : (m : (map R))
     (simple_map m) -> (map_colorable (4) m).
Proof.
Exact (compactness_extension four_color_finite).
Qed.
```

+ about 60,000 more lines

Study Guide

First order logic:

- well-formed sentences and vocabulary
- interpretations
- satisfiability/tautology
- the idea of Deductive Calculus and Gödel’s Completeness Thm.
- entailment