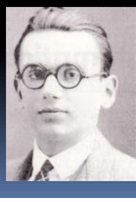
15-251: Great Theoretical Ideas in Computer Science
Fall 2014, Lecture 3 Venkat Guruswami

Axiomatic Systems & Logic

$$\frac{P,\ P \rightarrow Q}{Q}$$

$\Rightarrow$



Administrative stuff

I (Venkat) will be giving the next 6 lectures (3 weeks): Logic, Proofs, Counting, Games.

My office hours:
Thursdays, 1:15-2:45pm (before class)

The class is full, so not accepting people from waitlist (this could change after HW1 is due)

People high on waitlist can submit HW1 via email to correspondent TA as instructed by Prof. Adamchik. You're also welcome to sign up for Piazza.

In mathematics, sometimes your intuition can be quite wrong.

Here's a *theorem* (called Banach-Tarski paradox):

A solid ball in 3-dimensions can be cut up into six non-overlapping pieces, so that these pieces can be moved around & assembled into two identical copies of the original ball.

So it is important to:

Formalize concepts, give precise definitions

Make implicit assumptions explicit

Write careful proofs, where every step can be checked carefully.

Even "mechanically"

using a "computing machine", if you will

This week, we will talk a bit about formal logical reasoning and proofs.

TODAY

Part 1: Axiomatic systems.

Part 2: Propositional logic.

Part 3: First order logic. (just some basics)

Axiomatic Systems

An ATM has \$2 bills and \$5 bills
What dollar amounts can it dispense?

2
5
7 = 2+5
4 = 2+2
10 = 5+5

6 = 4+2
8 = 6+2
10 = 8+2
12 = 10+2

…all even amounts

9 = 4+5
11 = 6+5
13 = 8+5
15 = 10+5

…any odd amount at least 5

Cannot make 1 or 3

∴ All m in N except 0, 1, 3.

This is an example of an axiomatic system.

Initial amounts (2 & 5): "axioms"

"If you can make x and y, you can make x+y." "deduction rule"

The quantities you can make: "theorems"

In *this* axiomatic system: x is a "theorem" ⇔ $x \neq 0,1,3$

Different axioms ⇒ Different theorems

axioms = {0,2}:

⇒ theorems = all even natural #'s

axioms = {10,30}:

⇒ theorems = all positive multiples of 10

axioms = {2,3}:

⇒ theorems = all natural #'s except 0,1

Another axiomatic system

"Vocabulary": all strings using symbols **(,)**

axiom: **()**

deduction rules:

- WRAP: from S, deduce **(S)**
- CONCAT: from S, T, deduce ST

theorems: **(), (()), ((())), (((()))), …**

()(), ()()(), ()(()), …

Example: Show that **(()(()))** is a theorem.

Deduction:

1.	**()**	axiom
2.	**(())**	WRAP line 1
3.	**()(())**	CONCAT lines 1,2
4.	**(()(()))**	WRAP line 3

Each line (theorem) either an axiom, or is formed by applying deduction rule to previous theorems.

Example: Show that **()))** is NOT a theorem.

Claim: any theorem has equally many **(** and **)**

Proof sketch:
True for the axiom.
WRAP: If S has equally many, so does **(S)**
CONCAT: If S, T have equal, so does ST

Formal proof:

structural induction
(or strong induction on # of steps in deduction)

Exercise: Write a formal proof using structural induction.

For comparison, here is a proof by induction…

For $k \geq 1$, let F_k be the statement "any theorem derived in exactly k lines has equally many **(,)**".

The base case is $k = 1$. F_1 is true because a 1-line deduction must be an axiom, and the only axiom, **()**, has equally many **(,)**.

For general $k > 1$, let us suppose that F_i is true for all $1 \leq i < k$. For the induction step, we must show that F_k is true.

So suppose W is a theorem derived at the end of a k-line deduction.

The final line of this deduction (which derives W) is either an axiom, an application of WRAP to some previous line $j < k$, or an application of CONCAT to some two previous lines, $j_1, j_2 < k$. We verify that W has equally many **(,)** in all three cases.

In case the k^{th} line is an axiom, W must be **()**, which has equally many **(,)**.

In case the k^{th} line is WRAP applied to line $j < k$, we have W = **(S)**, where S is the theorem on line j. Since F_j is true by assumption, S has the same number of **(,)** — say c each. Then W has c+1 many **(** and c+1 many **)**, an equal number.

In case the k^{th} line is CONCAT applied to lines $j_1, j_2 < k$, we have $W = T_1T_2$ where T_1 is the theorem on line j_1 and T_2 is the theorem on line j_2. Since F_{j_1} is true by assumption, T_1 has the same number of **(,)** — say d_1 each. Similarly T_2 has the same number of **(,)** — say d_2 each. Hence W has d_1+d_2 many **(** and d_1+d_2 many **)**, an equal number.

In each of the three cases we have shown W has an equal number of **(,)**. Thus F_k is indeed true. The induction is complete.

Soundness and Completeness

Truth concept [a subset of strings over (,)]:
"There are equal numbers of (and) in the string"

This axiomatic system is **"sound"** for above truth concept.

- All theorems are "true"

Is it **"complete"** for above truth concept?

- i.e., are *all* "true" strings also theorems?

Question: Is **())(()** a theorem?

Answer: No.

Claim: a string of **(,)** is a theorem in this system *if and only if* it's a sequence of "balanced parentheses".

Proof: Exercise (or ask one of the course staff)

That is, this axiomatic system is **sound & complete** for the truth concept: *"The parens are balanced"*

Axiomatic systems: summary

- Vocabulary (or universe) (numbers, strings, tiles, graphs, …
  Elements called *expressions.*
- Axioms: initial set of expressions.
- Deduction rules: rules for obtaining new expressions from old ones.
- Theorem: an obtainable expression.
- Typical problems: Is X a theorem?
  Show Y is not a theorem.
  Is it sound/complete for some "truth" concept?
  "Characterize" the set of all theorems.

Logic

Logic: a formal game played with symbols which turns out to be useful for modeling mathematical reasoning.

Math: a formal game played with symbols which turns out to be useful for modeling the world.

0^{th} order logic

AKA propositional logic

A model for a simple **subset** of mathematical reasoning

"Not, And, Or, Implies, If And Only If"

An English statement that can be true or false	Propositional variable: a symbol (letter) representing it
"Potassium is observed."	k
"Hydrogen is observed."	h
"Pixel 29 is black."	p_{29}
"It's raining."	r

Compound sentence	Propositional formula
Potassium is not observed.	$\neg k$
At least one of hydrogen and potassium is observed.	$(h \vee k)$
If potassium is observed then hydrogen is also observed.	$(k \rightarrow h)$
If I'm not in 251 lecture then I'm preparing the lecture, and if I'm not preparing the lecture then I'm thinking about HW problems	$((\neg l \rightarrow p) \wedge (\neg p \rightarrow w))$

Formally, formulas are strings made up of:

(	(punctuation)
)	(punctuation)
$\neg$	(not)
$\wedge$	(and)
$\vee$	(or)
$\rightarrow$	(implies)
$\leftrightarrow$	(if and only if)
$x_1, x_2, x_3, \ldots$	(variable symbols)

Propositional formula

= A string which is syntactically "legitimate".

Propositional formula	not a prop. formula
x_1	$x_1 \wedge$
$((x_1 \wedge (x_3 \rightarrow \neg x_2)) \vee x_1)$	$))x_2 \rightarrow \rightarrow$
$\neg((x_{10} \leftrightarrow x_{11}) \wedge (x_2 \rightarrow x_5))$	$((x_1 \wedge (x_3 \rightarrow \neg x_2)) \neg x_1)$

Propositional formulae

Formally, propositional formulae are defined by an axiomatic system!

axioms: x_1, x_2, x_3, …

deduction rules:
- from A, can obtain $\neg$A
- from A, B can obtain $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$

Definition: A formula is a propositional formula (aka "well-formed" formula (WFF)) *if and only if* it is a 'theorem' in this system.

The "meaning" of these connectives

$(A \wedge B)$ "A and B"

true if both A and B are true

A	B	A∧B
T	T	T
T	F	F
F	T	F
F	F	F

$(A \vee B)$ "A or B"

true if at least one of A and B is true

A	B	A∨B
T	T	T
T	F	T
F	T	T
F	F	F

$\neg$A "not A"

true if A is false

A	¬A
T	F
F	T

The "meaning" of these connectives

(A → B) "if A then B"
"A implies B"

what are the rules for this?

A	B	A→B
T	T	T
T	F	F
F	T	T
F	F	T

Equivalent to (¬A ∨ B)

(A↔B) "A if and only if B"

same as (A→B) and (B→A)

A	B	A↔B
T	T	T
T	F	F
F	T	F
F	F	T

Let's talk about **TRUTH**.

"If potassium is observed then carbon and hydrogen are also observed."

(k→(c∧h))

Q: Is this statement true?

A: Depends. The question is ill specified.

"If potassium is observed then carbon and hydrogen are also observed."

(k→(c∧h))

Whether this statement/formula is true/false depends on whether the variables are true/false ("state of the world").

If k = **T**, c = **T**, h = **F**… … the formula is **False**.

If k = **F**, c = **F**, h = **T**… … the formula is **True**.

Truth assignment ς :
assigns **T** or **F** to each variable

Extends to give a truth value ς [S] for any formula S by (recursively) applying these rules:

A	B	¬A	(A∧B)	(A∨B)	(A→B)	(A↔B)
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

Recursive Evaluation for S

```
eval(formula S, input ς from {T,F}^n)
{
    If (S == "T") return T;
    if (S == "F") return F;
    if (S == "S_1 ∧ S_2")
        return eval(S_1,ς) ∧ eval(S_2,ς);
    …
    …
}
```

Truth assignment example

$S = (x_1 \rightarrow (x_2 \wedge x_3))$

ς : $x_1 = \mathbf{T}$, $x_2 = \mathbf{T}$, $x_3 = \mathbf{F}$

$\varsigma[S] = (\mathbf{T} \rightarrow (\mathbf{T} \wedge \mathbf{F}))$

$\varsigma[S] = (\mathbf{T} \rightarrow \mathbf{F})$

$\varsigma[S] = \mathbf{F}$

Satisfiability

ς satisfies S:

$\varsigma[S] = \mathbf{T}$

S is satisfiable:

there exists ς such that $\varsigma[S] = \mathbf{T}$

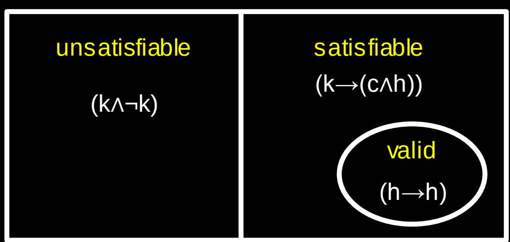
S is unsatisfiable:

$\varsigma[S] = \mathbf{F}$ for all ς

S is valid (AKA a tautology):

$\varsigma[S] = \mathbf{T}$ for all ς

All well-formed formulas



"Potassium is observed and potassium is not observed."

"If potassium is observed then carbon and hydrogen are observed."

"If hydrogen is observed then hydrogen is observed."

Valid: automatically true, for 'purely logical' reasons

Unsatisfiable: automatically false, for purely logical reasons

Satisfiable (but not valid): truth value depends on the state of the world

Example: S = (p ∧ (p → q)) → q

Truth table

p	q	p → q	p ∧ (p → q))	(p ∧ (p → q)) → q
T	T			
T	F			
F	T			
F	F			

Example: S = (p ∧ (p → q)) → q

Truth table

p	q	p → q	p ∧ (p → q))	(p ∧ (p → q)) → q
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

Formula S is **valid**!

$S = ((x \to (y \to z)) \leftrightarrow ((x \wedge y) \to z))$

Truth table

x	y	z	$((x\to(y\to z))\leftrightarrow((x\wedge y)\to z))$
F	F	F	T
F	F	T	
F	T	F	
F	T	T	
T	F	F	
T	F	T	
T	T	F	
T	T	T	

S is **satisfiable**!

$S = ((x \to (y \to z)) \leftrightarrow ((x \wedge y) \to z))$

Truth table

x	y	z	$((x\to(y\to z))\leftrightarrow((x\wedge y)\to z))$
F	F	F	T
F	F	T	T
F	T	F	T
F	T	T	T
T	F	F	T
T	F	T	T
T	T	F	T
T	T	T	T

S is **valid**!

Deciding Satisfiability (or Validity)

Truth table method:

Pro: Always works

Con: If S has n variables, takes ≈ 2^n time

Conjecture: (stronger than P ≠ NP)

There is **no** $O(1.999^n)$ time algorithm that works for every formula.

But for a *given* formula, sometimes you can prove/disprove satisfiability cleverly.

Quick recap

propositional formulas

n-variable formula maps each possible "world" in $\{T,F\}^n$ into either T or F

Some formulas are "truths" (tautologies): they are true in all possible 2^n worlds

Can check if a formula is a tautology in ≈ 2^n time by truth table method.

Truth table method for proving tautologies

SOME CONS

Does not give much "intuition"

Even simple things have very long proofs

$((p_1 \to p_2) \wedge (p_2 \to p_3) \wedge \ldots \wedge (p_{n-1} \to p_n)) \to (p_1 \to p_n))$

Does not scale to non-Boolean proofs.

If we want to prove things about all the naturals, then we're in trouble with brute-force.

A "more natural" way to prove things…

Let us start with a simple tautology

$(\neg A \vee A)$ we'll call this an "axiom"

And use one of these rules at each step:

$$\frac{A \vee (B \vee C)}{(A \vee B) \vee C} \text{ associativity} \qquad \frac{A \vee A}{A} \text{ contraction} \qquad \frac{A}{B \vee A} \text{ expansion}$$

$$\frac{(A \vee B),\ (\neg A \vee C)}{(B \vee C)} \text{ cut rule}$$

we'll call these "inference rules"

Whatever we can prove, we'll call "theorems"

Proof of commutativity rule $\frac{A \vee B}{B \vee A}$

$A \vee B$	(hypothesis)	(1)
$\neg A \vee A$	(axiom)	(2)
$B \vee A$	(cut rule to 1,2)	

Proof of new expansion rule $\frac{A}{A \vee B}$

A	(hypothesis)
$B \vee A$	(expansion rule)
$A \vee B$	(commutativity)

Proof of "modus ponens" $\frac{A, A \rightarrow B}{B}$

Since the logical system does not have "$\rightarrow$"
we define it to be $\neg A \vee B$

A	(hypothesis)	(1)
$A \rightarrow B$	(hypothesis)	
$\neg A \vee B$	(def. of $\Rightarrow$)	(2)
$A \vee B$	(apply expansion to 1)	(3)
$B \vee B$	(cut rule to 2,3)	
B	(contraction)	

What is a proof?

A sequence of statements,
each of which

is an axiom,

or a hypothesis,

or follows from previous statements
using an inference rule

Recap: A logical System for Propositions

Axiom:

$(\neg A \vee A)$

Inference Rules:

$\frac{A \vee (B \vee C)}{(A \vee B) \vee C}$ associativity

$\frac{A \vee A}{A}$ contraction

$\frac{A}{B \vee A}$ expansion

$\frac{(A \vee B), (\neg A \vee C)}{(B \vee C)}$ cut rule

(well-formed) propositional formulas

some formulas are tautologies ("truths")	some formulas are "theorems"
$p \vee \neg p$	$p \vee \neg p$
$(p \wedge (p \rightarrow q)) \rightarrow q$	$(p \wedge (p \rightarrow q)) \rightarrow q$
can check by truth-table	these are formulas for which we can give proofs

For this logical system and propositional formulas

Are all theorems "true" (i.e., tautologies)?

Yes. (easy proof by induction)

Yay! Our logical system is "sound". We only prove truths.

Are all tautologies theorems?

Yes. (proof lot more involved)

Double yay! Our logical system is "complete". We can prove all the truths via inference rules.

This logical system is

sound — **"all theorems are true"**

and

complete — **"all truths are theorems"**

for propositional truths (tautologies)

Proving tautologies by hand

For small examples, eg. in your problems,
you can prove a formula is valid
by simplifying the formula by hand
(similar to calculating arithmetic expressions)

Logical Equivalence

Definition:

Prop. formulas S and T are **equivalent**, written $S \equiv T$,
if $\varsigma[S] = \varsigma[T]$ for all truth-assignments ς.

⇒ their satisfiability/validity is the **same**

Example equivalences

$\neg(x \wedge y) \equiv (\neg x \vee \neg y)$

$\neg(A \vee B) \equiv (\neg A \wedge \neg B)$

$A \rightarrow B \equiv (\neg A \vee B)$

$(A \vee B) \equiv (B \vee A)$

$((A \vee B) \vee C) \equiv (A \vee (B \vee C))$

remark: so it's okay to write $(A \vee B \vee C)$

$A \vee A \equiv A$

$\neg\neg A \equiv A$

$A \leftrightarrow B \equiv ((A \rightarrow B) \wedge (B \rightarrow A))$

$((A \wedge B) \vee C) \equiv ((A \vee C) \wedge (B \vee C))$

etc.

Problem: Show that $(((x \rightarrow y) \wedge x) \rightarrow y)$ is valid.

Solution 1: Truth-table method

Solution 2: Use equivalences:

$(((x \rightarrow y) \wedge x) \rightarrow y)$

$\equiv \neg((x \rightarrow y) \wedge x) \vee y$ (using $A \rightarrow B \equiv \neg A \vee B$)

$\equiv (\neg(x \rightarrow y) \vee \neg x) \vee y$ (using $\neg(A \wedge B) \equiv \neg A \vee \neg B$)

$\equiv \neg(x \rightarrow y) \vee (\neg x \vee y)$ (using $(A \vee B) \vee C \equiv A \vee (B \vee C)$)

$\equiv \neg(\neg x \vee y) \vee (\neg x \vee y)$ (using $A \rightarrow B \equiv \neg A \vee B$)

$= \neg S \vee S$, where $S = (\neg x \vee y)$.

And a formula of form $\neg S \vee S$ is clearly valid.

First order logic

A model for **pretty much all** mathematical reasoning

"Not, And, Or, Implies, If And Only If"

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

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

"Ben is taller than everyone":

∀x IsTaller(**Ben**,x)

variable: stands for an object (person)

constant name: stands for a particular object

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

"Ben is taller than everyone":

∀x IsTaller(**Ben**,x)

"Ben is taller than everyone else":

∀x(¬(x=**a**)→ IsTaller(**Ben**,x))

0th order logic, as usual

equality (of objects)

"Ben is taller than everyone":

∀x IsTaller(**Ben**,x)

"Ben is taller than everyone else":

∀x(¬(x=**Ben**)→ IsTaller(**Ben**,x)

"Ben's dad is taller than everyone else's dad":

∀x(¬(x=**Ben**)→ IsTaller(Father(**Ben**),Father(x)))

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

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

Vocabulary from the previous slide:

one constant-name: **Ben**
one function-name: Father(·)
one relation-name: IsTaller(·, ·)

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 "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 means.

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

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:

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.

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

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**

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

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 / Validity

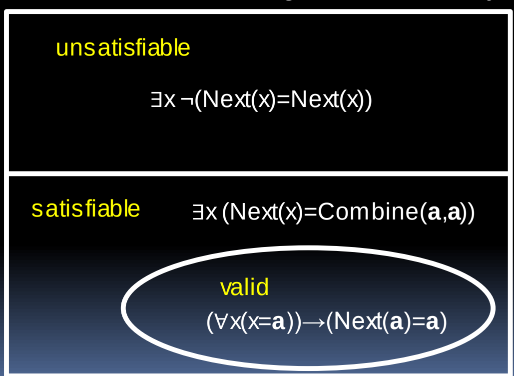
Interpretation I satisfies sentence S:
I [S] = **T**

S is satisfiable:
there exists I such that I [S] = **T**

S is unsatisfiable:
I [S] = **F** for all I

S is **valid**:
I [S] = **T** for all I

All sentences in a given vocabulary



Valid: automatically true,
for ‘purely logical’ reasons

Unsatisfiable: automatically false,
for purely logical reasons

Satisfiable (but not valid):
truth value depends
on the interpretation
of the vocabulary

(∃y ∀x (x=Next(y))) → (∀w ∀z (w=z))

Problem 1: Show this is satisfiable.

Let’s pick this interpretation:
Universe = integers, Next(y) = y+1.

Now (∃y∀x (x=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.**

(∃y ∀x (x=Next(y))) → (∀w ∀z (w=z))

Problem 2: Is it valid?

There is no “truth table method”.

You can’t enumerate all possible interpretations!

You have to use some cleverness.

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

Problem 2: Is it valid?

Solution: Yes, it **is valid**!

Proof: Let **I** be any interpretation.

If **I** [∃y ∀x (x=Next(y))] = **F**,
then the sentence is **True**.

If **I** [∃y ∀x (x=Next(y))] = **T**,
then every object equals Next(y).

In that case, **I** [∀w ∀z (w=z)] = **T**.

So no matter what, **I** [the sentence] = **T**.

Axiomatic System for Validity?

Can we find axioms & deduction rules so that
set of theorems = set of valid sentences ?

A ridiculous way:
Let axioms = "set of all valid sentences".

That is dumb because we at least want an
algorithmic way to check if
a given expression is an axiom.

Axiomatic System for Validity?

Open any textbook on logic.
You'll see an axiomatic system like this:

axioms:
1. A∨¬A for any sentence A
2. any 0^{th}-order tautology,
   with sentences for variables
3. ∀x ∀y ((x=**a**∧y=**b**)→(Func(x,y)=Func(**a**,**b**)))
4. IsR(**a**)→(∃x IsR(x))
5. blah blah blah, bunch more obviously valid kinds of sentences (algorithmically checkable)

deduction rule: from A and A→B can deduce B

Axiomatic System for Validity?

Let's call this the
"LOGIC TEXTBOOK" axiomatic system.

axioms:
1. A∨¬A for any sentence A
2. any 0^{th}-order tautology,
   with sentences for variables
3. ∀x ∀y ((x=**a**∧y=**b**)→(Func(x,y)=Func(**a**,**b**)))
4. IsR(**a**)→(∃x IsR(x))
5. blah blah blah, bunch more obviously valid kinds of sentences (algorithmically checkable)

deduction rule: from A and A→B can deduce B

Axiomatic System for Validity?

Let's call this the
"LOGIC TEXTBOOK" axiomatic system.

(Usually called a
"Hilbert axiomatic system")

Easy claim: any 'theorem' is **valid** sentence.

Question: is every valid sentence a 'theorem'?

Kurt Gödel

His PhD thesis: Yes!

"Gödel's COMPLETENESS Theorem"

Consequence:

There is a computer algorithm which finds a **proof** of any **valid** logical sentence.

The set of logically valid sentences is interesting, but it's not THAT interesting.

More typical use of first order logic:

1. Think of some universe you want to reason about.
2. Invent an appropriate vocabulary (constants, functions, relations).
3. ADD in some axioms which are true under the interpretation you have in mind.
4. See what you can deduce!

Example 1: Euclidean geometry

constant-names, function-names: none

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

extra 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)∧IsSameLength(x_2,z,y_1,y_2)
… 7 more …

Euclid

Alfred Tarski

Cool fact: this deductive system is **complete** for Euclidean geometry.

I.e., every **true** statement about Euclidean geometry is **provable** in this system.
"Decidability of the *theory of real closed fields*"

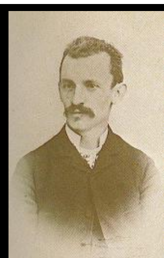
Example 2: Arithmetic of ℕ

constant-name: **0**

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

extra 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**)∧($\forall x$ F(x)→F(Successor(x)))) → $\forall x$ F(x)



Giuseppe Peano

Peano arithmetic is sound (i.e., every 'theorem' is a valid statement about arithmetic of natural numbers)

Is it complete for truths about natural numbers?

We’ll be back…

Example 3: Set theory

constant-names, function-names: none

relation-name: IsElementOf(x,y) [“x∈y”]

extra axioms, catchily known as “ZFC”:

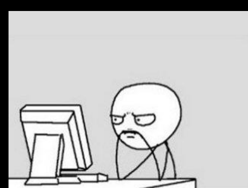
$\forall x \forall y\ ((\forall z\ \ z{\in}x \leftrightarrow z{\in}y) \ \rightarrow\ x = y)$

$\forall x \forall y\ \exists z\ (x{\in}z \wedge y{\in}z)$

… 7 more axiom/axiom families …

Empirical observation:
Almost all true statements about **MATH** can be formalized & deduced in this system.

Including every single fact we will prove in 15-251 (though we will work at a “higher level” of abstraction)



Study Guide

Axiomatic systems:
- definitions of axiom, deduction rules, theorems
- soundness & completeness

0^{th}-order logic:
- propositional formulas
- truth assignments
- valid/satisfiable
- truth-table method
- equivalences

1^{st}-order logic:
- understand examples
- interpretations
- valid/satisfiable