18-734/08-673: Foundations of Privacy
Recitation on Logic

Lay Kuan Loh

September 2, 2016

Administrative

- Thanks to everyone who posted about Privacy Policies on Piazza!

- Projects
  – See Piazza for a list of possible projects
  – Form groups of 2 or 3
  – You can propose your own project but must discuss it with the instructors
  – Use Piazza "Search for Teammates" function to find partners if necessary

Learning goals

- Translating declarative English sentences into logical formulas
  - "My password is secure"
- Understanding satisfiability and validity in propositional and first-order logic
  - Satisfiable: $x > 3$
  - Valid: $x = x$
- Using quantifiers:
  - Predicate: $x > 3$
  - Proposition: $\forall x(x > 3)$
  - Proposition: $\exists x(x > 3)$

These topics are explored more in Homework 1

Introduction to Propositional Logic

0^{th} order

Propositions

- Statements that are either true/false
- Which of these are propositions?
  1. “Google is collecting information about you online”
  2. Given that 5% of men and 80% of women use makeup, can we tell the boss that 90% of online users should be served ads about makeup?
  3. “Please don’t write down your password.”
  4. IsEncrypted(x) → SecurelyStored(x)

Logical Operators

Meaning	Logical Symbol
Not	$\neg$
And	$\wedge$
Or	$\vee$
Implies	$\rightarrow$
If and only if	$\leftrightarrow$

Translating sentences into logical notation

Propositional Statement	Propositional Variable
Has a Gmail account	Gm
Has a Facebook account	Fb
Has a MySpace account	Ms
Has a Yahoo account	Yh

Compound sentence	Propositional Formula
John does not have a Gmail account	$\neg Gm$
John has at least one account with Yahoo or Gmail	$Yh \vee Gm$
If John has a Facebook account, then he also has a Gmail account	$Fb \rightarrow Gm$
If John does not have Gmail account, then he has a Yahoo account; and if John do not have a Yahoo account, then he have a MySpace account.	$(\neg Gm \rightarrow Yh) \wedge (\neg Yh \rightarrow \mathrm{Ms})$

Well-Formed Formula (WFF)

- A string that is syntactically legitimate according to the inductive definition

- Base Case:
  - Single variables (such as K,H) are WFFs

- Inductive Case:
  - If A is a WFF, then $\neg A$ is a WFF
  - If A, B are WFFs, then $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$ are WFF

Semantics and the truth

Is this statement true?

"If John has a MySpace account, then he also has a Facebook account and a Gmail account"

$$Ms \rightarrow (Fb \wedge Gm)$$

Propositional Statement	Propositional Variable
Has a Gmail account	Gm
Has a Facebook account	Fb
Has a MySpace account	Ms
Has a Yahoo account	Yh

Truth assignments

- Truth assignment V: assigns T or F to each propositional variable
- Gives a truth value $V[\varphi]$ to any formula φ by applying these rules:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
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

Example truth assignment

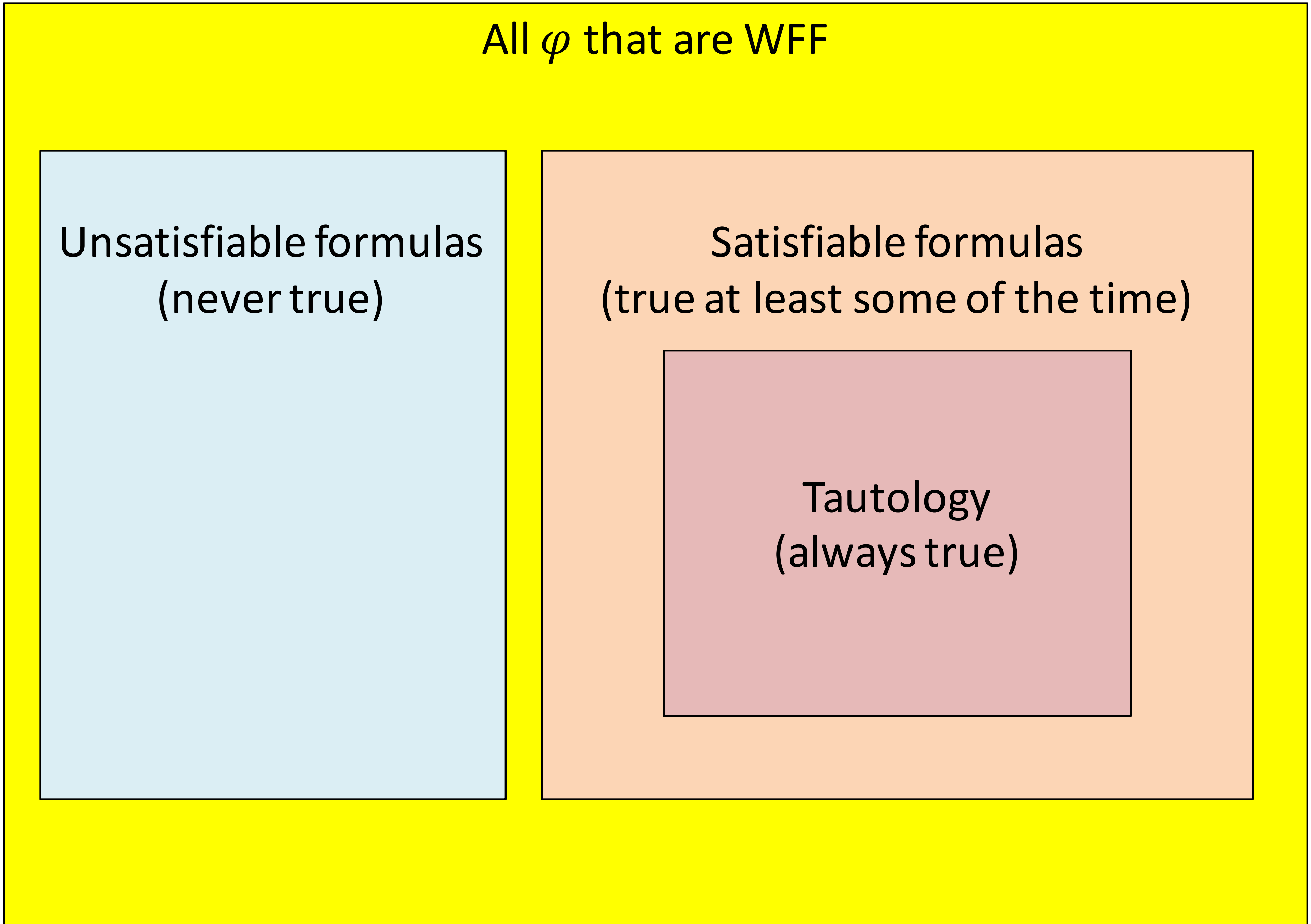
$$\varphi = Ms \rightarrow (Fb \wedge Gm)$$

Truth assignment V: Ms=T, Fb=T, Gm=F

$$V[\varphi] = T \rightarrow (T \wedge F) = T \rightarrow F = F$$

Satisfiability

- V satisfies φ: $\mathrm{V}[\varphi] = T$
  - *Example:* Given V[A=F, B=T] and $\varphi = A \rightarrow \neg$B, then $V[\varphi] = F \rightarrow \neg T = F \rightarrow F = T$
- φ is satisfiable: $\exists V$ s.t. $V[\varphi] = T$
  - Example:
    - Given V[A=F, B=T] and $\varphi = A \rightarrow \neg$B, then $V[\varphi] = F \rightarrow \neg T = F \rightarrow F = T$
    - Given V[A=T, B=T] and and $\varphi = A \rightarrow \neg$B, then $V[\varphi] = T \rightarrow \neg T = T \rightarrow F = F$
- φ is unsatisfiable: $\forall V, V[\varphi] = F$
  - Example: $\varphi = A \wedge \neg A$
- φ is a tautology: $\forall V, V[S] = T$
  - Example: $\varphi = A \vee \neg A$


All φ that are WFF
Unsatisfiable formulas
(never true)
Satisfiable formulas
(true at least some of the time)
Tautology
(always true)

Truth Table

$$\varphi = (x \to (\neg y \to z)))$$

x	y	z		$\neg y$	$\neg y \to z$	$x \to (\neg y \to z)$
T	T	T		F	T	T
T	T	F		F	T	T
T	F	T		T	T	T
T	F	F		T	F	F
F	T	T		F	T	T
F	T	F		F	T	T
F	F	T		T	T	T
F	F	F		T	F	T

Proof that $((x \to y) \land x) \to y$ is a tautology

- Method 1
  - Using truth table
  - *Semantic proof*
- Method 2
  - Using inference rules
  - *Syntactic proof*

Method 1: Truth Table

x	y		$x \rightarrow y$	$(x \rightarrow y) \land x$	$((x \rightarrow y) \land x) \rightarrow y$
T	T		T	T	T
T	T		F	F	T
T	F		T	T	T
T	F		F	F	T
F	T		T	T	T
F	T		T	F	T
F	F		T	F	T
F	F		T	F	T

Completeness of propositional logic

- [Soundness] All theorems that can be proven are tautologies
- [Completeness] All tautologies are theorems

First Order Logic (FOL)

Uses quantifiers such as
“for all” and “exists”

Logical Operators

Meaning	**Logical Symbol**
Not	$\neg$
And	$\wedge$
Or	$\vee$
Implies	$\rightarrow$
If and only if	$\leftrightarrow$

For All	$\forall$
Exists	$\exists$
Binary operators	$=, <, >, \leq, \geq$

Constants
Predicates
Functions

"*Alex's password* is different from *everyone's password*"

Variable
Stands for an object (person)

Quantifier

$$\forall x, \neg(Password(a) = Password(x))$$

Function name:
Maps object(s) → object

Constant name:
Stands for a particular object, "Alex"

"*Alex's password* is different from *everyone else's password*"

Propositional logic

$$\forall x, \neg(a = x) \rightarrow \neg(Password(a) = Password(x))$$

Function name:
Maps object(s) → object

"If there is someone else with the same password as Alex's password, Alex is not a security expert"

$$\exists x\ (\neg(x = a) \land \left(Password(x) = Password(a)\right) \rightarrow \neg SecurityExpert(Alex)$$

Predicate name:
Maps object(s) → T/F

Vocabulary

A collection of constant names, function names, and predicate names

“Alex’s father is smarter than everyone else’s father”

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

Constant name: a
Function name: Father
Predicate name: IsSmarter

Vocabulary

$$\exists x\ (Next(x) = a)$$

$$\forall x\ \forall y\ (IsPrior\big(x, Combine(a, y)\big) \rightarrow (Next(x) = y))$$

$$(\forall x\ IsPrior(x, Next(x))) \rightarrow (Next(a) = Next(a))$$

Vocabulary
Constant name: a
Function name: Next(.), Combine(.,.)
Predicate name: IsPrior(.,.)

Truth and Interpretations

$$\exists x(IsPatientOf(x, H) \rightarrow HasCancer(x))$$

- Truth of statement depends on the *interpretation* of the vocabulary

- ***Interpretation:*** Establishes what the vocabulary means

Interpretation

- Specifies a nonempty set ("universe") of objects
- Constant-name $\mapsto$ specific object
- Predicate-name $\mapsto$ actual predicate
- Function-name $\mapsto$ actual function

$$\exists x(IsPatientOf(x, H) \rightarrow HasCancer(x))$$

Interpretation #1:

- Universe = “All animals in Pittsburgh”
- H = “University of Pittsburgh Medical Center”
- x = “Rudolf”

False

$$\exists x(IsPatientOf(x, H) \rightarrow HasCancer(x))$$

Interpretation #2:

- Universe = “All human beings in Pittsburgh”
- H = “University of Pittsburgh Medical Center”
- x = “A cancer patient at the University of Pittsburgh Medical Center”

True

Satisfiability

- Interpretation I satisfies sentence φ: $I[\varphi] = T$
- φ is satisfiable: $\exists I$ s.t. $I[\varphi] = T$
- φ is unsatisfiable: $\forall I, I[\varphi] = F$
- φ is a tautology: $\forall I, I[\varphi] = T$

All well-formed sentences in a given vocabulary

Unsatisfiable

$\exists x \neg (x = x)$

Satisfiable

$\exists x (IsPatientOf\,(x, H) \rightarrow HasCancer(x))$

Tautology

$\forall x\ (x \rightarrow x)$

$$\exists x\ \forall y\ (y = sha1(x))$$
$$\rightarrow \forall z\ \forall w(sha1(z) = sha1(w))$$

Problem: Show this is satisfiable.

Interpretation

- Universe = All non-empty ASCII strings
- sha1(.) = sha1 algorithm used for encryption

Solution

- $\exists x\ \forall y\ (y = sha1(x))$ means "there exists an ASCII string x such that every ASCII string = sha1(x)"
- That is FALSE
- So the whole sentence becomes TRUE
- Hence the sentence is SATISFIABLE

$$\exists x\, \forall y\, (y = sha1(x)) \\ \rightarrow \forall z\, \forall w (sha1(z) = sha1(w))$$

Problem: Is this a tautology?

There is no “truth table” method

Not possible to enumerate all interpretations

$$\exists x\, \forall y\, (y = sha1(x)) \rightarrow \forall z\, \forall w\big(sha1(z) = sha1(w)\big)$$

Problem: Is this a tautology?

Solution: Yes

Proof:

- Let I be any interpretation
- Case $I\big[\exists x \forall y\big(y = sha1(x)\big)\big] = F$
  - Sentence becomes TRUE
- Case $I\big[\exists x \forall y\big(y = sha1(x)\big)\big] = T$
  - Every ASCII string equals $sha1(x)$
  - In that case,
    - $\forall z\, \forall w\big(sha1(z) = sha1(w)\big) = T$
- No matter what, $I[the\ sentence] = T$

Mechanical method to show that

$$\exists x\, \forall y\, \big(y = sha1(x)\big) \rightarrow \forall z\, \forall w\big(sha1(z) = sha1(w)\big)$$

is a tautology

Inference Rules

Temporal Logic

Propositional/First-Order logic vs Temporal logic

Propositional/First-Order logic

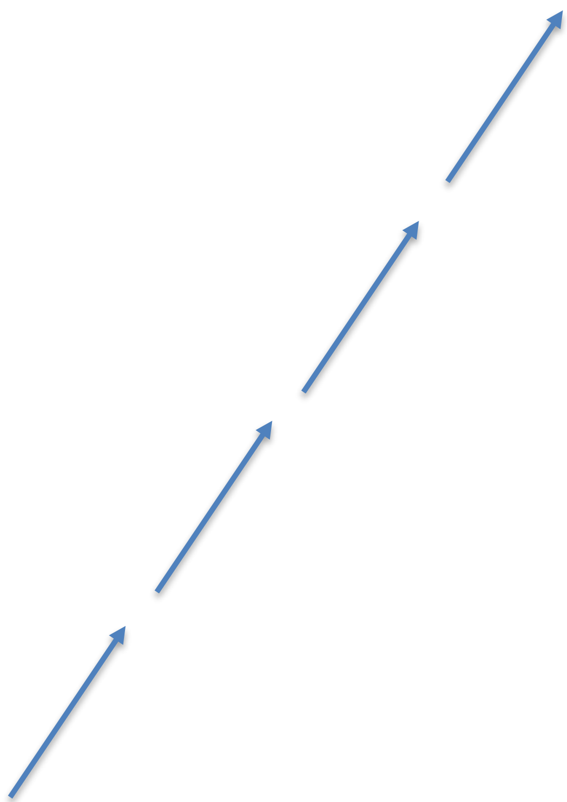
- One static state where formulae is evaluated
- Example:
  - S = “It is snowing”
  - Is k true? No, but only today.

Temporal logic

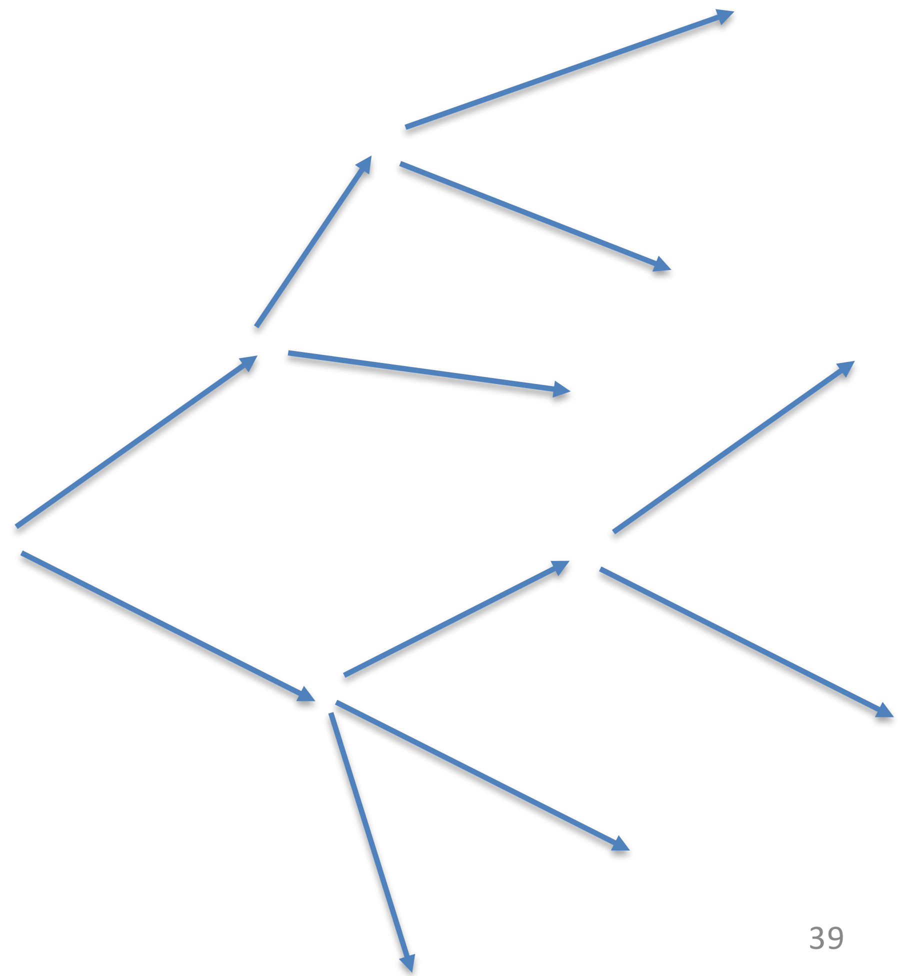
- Formalizes statements such as
  - It will snow someday in the future
  - It will snow everyday in future

What does time look like?

Linear Temporal Logic



Branching Temporal Logic



Linear Temporal Logic Operators - Unary

- $\bigcirc\varphi$:
  - Next: φ has to hold at the next state
  - Example:
    - Google will collect information about me tomorrow
    - $\bigcirc\ CollectInfo$
- $\Box\varphi$:
  - Globally: φ has to hold on the entire subsequent path
  - Example:
    - Google is always collecting information about you
    - $\Box\ CollectInfo$
- $\Diamond\varphi$
  - Finally: φ eventually has to hold somewhere on the subsequent path
  - Example:
    - Google will eventually collect information about me
    - $\Diamond\ CollectInfo$

Linear Temporal Logic Operators - Binary

- $\varphi \mathcal{U} \phi$
  - φ has to hold at least until ϕ, which holds at the current or future position
  - Example:
    - Google will collect information about you until you die
    - $CollectInfo\ \mathcal{U}\ Die$
- $\varphi \mathcal{R} \phi$
  - φ has to be true until and including the point where ϕ first becomes true. If ϕ never becomes true, φ must remain true forever.
  - Example
    - Google will collect information about you until you install a Privacy tool
    - $CollectInfo\ \mathcal{R}\ InstallPrivacyTool$

In the future, I will install a privacy tool, and then Google will never collect information about me again

$$\Diamond(InstallPrivacyTool \wedge \Box \neg CollectInfo)$$

Inference rules

(Optional)

What is a logical proof?

- A sequence of statements
- Each statement is an axiom / hypothesis, or follows from previous statements using an inference rule

Example Inference Rule

Assumptions

$$\frac{A \rightarrow C \qquad B \rightarrow C \qquad A \vee B}{C} \vee\text{-}\textsc{Elim}$$

Conclusion

A	“Need apples”
B	“Need beans”
C	“Went to convenience store”

Assumptions

- If I need asparagus, I will go to the convenience store
- If I need broccoli, I will go to the convenience store
- I need either asparagus or broccoli.

Conclusion

- I went to the convenience store

Checking that the rule makes sense

Assumptions

$$\frac{A \rightarrow C \qquad B \rightarrow C \qquad A \vee B}{C} \vee\text{-}\textsc{Elim}$$

Conclusion

Assumptions imply conclusion for all possible truth assignments to the propositional variables

A	B	C		A → C	B → C	A V B	(A→ C) ^ (B → C) ^ (A V B) → C
T	T	T		T	T	T	T
T	T	F		F	F	T	T
T	F	T		T	T	T	T
T	F	F		F	T	T	T
F	T	T		T	T	T	T
F	T	F		T	F	T	T
F	F	T		T	T	F	T
F	F	F		T	T	F	T

Propositional Logic: Building up a proof system systematically

$$\frac{A \to B \qquad A \to \neg B}{\neg A}\ \neg\text{-}\textsc{Intro} \qquad\qquad \frac{B \qquad \neg B}{A}\ \neg\text{-}\textsc{Elim}$$

$$\frac{\begin{matrix} A \\ \vdots \\ B \end{matrix}}{A \to B}\ \to\text{-}\textsc{Intro} \qquad\qquad \frac{A \to B \qquad A}{C}\ \to\text{-}\textsc{Elim}$$

$$\frac{A \qquad B}{A \wedge B}\ \wedge\text{-}\textsc{Intro} \qquad\qquad \frac{A \wedge B}{A}\ \wedge\text{-}\textsc{Elim}$$

$$\frac{A}{A \vee B}\ \vee\text{-}\textsc{Intro} \qquad\qquad \frac{A \to C \qquad B \to C \qquad A \vee B}{C}\ \vee\text{-}\textsc{Elim}$$

Propositional logic: Proof via inference rules

$$
\dfrac{
  \dfrac{\dfrac{(x \to y) \land x}{x \to y}\ \land\text{-}\textsc{Elim}
  \qquad
  \dfrac{(x \to y) \land x}{x}\ \land\text{-}\textsc{Elim}}
  {y}\ \to\text{-}\textsc{Elim}
}{((x \to y) \land x) \to y}\ \to\text{-}\textsc{Intro}
$$

First-order Logic: Building up a proof system systematically

$$\frac{P(a) \text{ arbitrary a}}{\forall x.P(x) \textsf{ true}}\ \forall\text{-}\textsc{Intro}$$

$$\frac{\forall x.P(x) \textsf{ true}}{P(a) \text{ arbitrary a}}\ \forall\text{-}\textsc{Elim}$$

$$\frac{P(a) \text{ for some element a}}{\exists x.P(x) \textsf{ true}}\ \exists\text{-}\textsc{Intro}$$

$$\frac{\exists x.P(x) \textsf{ true}}{P(a) \text{ for some element a}}\ \exists\text{-}\textsc{Elim}$$

Propositional logic: Proof via inference rules

Question:

- Given that Google collects information on all its users, and that John is a user of Google, does Google collect information about John?

Formalization

- $\forall x(UserOf(x, Google) \rightarrow CollectsInfo(Google, x))$
- $UserOf(John, Google)$
- $??CollectsInfo(Google, John)$

$$
\dfrac{\dfrac{\forall x, \mathsf{UserOf}(x, \mathsf{Google}) \rightarrow \mathsf{CollectsInfo}(\mathsf{Google}, x)}{\mathsf{UserOf}(\mathsf{John}, \mathsf{Google}) \rightarrow \mathsf{CollectsInfo}(\mathsf{Google}, \mathsf{John})}\ \forall\text{-}\textsc{Elim} \qquad \mathsf{UserOf}(\mathsf{John}, \mathsf{Google})}{\mathsf{CollectsInfo}(\mathsf{Google}, \mathsf{John})}\ \rightarrow\text{-}\textsc{Elim}
$$

Acknowledgements

- Slides based off past versions of 18-734 recitations, created by Arunesh Sinha and Amit Datta