Intro to Logic

September-21-11 10:27 AM

Propositional Logic

A proposition, or statement, is either true or false.

Valuations(interpretations)(of formulas
Determinations (proof) their relationship to valuations

Propositional expression

A propositional expression is a string sequence of symbols from

i) propositional variables: p, q, r
ii) connectives: {∧,∨, →, ¬}
iii) parentheses: { (,) }

Well-Formed Formula (WFF)

A WFF is as follows:

1. A propositional variable is a WFF by itself
2. If φ is a WFF and ψ is a WFF then
   $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi), (\neg\varphi)$
   are well formed.
3. Nothing else is a WFF

Lemma

Every WFF has an equal number of '(' and ')'

Lemma

Every non-empty proper prefix of a WFF has more "(" than ")"

- x is a prefix of y if for some z xz=y
- x is a proper prefix of y if xz=y and $x \neq y$

Proof: similar

Lemma

Every well-formed formula is a WFF in exactly one way.
If φ *is* $(\psi_1, *_1, \psi_2)$ and φ is $(\psi_3, *_2, \psi_4)$ when $* \in \{\wedge, \vee, \rightarrow, \neg\}$ and $\psi_1, \psi_2, \psi_3, \psi_4$ *are WFF*
Then $\psi_1 = \psi_3, \psi_2 = \psi_4, and\ *_1 = *_2$
Proof: induction in the structure of φ

Proof of Lemma

Induction on the structure of a WFF
Basis: if φ is P, then φ has no '(' or ')'

Suppose that φ_1 and φ_2 each have equal # of '(' and ')', say n_1 and n_2 respectively.

then $'(\varphi_1 \wedge \varphi)'$ has $1 + n_1 + n_2$ '(' and ')'. Same for all other cases.

Connectives

Conjunction: ∧

P	B	$P \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

Disjunction: ∨

P	B	$P \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

Implication: →

P	B	$P \rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1

Negation: ¬

B	$\neg B$
1	0
0	1

Formulae

10:13 AM

Valuation

A valuation is an assignment of 1 (true) or 0 (false) to each proposition variable.

Let t be a valuation. Each φ has a value under t, denoted φ^t as follows

1. For a variable p that $p^t = t(p)$
2. $(\varphi \wedge \psi)^t = \begin{cases} 1\ if\ \varphi^t = 1\ and\ \psi^t = 1 \\ 0\ otherwise \end{cases}$

   $(\varphi \vee \psi)^t = \begin{cases} 0\ if\ \varphi^t = \psi^t = 0 \\ 1\ otherwise \end{cases}$

   $(\varphi \to \psi)^t = \begin{cases} 0\ if\ \varphi^t = 1\ and\ \psi^t = 0 \\ 1\ otherwise \end{cases}$

   $(\neg\varphi)^t = \begin{cases} 0\ if\ \varphi^t = 1 \\ 1\ if\ \varphi^t = 0 \end{cases}$

Definition

Let φ be a WFF. Then φ is **valid** or a **tautology** iff for every t $\varphi^t = 1$ and **satisfiable** iff for some t $\varphi^t = 1$. **Unsatisfiable** iff for every t $\varphi^t = 0$

Equivalent

The formulae φ and ψ are equivalent iff for every valuation t, $\varphi^t = \psi^t$
Can say $\varphi \equiv \psi$

i.e. $\{\varphi\} \vDash \psi\ and\ \{\psi\} \vDash \varphi$

Adequate

Let C be a set of propositional connectives C is an adequate set of connectives iff for every WFF φ, $\exists\varphi_C$ such that φ_c uses only connectives in C, and φ is equivalent to φ_c

Definition

Let $\Sigma \subseteq WFF$ be a set of WFFs
$\Sigma^t = 1$ iff for every $\varphi \in \Sigma,\ \varphi^t = 1$
$\Rightarrow \emptyset^t = 1$

Logical Consequence

Let $\Sigma \subseteq WFF, \varphi \in WFF$
φ is a (logical) consequence of Σ, denote $\Sigma \vDash \varphi$ iff for every t if $\Sigma^t = 1 \to \varphi^t = 1$

Deductions

$\vdash$ relation between sets of formulas and formulas based on deduction rules.
$\Sigma \vdash \varphi$ means there exists a proof of φ using the formula in Σ

Examples of Equivalent Formulae

$\neg\neg p \equiv p$
$p \to q \equiv \neg p \vee q$

Example

If $\varphi \in \Sigma$ then $\Sigma \vDash \varphi$
If φ is valid then $\Sigma \vDash \varphi\ \forall\Sigma$

If Σ is finite then can determine whether $\Sigma \vDash \varphi$ by a truth table.

Equivalence & Consequence

September-22-11 9:59 AM

Lemma

Suppose that φ_1 is equivalent to φ_2 then $\neg\varphi_1$ is equivalent to $\neg\varphi_2$ and for every WFF ψ:
$(\varphi_1 \wedge \psi) \equiv (\varphi_2 \wedge \psi)$
$(\varphi_1 \vee \psi) \equiv (\varphi_2 \vee \psi)$
$(\varphi_1 \rightarrow \psi) \equiv (\varphi_2 \rightarrow \psi)$
$(\psi \rightarrow \varphi_1) \equiv (\psi \rightarrow \varphi_2)$

Also if $\varphi_1 \equiv \varphi_2, and\ \varphi_2 \equiv \varphi_3$ then $\varphi_1 \equiv \varphi_3$

Proof: By definition of value of a formula

Corollary

If $\varphi_1 \equiv \varphi_2$ then whenever φ_1 is a sub-formulae of ψ_1 and ψ_2 is ψ_1 but with φ_1 replaced with φ_2 then $\psi_1 \equiv \psi_2$

Proof: By induction on the structure of ψ_1 using lemma

Consequence

Let Σ be a set of formulae and A and B be formulae.
Then $\Sigma \vDash A \rightarrow B\ iff\ \Sigma \cup \{A\} \vDash B$

Proof of Consequence

(⇒)
$\Sigma \vDash A \rightarrow B$ mean for every t if $\Sigma^t = 1$ then $(A \rightarrow B)^t = 1$. Suppose this holds.
Consider a valuation t. Need to show how if $(\Sigma \cup \{A\})^t = 1 = 1\ then\ B^t = 1$
Case 1: $A^t = 0$. The implication is vacuously true.
Case 2: $A^t = 1$.
    If $(\Sigma \cup \{A\})^t = 1$ then $\Sigma^t = 1$ Thus $(A \rightarrow B)^t = 1$
    $A^t = 1\ and\ (A \rightarrow B)^t = 1$ so by definition of valuation of $\varphi \rightarrow \psi$ we must have $B^t = 1$ as required.
(⇐)
Suppose for every valuation t, if $(\Sigma \cup \{A\})^t = 1$ then $B^t = 1$
Need to show $(A \rightarrow B)^t = 1$
Have $B^t = 1$, thus $(A \rightarrow B)^t = 1$ by definition of →
Otherwise,
$(\Sigma \cup \{A\})^t \neq 1$
    case I: $A^t = 0$ Regardless of B^t $(A \rightarrow B)\hat{}t = 1$
    case II: $A^t = 1$...

Deduction in Propositional Logic

September-27-11 10:00 AM

Deduction System

A deduction system consists of axioms and (inference) rules.

Axiom

An axiom is a formula

Rule

A rule is a tuple of the form $< A_1, A_2, \dots, A_n >$ for some n and formulas $A_1, \dots, A_n$

Deduction (Formal Proof)

A deduction, or proof, in a deduction system S is a sequence of formulas with the following property:
In a proof $B_1, B_2, \dots, B_m$ for each $1 \le i \le m$
either B_i is an axiom of S
There is some sequence of $j_1, j_2, \dots, j_{n-1}, j_k < i \, \forall 1 \le k \le n-1$ s.t. $< B_{j_1}, \dots, B_{n-1}, B_i >$ is a rule of S

A deduction (in system S) for a set of formulas Σ is a sequence $B_1, \dots, B_m$ s.t. for each $1 \le i < n$
either B_i satisfies the conditions of a deduction
or $B_i \in \Sigma$

Our Deduction System

Rule of Inference

Want to be simple and few

Rule MP (modus ponens): For all formulae A and B
$< A, A \to B, B >$ is a rule
"From A and $A \to B$ deduce B"

Axioms

For all WFF A, B and C

1. $A \to (B \to A)$
2. $\left(A \to (B \to C)\right) \to \left((A \to B) \to (A \to C)\right)$
3. $(\neg A \to B) \to \left((\neg A \to \neg B) \to A\right)$

Yields

Suppose that there exists a proof (in a system S), whose last formula is φ, from a set Σ. Then we say Σ yields φ and write
$\Sigma \vdash_s \varphi \ \ (or\ \Sigma \vdash \varphi)$

e.g. $\{(p \to q)\} \vdash \left(r \to (p \to q)\right)$

If $\Sigma = \emptyset$, write $\vdash \varphi$

Formal Proof

Sequence of formulas s.t. each is an axiom, a hypothesis, or follows from earlier ones by an inference rule.

Lemma

Suppose that $\Sigma \vdash \varphi$ for each $\varphi \in \Gamma$. Then whenever $\Gamma \vdash \varphi$, also $\Sigma \vdash \gamma$

Deduction Theorem

For each set $\Sigma \subseteq$ WFF and $\varphi \in WFF, \psi \in WFF$
$\Sigma, \varphi \vdash \psi \ iff \ \Sigma \vdash (\varphi \to \psi)$

Example of Deduction

Let $\Sigma = \{p \to q\}$ a simple deduction for Σ

1. $(p \to q)$
2. $(p \to q) \to \left(r \to (p \to q)\right)$ $[Ax1]$
3. $\left(r \to (p \to q)\right)$

Example

For any WFF A
$\vdash A \to A$
Proof:

1. $\left(A \to \left((A \to A) \to A\right)\right)$ $[Ax1]$
2. $\left(A \to \left((A \to A) \to A\right)\right) \to \left(\left(A \to (A \to A)\right) \to (A \to A)\right)$ $[Ax2]$
3. $\left(\left(A \to (A \to A)\right) \to (A \to A)\right)$ $[MP{:}\,1\&2]$
4. $\left(A \to (A \to A)\right)$ $[Ax1]$
5. $(A \to A)$ $[MP{:}\,4\&3]$

Proof of Lemma

$A_1 A_2 \dots A_n$ a proof of φ from Γ
If $A_i = \varphi_i \in \Gamma$, there is a proof $B_{i_1}, \dots, B_{i_{n_1}}$ of φ_{i_j} from Σ
thus $B_{11}, \dots, B_{1n_1}, \dots, B_{n1}, B_{nn_n}$ is a proof of φ from Σ

Proof of Deduction Theorem

"$\Leftarrow$" Let $A_1 A_2 \dots A_n$ be a proof of $\varphi \to \psi$ from Σ
where $A_n = \varphi \to \psi$
By modus ponens $A_1 A_2 \dots A_n \varphi \psi$ is a proof of ψ from Σ, φ

"$\Rightarrow$"
To show: for every Σ, φ, ψ if $B_1, \dots, B_n$ is a proof of ψ from $\Sigma \cup \{\varphi\}$ then $\Sigma \vdash \varphi \to \psi$
use induction on n

Base case: $n = 1$. Then $\psi = B$, so ψ must either be in Σ, an axiom, or φ itself.
In the first two cases, $\Sigma \vdash \psi$
Since axiom 1 has instance $\psi \to (\varphi \to \psi)$
$\psi, \left(\psi \to (\varphi \to \psi)\right), (\varphi \to \psi)$ is a proof from Σ of $\varphi \to \psi$
If $\psi = \varphi$ need to show $\Sigma \vdash \psi \to \psi$ since $\vdash A \to A$ already done such a proof exists.

Induction hypothesis: Suppose the claim holds for all $n < m$
Consider $A_1, \dots, A_m = \psi$ from Σ, φ
By hypothesis have a proof
$\dots, \varphi \to A_1, \varphi \to A_2, \dots, \varphi \to A_{m-1}$ $\forall A_j \in \Sigma$
Case 1, 2 as in basis
Case 3 A_m is derived by MP from A_i and $A_j = A_i \to A_m$. Need to prove $\varphi \to A_m$
$\left(\varphi \to (A_i \to A_m)\right) \to \left((\varphi \to A_i) \to (\varphi \to A_m)\right)$
$M.P. (\varphi \to A_i) \to (\varphi \to A_m)$
$M.P. (\varphi \to A_m)$
■

Example

To show $\vdash \neg p \to (p \to q)$
By deduction theorem, this holds iff
$\neg p \vdash p \to q$ iff
$\neg p, p \vdash q$

Prove the last one:
Ax3: $(\neg q \to p) \to \left((\neg q \to \neg p) \to q\right)$
Ax2: $p \to (\neg q \to p)$
$\Sigma{:}\, p$
$MP{:}\, \neg q \to p$
$MP{:}\, (\neg q \to \neg p) \to q$
Ax2: $\neg p \to (\neg q \to \neg p)$
$\Sigma{:}\, \neg p$
$MP{:}\, \neg q \to \neg p$
$MP{:}\, q$

So $\neg p, p \vdash q \Rightarrow \vdash \neg p \to (p \to q)$

Example

To show $\vdash (\neg p \to \neg q) \to (q \to p)$
iff $\neg p \to \neg q \vdash q \to p$
iff $(\neg p \to \neg q), q \vdash p$
iff $q \vdash (\neg p \to \neg q) \to p$

Ax1: $q \vdash \neg p \to q$
Ax3: $q \vdash (\neg p \to q) \to \left((\neg p \to \neg q) \to p\right)$
$MP{:}\, q \vdash (\neg p \to \neg q) \to p$

Example

$\neg\neg p \vdash p$

$H: \neg\neg p$
$Ax1: \neg\neg p \rightarrow (\neg\neg\neg\neg p \rightarrow \neg\neg p)$
$MP: \neg\neg\neg\neg p \rightarrow \neg\neg p$
$Prev\ Ex: (\neg\neg\neg\neg p \rightarrow \neg\neg p) \rightarrow (\neg p \rightarrow \neg\neg\neg p)$
$MP: \neg p \rightarrow \neg\neg\neg p$
$Prev\ Ex: (\neg p \rightarrow \neg\neg\neg p) \rightarrow (\neg\neg p \rightarrow p)$
$MP: \neg\neg p \rightarrow p$
$MP: p$

Soundness & Completeness

11:09 AM

Soundness

For all Σ, φ

$\Sigma \vdash \varphi \Rightarrow \Sigma \vDash \varphi$

Theorem

Our deduction system is sound.

Completeness

$If\ \Sigma \vDash \varphi, then\ \Sigma \vdash \varphi$

Inconsistent

A set $\Sigma \subseteq WFF$ is inconsistent iff there is a WFF φ s.t. $\Sigma \vdash \varphi$ and $\Sigma \vdash \neg\varphi$

Lemma 1

If Σ is inconsistent, then for every WFF $\varphi, \Sigma \vdash \varphi$ and $\Sigma \vdash \neg\varphi$

Lemma 2

Suppose Σ, φ is inconsistent. Then $\Sigma \vdash \neg\varphi$

Maximal Consistent Set

Σ is a maximal consistent set iff

1) Σ is consistent, and
2) If a WFF φ is not in Σ then $\Sigma \cup \{\varphi\}$ is inconsistent.

Lemma 3

Let Σ be any maximal consistent set. For all φ, η

1) $\neg\varphi \in \Sigma$ iff $\varphi \notin \Sigma$
2) $\varphi \to \eta \in \Sigma$ iff $\varphi \notin \Sigma$ or $\eta \in \Sigma$ (or both)

Lemma 4

Suppose Σ is consistent. Then there is a maximal consistent set Σ' with $\Sigma \subseteq \Sigma'$

Lemma 5

Let Σ' be a maximal consistent set.

Define the valuation t by

$$p^t = \begin{cases} 1\ if\ p \in \Sigma' \\ 0 \text{ otherwise} \end{cases}$$

Then for all $\eta, \eta^t = 1$ iff $\eta \in \Sigma'$ (induction)

$\Sigma' \vDash \varphi \Rightarrow \Sigma' \vdash \varphi$

Theorem

If Σ is consistent, then Σ is satisfiable.

Corollary

$\Sigma \vdash \varphi$, then $\Sigma \vDash \varphi$

Proof: Contrapositive

Σ unsatisfiable $\Rightarrow \Sigma$ is inconsistent

Proof of Theorem

Induction on the length of a proof for $\Sigma \vdash \varphi$

Basis: $\varphi \in \Sigma$ Then $\Sigma \vDash \varphi$

φ is an axiom:

All axioms φ are valid $\to \Theta(\varphi)$ are valid

Induction step: suppose true for proofs of length n-1

Need M.P to preserve valuation to 1

$\Sigma \vDash \varphi_1\ and\ \Sigma \vDash \varphi_1 \to \varphi_2\ then\ \Sigma \vDash \varphi_2$

Proof of Lemma 1

Σ inconsistent implies an η s.t. $\Sigma \vdash \eta$ and $\Sigma \vdash \neg\eta$

Last time: $\eta, \neg\eta \vdash \varphi\ \forall\varphi \therefore \Sigma \vdash \varphi\ \forall\varphi$

Proof of Lemma 2

$\Sigma, \varphi \vdash \neg\varphi$

Deduction Theorem: $\Sigma \vdash \varphi \to \neg\varphi$

Recall: $\neg\neg\varphi \vdash \varphi$

thus $\Sigma, \neg\neg\varphi \vdash \neg\varphi$

and $\Sigma \vdash \neg\neg\varphi \to \neg\varphi$

Ax3: $\Sigma \vdash (\neg\neg\varphi \to \varphi) \to ((\neg\neg\varphi \to \neg\varphi) \to \neg\varphi)$

MP: $\Sigma \vdash (\neg\neg\varphi \to \neg\varphi) \to \neg\varphi$

MP: $\Sigma \vdash \neg\varphi$ ■

Proof of Lemma 3

1) If $\neg\varphi \in \Sigma$ then $\Sigma \vdash \neg\varphi$ thus $\Sigma \cup \{\varphi\}$ is inconsistent $\Rightarrow \varphi \notin \Sigma$
   Suppose $\neg\varphi \notin \Sigma$. By definition $\Sigma \cup \{\neg\varphi\}$ is inconsistent. Hence $\Sigma, \neg\varphi \vdash \varphi$ and $\Sigma \vdash \neg\varphi \to \varphi$
   Thus $\Sigma \vdash \varphi$ and have Σ is consistent so $\varphi \in \Sigma$

2) If $\varphi \notin \Sigma$ then $\neg\varphi \in \Sigma$
   Know $\neg\varphi \vdash \varphi \to \eta \Rightarrow \varphi \to \eta \in \Sigma$
   If $\eta \in \Sigma$ then know $\eta \vdash \varphi \to \eta \Rightarrow \varphi \to \eta \in \Sigma$
   If $\varphi \to \eta \in \Sigma$
   a) If $\varphi \notin \Sigma$ done
   b) If $\varphi \in \Sigma$ then $\Sigma \vdash \eta \therefore \eta \in \Sigma$

Proof of Lemma 4

Consider a list (enumeration) of all $WFFs\ \varphi_1, \varphi_2, \ldots$

Will define a sequence of sets $\Sigma_0, \Sigma_1, \ldots$ s.t. for each $i \geq 1, \Sigma \subseteq \Sigma_i$ and Σ_i is consistent and either $\varphi_i \in \Sigma_i$ or $\Sigma_i \cup \{\varphi_i\}$ is inconsistent.

Let $\Sigma_0 = \Sigma$. Suppose that $\Sigma_0, \Sigma_1, \ldots, \Sigma_{i-1}$ are defined

$$\Sigma_i = \begin{cases} \Sigma_{i-1} \cup \{\sigma_i\} \text{ if this is consistent} \\ \Sigma_{i-1} \text{ otherwise} \end{cases}$$

Let $\Sigma' = \bigcup_{i \geq 0} \Sigma_i$

Claim: Σ' is a maximal consistent set.

Proof:

1) Σ' is consistent
   Suppose that Σ' is inconsistent: for some φ, $\Sigma' \vdash \varphi$ and $\Sigma' \vdash \neg\varphi$
   Both proofs are finite, thus for some $j \in \mathbb{N}$ all formulae from the proof lie in Σ_j so $\Sigma_j \vdash \varphi$ and $\Sigma_j \vdash \neg\varphi$. Let j be the least such j with this property. $j \neq 0$ since $\Sigma_0 = \Sigma$ is consistent.
   Therefore Σ_{j-1} is consistent and Σ_j is not consistent, but this is impossible by construction.
2) Σ' is maximally consistent
   Suppose not $\Sigma' \cup \{\varphi\}$ is consistent, $\varphi = \varphi_i$ for some i
   Thus $\Sigma_i = \Sigma_{i-1} \cup \{\varphi_i\}$

Proof of Lemma 5

Suppose $\Sigma' \vDash \varphi$. t is an interpretation which satisfies Σ' so it must be that $\varphi^t = 1$. Therefore $\varphi^t \in \Sigma'$ so $\Sigma' \vdash \varphi$

$\Sigma' \vDash \varphi \Rightarrow \Sigma' \vdash \varphi$

Proof of Corollary

Suppose $\Sigma \nvdash \varphi$. Then create the maximally consistent set $\Sigma' \supseteq \Sigma \cup \{\neg\varphi\}$

Note that $\Sigma \cup \{\neg\varphi\}$ is consistent by lemma 2 (Since $\Sigma \cup \{\neg\varphi\}$ inconsistent $\Rightarrow \Sigma \vdash \varphi$)

Then $\Sigma' \nvdash \varphi$ so $\Sigma' \nvDash \varphi$. But since $\Sigma \subseteq \Sigma'$, $mod(\Sigma) \supseteq mod(\Sigma')$ so $\Sigma \nvDash \varphi$

Therefore

$\Sigma \vDash \varphi \Rightarrow \Sigma \vdash \varphi$

Models

October-11-11 10:01 AM

Model

In propositional logic, a model is a valuation for a formula φ. A model of φ is a valuation that satisfies φ: $\varphi^t = 1$

The set of models of φ is denoted $mod(\varphi)$

For $\Sigma \subseteq WFF\ mod(\Sigma) = \bigcap_{\varphi \in \Sigma} mod(\varphi)$

$mod(\varphi) = \{t \mid p^t = 1\}$
$mod(\varphi \wedge \eta) = mod(\varphi) \cap mod(\eta)$
$mod(\varphi \vee \eta) = mod(\eta) \cup \overline{mod(\varphi)} = mod(\neg\varphi \vee \eta)$

Lemma

$\Sigma \vDash \varphi\ iff\ mod(\Sigma) \subseteq mod(\varphi)$

Sequential Calculus (LK)

Notation has a concept of the method of deduction

$$\dfrac{\dfrac{A \quad B}{C} \quad D}{E}$$

Sequent

A sequent is Γ ⊢ Δ, where Γ and Δ are sets of WFFs

The intended meaning of "Γ ⊢ Δ" whenever every formula of Γ, then some (one or more) formula of Γ is true.

System LK

Identity Rules

Axiom

$$\dfrac{\square}{\Gamma, \varphi \vdash \varphi, \Delta}$$

Cut

$$\dfrac{\Gamma \vdash \varphi, \Delta \quad \Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta}$$

Logical Rules

¬L $\dfrac{\Gamma \vdash \varphi, \Delta}{\Gamma, (\neg\varphi) \vdash \Delta}$	**¬R** $\dfrac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \neg\varphi\Delta}$
→ L $\dfrac{\Gamma \vdash \varphi, \Delta \quad \Gamma, \eta \vdash \Delta}{\Gamma, (\varphi \to \eta) \vdash \Delta}$	**→ R** $\dfrac{\Gamma, \varphi \vdash \eta, \Delta}{\Gamma \vdash (\varphi \to \eta), \Delta}$

Soundness (Theorem)

If $\Gamma \vdash_{LK} \Delta$, then $\Gamma \vDash \Delta$
i.e. $mod(\Gamma) \subseteq \bigcup_{\varphi \in \Delta} mod(\varphi) \equiv MOD(\Delta)$

Proof: Induction on the structure of the proof of $\Gamma \vdash_{LK} \Delta$

Completeness (Theorem)

If $\Gamma \vDash \Delta$, then $\Gamma \vdash_{LK} \Delta$

Method: Show that if $\Gamma \vdash_{H} \varphi$ and $\varphi \in \Delta$ then $\Gamma \vdash_{LK} \Delta$
Induction on the length of the Hilbert Proof:
Induction steps: Show MP and axioms have equivalences in LK.

A step in the proof: Prove the deduction theorem for LK
$\Gamma, \varphi \vdash q, \Delta \; iff \; \Gamma \vdash \varphi \rightarrow \eta, \Delta$
$\Rightarrow (\rightarrow R)$
$\Leftarrow$ exercise

Theorem (Cut Elimination)

For every proof of a sequent $\Gamma \vdash \Delta$, there is a proof in LK that never uses "Cut"

Proof: Induction on the number of cuts and on the structure of formulas.

Modal Logic

October-13-11 10:09 AM

Sets

Cross Product

$S_1 \times S_2 = \{(a, b) \mid a \in S_1, b \in S_2\}$

k-ary Relation

Subset of $S_1 \times S_2 \times \cdots \times S_k$, $(a_1, a_2, \ldots, a_k)$
$R(a_1, \ldots, a_k)$ means $(a_1, \ldots, a_k) \in \mathbb{R}$

$F(a_1, \ldots, a_k) = a$ k-ary function: a (k+1)-ary relation s.t. if $(a_1, \ldots, a_k, a) \in F$ and $(a_1, \ldots, a_k, b) \in F$ then $a = b$

Relational Database

A relation is defined by a finite list of members
e.g.
(Joe, Id=2717, age=27)

Modal Logic

Mood → Modal
"φ is true"
"φ must be true"
"φ may be true"

Syntax

New unary operators (s) ◊, □

Definition

Of WFF in a modal (propositional) logic

1) If p is a propositional variable, p is a WFF
2) If φ and η are WFFs so are
   a. $(\neg\varphi)$
   b. $(\varphi \to \eta)$
   c. $(\Box\varphi)$
3) We may also write
   a. φ/η, which means $(\neg\varphi \to \eta)$
   b. $\varphi \wedge \eta$, which means $\neg(\varphi \to \neg\eta)$
   c. $\Diamond\, \varphi$, which means $\left(\neg(\Box(\neg\varphi))\right)$

Definition

A Modal interpretation (or **Kripke structure**) consists of a set W of "worlds", a relation R on $W \times W$ called the "**accessibility**" or "**visibility**" relation, and a function V that assigns a valuation to each world.

Frame

A frame is a pair W, R of a modal interpretation.

Definition

For a model interpretation $I = (W, R, V)$, a pointed model is a pair I, w where $w \in W$

Definition

For a pointed model I, w; for a variable p
$I, w \vDash P$ iff $V(w)(p) = 1$
For each WFF φ and η

$I, w \vDash \neg\varphi$ iff $I, w \nvDash \varphi$
$I, w \vDash \varphi \to \eta$ iff $I, w \vDash \eta$ or $I, w \nvDash \eta$
$I, w \vDash \Box\varphi$ iff for every $x \in W$ if $R(w, x)$ then $I, x \vDash \varphi$

Relation Example

function $g(a, b) = a + b : (0, 0, 0), (0, 1, 1), \ldots, (12, 27, 40), \ldots$
define $f_a(b) = a + b$; $f_3 = (0, 3), (1, 4), \ldots$
define $h(f, b) = f(b)$
$h(f_a, b) = f_a(b) = a + b = g(a, b)$
define $c(a) = f_a$; $c(a)(b) = f_a(b) = a + b$

Modal Logic Example

Design a print server

If a request is received by the server then the file will be printed.
Order to print?

- Fastest request first?
- Largest first?
- Smallest first?
- Earliest request first? FIFO <- This guarantees all documents will eventually be printed. The above do not.

Examples

$W = \{w\}, R = \{(w, w)\}$. Equivalent to propositional logic.

$W = \mathbb{N} = \{0, 1, \ldots\}$. (Possibly # of files in the print queue)
$R = \mathbb{N} \times \mathbb{N}$; $V(a, b) =$ some valuation
* I'm not sure what V(a, b) means. I think shorthand for V(a)(b) where b is a propositional variable index. *

$$W = \mathbb{N}; R = \{(a, b) \mid a \leq b\}; V(a) = p_i = \begin{cases} 1\ if\ i \geq a \\ 0\ if\ i \leq a \end{cases}$$

$p_i \to p_{i+1}$ is true for each $V(a)$

Modal Logic

October-18-11 10:15 AM

Meanings

□ "necessarily" or "always"
◊ "possibly" or "eventually"

Formulae

p a variable
$\neg\varphi, \varphi$ a formula
$\Box\varphi, \varphi$ a formula
$\varphi \rightarrow \eta, \varphi, \eta$ formulae

Frame

A **frame** is a pair W,R where W is a set whose elements are "worlds" and $R \subseteq W \times W$ is a relation.

Modal Interpretation

A modal interpretation (or Kripke structure) is a frame and a relation function.
$I = (W, R, V)$ where $V(w)$ is a valuation for each $w \in W$

Definition

An interpretation I and world w model a formula φ denoted $I, w \vDash \varphi$ as follows

1) $I, w \vDash p\ iff\ p^{V(w)} = 1$
2) $I, w \vDash \neg\varphi\ \ iff\ I, w\ \nvDash \varphi$
3) $I, w \vDash \varphi \rightarrow \eta\ iff\ I, w \vDash \eta$ or $I, w \vDash \varphi$
4) $I, w \vDash \Box\varphi\ iff$ for every $x\ s.t. R(w, x)$: $I, x \vDash \varphi$

$I, w \vDash \Diamond\ \varphi \Leftrightarrow$
$I, w \vDash \neg\Box\neg\varphi \Leftrightarrow$
$I, w \nvDash \Box\neg\varphi \Leftrightarrow$
$\exists x\ s.t. R(w, x)$ and $I, x \nvDash \neg\varphi \Leftrightarrow$
$\exists x\ s.t. R(w, x)$ and $I, x \vDash \varphi$

Definition

φ is **valid** iff for every $I, w, I, w \vDash \varphi$
φ is **satisfiable** iff for some $I, w\ \ I, w \vDash \varphi$
φ is **unsatisfiable** 0 W (what?)
$mod(\varphi) = \{I, w | I, w \vDash \varphi\}$

Definition

For $\Sigma \subseteq WFF\ modal\ formulae$
$\Sigma \vDash \varphi\ iff$ for every I, w s.t. $I, w \vDash \eta\ \forall\eta \in \Sigma$ then $I, w \vDash \varphi$

Lemma

If R is reflexive (i.e. $R(w, w) \forall w \in W$) Then $I, w \vDash \Box\varphi$ implies $I, w \vDash \varphi$.
If R is not reflexive then possibly $I, w \vDash \Box\varphi$ without $I, w \vDash \varphi$

Equivalence Relation

A relation is an equivalence relation iff it is

- Reflexive ($R(w, w)$)
- Transitive ($R(w, x), R(x, y) \Rightarrow R(w, y)$)
- Symmetric ($R(w, x) \Leftrightarrow R(x, w)$)

Deduction Systems

System K
Axioms 1-3 of [H]
Rule MP of [H]
Axiom K.
$\forall\varphi, \eta, \Box(\varphi \rightarrow \eta) \rightarrow (\Box\varphi \rightarrow \Box\eta)$
Rule "Necessity" or "nec"
If φ is derived without assumptions then
$< \varphi, \Box\varphi >$

Theorem

$\vdash_K \varphi$ iff $\vDash \varphi$ in every interpretation of every frame.

Question

Suppose $I, w \vDash \Box\varphi$
Does this imply that $I, w \vDash \varphi$? No

Question

Suppose $I, w \vDash \Box\varphi$. Does this imply that $I, w \vDash \Box\Box\varphi$?
If R is transitive (i.e. if whenever $R(w, x)$ and $R(x, y)$ then also $R(w, y)$) then the implication holds.

Addition of Axioms

$\Box\varphi \to \varphi$ (reflexive frames only)
$\Box\varphi \to \Box\Box\varphi$ (transitive frames only)
$\varphi \to \Box \lozenge \varphi$ (symmetric frames only)
S_5 is $K +$ the above

Theorem

$\vdash_{S_5} \varphi$ iff φ is valid in every equivalence frame.

First Order Logic

October-25-11 10:09 AM

Alphabet

An alphabet for first order logic has symbols for:

- Constants (a, b, c_{23}, 0,...)
- Functions (f, g, +,-...)
- Relations, or predicates (P, a, ...)
- Variables (x, y, ...)
- Logical connectives (¬, →,∧,∨)
- Punctuation (,), ., ','
- Quantifiers (∃,∀)
    - Existential quantifier, universal quantifier.

Structure

A structure consists of

- A domain - any non-empty set.
- Constants, functions, and relations

Arity

The number of arguments taken by a function.

Term

Value is a member of the domain

1) Each constant or variable is a term.
2) If $t_1, t_2, \dots, t_n$ are terms and f is a function with arity n then $f(t_1, t_2, \dots, t_n)$ is a term.
3) Nothing else.

Well-Formed Formula

1) If P is a predicate of arity n and $t_1, \dots, t_n$ are terms then $P(t_1, t_2, \dots, t_n)$ is a WFF.
   $(n = 0 \Rightarrow$ propositional variable)
2) If φ and η are WFFs so are
   $(\neg\varphi), (\varphi \to \eta), (\varphi \wedge \eta), (\varphi \vee \eta)$
3) If φ is a WFF and x is a variable then $\exists x. \varphi$ and $\forall x. \varphi$ are WFF.
4) Nothing else.

Lemma

Each WFF is a WFF in only one way.

Meaning

Constants, functions, predicates: obvious
Connectives as before.
$\exists x.$ means there is some element of the domain now called x $s.t.$ φ is true.
$\forall x.$ means for each element of the domain, call it x, φ is true.

Free Variables

For each WFF φ the set $FV(\varphi)$ the free variables of φ is as follows:

1) If $\varphi = P(t_1, t_2, \dots, t_n)$ where $t_1, t_2, \dots, t_n$ are terms then $F \vee (\varphi)$ is the set of variables used in $t_1, \dots, t_n$
2) If φ is $\eta \to \zeta$ then $FV(\varphi) = FV(\eta) \cup FV(\zeta)$
   (Same for ∧,∨)
   $FV(\neg\varphi) = FV(\varphi)$
3) If φ is $\exists.\eta$ or $\forall x.\eta$ then $FV(\varphi) = FV(\eta)/\{x\}$

Example Statements

3 is prime
If x is an integer, then $x \le x^2$
There is a y s.t. $y^2 = y$

Example of Structure

ℕ: domain
+: a function (arity 2)
≤: a relation (arity 2)
successor: a function of arity 1

V vertices of a graph
E edge relation

Examples of terms

$\times\left(+\left(Succ\left(Succ(0)\right), 2, \right), 0\right)$

Examples of WFF

$< \left(x, +(y, z)\right)$ is a WFF "$x < y + z$"
$\left(= \left(x, \times (x, x)\right)\right) \to \le (x, 1)$
which means $x = x \times x \ \to x \le 1$
$\exists z_1. \exists z_2. \left(\left((x > z_1) \wedge (x > z_2)\right) \wedge (x = z_1 \times z_2)\right)$
Means x is a composite number
$\forall y. \exists x. \forall z_1. \forall z_2. \left((x > y) \wedge \left(\left((x > z_1) \wedge (x > z_2)\right) \to \neg(x = z_1 \times z_2)\right)\right)$
Means for every y, there is an x greater than it such that x is does not have factors smaller than it. That is, there are infinitely many primes.

Semantics of First Order Logic

October-27-11 10:05 AM

First Order Interpretation (Structure)

A first order interpretation, or structure, is a non-empty set D and a mapping $(\square)^I$ from symbols to domain objects.

Constants → elements of D
Functions of arity n → function $D^n \to D$
Relation of arity n → relation in D^n

Valuation

A valuation is a mapping from to elements of the domain.
$\Theta: \{variables\} \to D$

For a valuation θ, a variable x and an element $a \in D$
$\Theta\left[\frac{x}{a}\right]$ is the valuation s.t.
$\Theta\left[\frac{x}{a}\right](y) = \begin{cases} a \text{ if } y = x \\ \Theta(y) \text{ otherwise} \end{cases}$

Models

For a first order interpretation I and a valuation Θ for a term t, the value of t under I and Θ is:

- $t = c\ (constant)$: $(c)^{I,\Theta} = (c)^I$
- $t = v\ (variable)$: $(c)^{I,\Theta} = (c)^\Theta$
- $t = f(t_1, \dots, t_n)$: $\left(f(t_1, \dots, t_n)\right)^{I,\Theta}$
  $= f^I\left((t_1)^{I,\theta}, \dots, (t_n)^{I,\Theta}\right)$

We say I, Θ models a formula φ denoted $I, \Theta \vDash \varphi$ as follows

1. $\varphi = R(t_1, \dots, t_n), relation$
   $I, \Theta \vDash \varphi$ iff $(t_1)^{I,\Theta}, \dots, (t_n)^{I,\Theta} \in R^I$
2. $I, \Theta \vDash \neg\varphi\ iff\ I, \Theta \nvDash \varphi$
   $I, \Theta \vDash \varphi \to \eta\ iff\ I, \Theta \nvDash \varphi\ or\ I, \Theta \vDash \eta$
   $I, \Theta \vDash \exists x. \varphi\ iff\ there\ is\ an\ a \in D\ s.t. I, \Theta\left[\frac{x}{a}\right] \vDash \varphi$
   $I, \Theta \vDash \forall x. \varphi\ iff\ for\ each\ a \in D, I, \Theta\left[\frac{x}{a}\right] \vDash \varphi$

Free Variable

x is free in a formula φ iff
φ is a term and x occurs in φ or
φ is $\neg\eta$ (or $\eta \to \zeta$) and x is free in η $(and\ \zeta)$
φ is $\exists y. \eta$ and x is free in η and x is not y
φ is $\forall y. \eta$ and x is free in η and x is not y

Free Variable Relation

If φ has free variables $x_1, \dots, x_n$ φ defines a relation
$\{(\Theta(x_1), \dots, \Theta(x_n)) : I, \Theta \vDash \varphi\}$
for a fixed I

Closed (Sentence)

φ is closed (or φ is a sentence) iff φ has no free variables.
$FV(\varphi) = \emptyset$

Definable Set

A sentence φ defines a set K of interpretations iff
$I \in K \Leftrightarrow I \vDash \varphi$

A set Σ of sentences defines K iff
$I \in K \Leftrightarrow \forall\varphi \in \Sigma, I \vDash \varphi$

If such a Σ exists that defines K then K is **definable.**

If a finite Σ exists then K is **strongly definable.**

Broad Categories

- Logical symbols (→ ¬)
- Punctuation (.,∃∀)
- Non-logical symbols
  Constants, functions, predicates
- Variables

Examples of Structure

Examples

$D = \mathbb{N}$
$0 \to zero$
$1 \to one$
$+ \to$ addition
$\times \to$ multiplication
$< \to$ less than

Another Example

$D =$ set of vertices (of a graph)
$E \to$ edge relation

Example

$D =$ strings
$C_\forall \to \forall$
$C_(\to ($
WF →well-formedness
$+ \to$ Concatenation
If $WF(x)$ then $WF(C_\neg + x)$
$WF(x) \to WF(C_\neg + x)$

Example of Modeling

Let $I = (\mathbb{N}, 0, 1, +, <) = \mathcal{N}$
$(0)^\mathcal{N} = 0 \in \mathbb{N}$
$(1)^\mathcal{N} = 1 \in \mathbb{N}$
$(<)^\mathcal{N} = \{(a, b) | a < b\}$
$(+)^\mathcal{N} = \{(a, b, c) | a + b = c\}$

Let $\varphi_1 : x < (1 + 1)$
$\mathcal{N}, \Theta \vDash \varphi\ iff\ \Theta(x) \in \{0, 1\}$
$\varphi_2 : \exists x. x < (1 + 1)$
$\mathcal{N}, \Theta \vDash \varphi_2$ for any Θ $\mathcal{N}, \Theta\left[\frac{x}{0}\right] \vDash \varphi_1$
$\varphi_3 : \forall x. x < (1 + 1)$
$\mathcal{N}, \Theta \nvDash \varphi_3$ for every Θ
$\varphi_4 : \exists x. \left(x < (1 + 1)\right) \wedge \left((1 + 1) < x\right)$
$\mathcal{N}, \Theta \nvDash \varphi$ for every Θ
φ_5: $(\exists x. x < (1 + 1)) \wedge (\exists x. (1 + 1) < x)$
$\mathcal{N}, \Theta \vDash \exists x. x < (1 + 1)$ and $\mathcal{N}, \Theta \vDash \exists x. (1 + 1 < x)$
$\therefore \mathcal{N}, \Theta \vDash \varphi_5$
$\varphi_6 : \exists x. \left(x < (1 + 1) \wedge (\exists x. (1 + 1) < x)\right)$
$\mathcal{N}, \Theta \vDash (\exists x. (1 + 1) < x)$ for each Θ
$\mathcal{N}, \Theta \vDash x < (1 + 1)$ iff $\Theta(x) \in \{0, 1\}$
$\mathcal{N}, \Theta \vDash \left(x < (1 + 1) \wedge (\exists x. (1 + 1) < x)\right)\ iff\ \Theta(x) \in \{0,1\}$
$\therefore \mathcal{N}, \Theta \vDash \varphi_6$

Example of Free Variable Relation

$I = \langle \mathbb{N}, +, \dots \rangle\ let\ \varphi : \exists x. (y = x + x)$
φ defines the set of even numbers

$I = \langle \mathbb{R}, +, \dots \rangle\ let\ \varphi : \exists x. (y = x + x)$
φ defines $\mathbb{R}$

Graph $G = (V, E)$ let $I = \langle V, E \rangle$
V a domain, E a binary relation
$\varphi : \forall y_1. \exists y_2\ E(z, y_2) \wedge E(y_2, y_1)$
φ defines the set of all vertices which have a path of length 2 to every other vertex in V.

Example of Sentence Interpretations

$\forall x. \forall y. \left(E(x, y) \to E(y, x)\right)$
Defines the set of undirected graphs.

$\exists x. \forall y. (+(x, y) = y)$
Defines identity for + operator
$\forall x. \forall y. \forall z. \left(+(+(x, y), z) = +\left(x, +(y, z)\right)\right)$
+ Operator is associative
$\forall y. \exists z. \forall x. (+(+(y, z), x) = x)$
+(y,z) acts like the identity

To make it a group, specify the identity explicitly

$\forall y.\left((+(Id, y) = y) \land (+(y, Id) = y)\right)$
$\forall x. \forall y. \forall z. \left(+(+(x, y), z) = +\left(x, \left(+(y, z)\right)\right)\right)$
$\forall y. \exists z\ (+(y, z) = id)$
This defines a group.

Hilbert Proof System

November-01-11 10:55 AM

Just get the notes here
http://www.student.cs.uwaterloo.ca/~cs245/cs-firstorder.pdf

cs-firstorder

The Hilbert Axioms for First-Order Logic

For any φ, ψ, η the following are axioms

1. $(\varphi \rightarrow (\psi \rightarrow \varphi))$
2. $((\varphi \rightarrow (\psi \rightarrow \eta)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \eta)))$
3. $(((\neg\varphi) \rightarrow (\neg\psi)) \rightarrow (\psi \rightarrow \varphi))$
4. $(\forall x.(\varphi \rightarrow \psi)) \rightarrow ((\forall x.\varphi) \rightarrow (\forall x.\psi))$
5. $(\forall x.\varphi) \rightarrow \varphi_t^x$ for any term t
6. $(\varphi \rightarrow \forall x.\varphi)$ for any variable $x \notin FV(\varphi)$

For any axiom φ and variable x not free in φ
$\forall x.\varphi$
is also an instance of the same axiom as φ

Substitution

A (syntactic) substitution of a term t for a variable x, written $(.)_t^x$ maps terms to terms and formulae to formulae as follows:

1. For a term t_1, $(t_1)_t^x$ is t_1 with each occurrence of the variable x replaced by the term t
2. If $\varphi = P(t_1, \ldots, t_{ar(P)}), then\ (\varphi)_t^x = P\left((t_1)_t^x, \ldots, (t_{ar(P)})_t^x\right)$
3. If $\varphi = (\neg\varphi), then\ (\varphi)_t^x = (\neg(\varphi)_t^x)$
4. If $\varphi = (\psi \rightarrow \eta), then\ (\varphi)_t^x = ((\varphi)_t^x \rightarrow (\eta)_t^x)$
5. If $\varphi = (\forall y.\psi)$ there are two cases
    a. If x is y then

Induction in First Order Logic

November-03-11 1:29 PM

Induction

Interpretation: $\mathbb{N}, 0, s$
s is the successor function

Axiom

$\forall x. \neg(s(x) = 0)$
$\forall x. \forall y. \left(\left(s(x) = s(y)\right) \to (x = y)\right)$
For each WFF φ with x free
$\left(\varphi(0) \wedge \left(\forall x. \left(\varphi(x) \to \varphi\left(s(x)\right)\right)\right)\right) \to \left(\forall x. \varphi(x)\right)$

In the Hilbert deduction system for FOL show that for any $\Sigma \subseteq WFF$ variable x not free in Σ
$\Sigma \vdash \varphi \Rightarrow \Sigma \vdash \forall x. \varphi$

Soundness, Completeness of FOL

November-08-11 10:04 AM

Theorem
Let Σ be a set of WFFs of FOL and φ a WFF of FOL then
$\Sigma \vdash \varphi \Leftrightarrow \Sigma \vDash \varphi$

Witnessing Property
A set of WFFs Σ has the witnessing property (aka. E-property) iff for every formula $\neg\forall x.\varphi$ in Σ there is a variable z such that
$\neg\varphi_z^x \in \Sigma$

Lemma
Let Σ be a consistent set of WFFs. Then there is a set $\Sigma', \Sigma \subseteq \Sigma'$ s.t. Σ' is consistent and Σ' has the witnessing property.

Gödel's Completeness Theorem (Gödel 1930)
The Hilbert deduction system for F.O.L. is complete.
If $\Sigma \vDash \varphi$, then $\Sigma \vdash \varphi$

Proof Outline
Soundness
Induction on the length of the proof

Completeness
1. Set up a 'witnessing' property
2. Construct a set → maximal consistent
3. Construct an interpretation satisfying a max consistent set.

1.
Proof of Lemma
Let $z_1, z_2, \ldots$ be an infinite set of variable symbols that don't occur in Σ
Consider a list $\neg\forall x_1.\alpha_1, \neg\forall x_2.\alpha_2, \ldots$ of all formulas of this form.
Inductive construction
Let $\Sigma_0 = \Sigma$
for each $i \in \mathbb{N}, \Sigma_{i+1} = \Sigma_i \cup \{(\neg\forall x_i \alpha_i) \to (\neg\alpha_i)_{z_i}^{x_i}\}$
Show by induction on i that Σ_i is consistent.
Let $\Sigma' = \bigcup_{i\geq 0} \Sigma_i$
Σ' has Witnessing Property ∎

2.
Extend Σ' to a maximal consistent set Σ^*, same as propositional case

3.
If Σ^* is any maximal consistent set then there are I, Θ s.t. $I, \Theta \vDash \Sigma^*$
Define I, Θ as follows:
Let $T = \{t' | t \text{ is a term}\}$ be the domain
Constant c: $c^I = c'$
Variable x: $x^\Theta = x'$
Function f: $f^I(t_1', t_2', \ldots, t_n') = f(t_1, \ldots, t_n)'$

Show for each $\varphi \in \Sigma^*, I, \Theta \vDash \varphi$ (Induction on $|\varphi|$)
Σ^* is a superset of Σ so
$\forall\varphi \in \Sigma, \quad I, \Theta \vDash \varphi$
So Σ consistent ⇒ Σ satisfiable

Proof of Theorem
Suppose $\Sigma \vDash \varphi$

Case 1: Σ is unsatisfiable.
Consistent ⇒ Satisfiable
So Unsatisfiable ⇒ Inconsistent
So $\Sigma \vdash \varphi$

Case 2: Σ is satisfiable
There are I, Θ such that $I, \Theta \vDash \Sigma$ and each such I, Θ also has $I, \Theta \vDash \varphi, \forall\varphi \in \Sigma$ by assumption.
∴ $\Sigma \cup \{\neg\varphi\}$ is unsatisfiable.
∴ $\Sigma \cup \{\neg\varphi\}$ is inconsistent ⇒ $\Sigma \cup \{\neg\varphi\} \vdash \varphi$ hence $\Sigma \vdash \neg\varphi \to \varphi$
Claim: $\neg\varphi \to \varphi \vdash \varphi$
Proof: Use axiom 3 with $A = B = \varphi$

∴ $\Sigma \vdash \varphi$
∎

Note
Recall the abstraction of a maximal consistent set
$$\Sigma_{i+1} = \begin{cases} \Sigma \cup \{\varphi_i\} \text{ if this is consistent} \\ \Sigma_i \text{ otherwise} \end{cases}$$
But cannot test for consistency in a finite time. Can only find it is inconsistent.

Compactness, Incompleteness

November-10-11 10:33 AM

Compactness Theorem

Suppose that Σ is unsatisfiable. Then Σ has a finite subset that is unsatisfiable.
Equivalently.
If every finite subset of Σ is satisfiable, then Σ is satisfiable.

Gödel's Incompleteness Theorem (1934)

Give any set of axioms (WFF of FOL) that suffice to define basic to arithmetic one of the following holds:

1. The set is inconsistent
2. The set is incomputable (e.g. if maximally consistent)
   i.e. There is no algorithm to list its members.
3. The axioms are incomplete
   i.e. There is a WFF φ such that neither φ nor $\neg\varphi$ is provable from these axioms.

Proof of Compactness Theorem

By soundness and completeness
Σ is satisfiable $\Leftrightarrow$ Σ is consistent
Suppose Σ unsatisfiable and hence inconsistent. for each $\varphi, \Sigma \vdash \varphi$ and $\Sigma \vdash \neg\varphi$
Let Σ_F be the set of assumptions used in these two proofs and $\Sigma_F \subseteq \Sigma, \Sigma_F$ is inconsistent and finite.

Example: Integers

Let
ζ_1 be $\forall x. 0 \neq succ(x)$
ζ_2 be $\forall x. (x \neq 0 \to \exists y. succ(y) = x)$
ζ_3 be $\forall x. \forall y. \left((succ(x) = succ(y)) \to (x = y)\right)$
For $i \in \mathbb{N}$, let

$$\varphi_i = (x \neq 0) \wedge (x \neq succ(0)) \wedge \cdots \wedge \left(x \neq succ\left(succ(\ldots(0))\right)\right), \qquad i\ times$$

Let $\Sigma = \{\zeta_1, \zeta_2, \zeta_3\} \cup \bigcup_{i \geq 0} \varphi_i$

Every finite subset of Σ is satisfiable by $\mathbb{N}$,
$0 = 0, succ(x) = x + 1$
$\theta: x = j + 1$ where $j = \max\{i:\ \varphi_i \in subset\}$
$\therefore$ Σ is satisfiable.

Now consider an interpretation modelling Σ
I: $\mathbb{N}, \infty, succ(\infty) = \infty, x \in \mathbb{N} \Rightarrow succ(x) = x + 1$

Now add the axiom
ζ_4 as $\forall x. \left(x \neq succ(x)\right)$
$I: \mathbb{N}, x_0, x_1, \ldots, x_{n-1}, \qquad succ(x_i) = x_{i \bmod n}$

Another axiom

$$\eta_i = \forall x. x \neq succ\left(succ\left(\ldots\left(succ(x)\right)\right)\right), \qquad i\ times$$

Let $\Sigma' = \Sigma \cup \bigcup_{i=0}^{\infty} \eta_i$, $\quad \Sigma'$ is satisfiable (by completeness)

$I: \mathbb{N}, x_0, x_1, \ldots, x_{n-1}, \qquad succ(x_i) = x_{i+1}$

Non-Standard Models

So Σ does not fully characterize the natural numbers.

Let Γ be any set of WFF satisfied by the natural numbers. Then
$\Gamma \cup \bigcup_{i \geq 0} \varphi_i$ is satisfiable since every finite set of $\Gamma \cup \bigcup_{i \geq 0} \varphi_i$ is satisfiable

These are called non-standard models of the natural numbers.

Resolution

November-15-11 1:07 PM

Conjunctive Normal Form (CNF)

φ is a cnf if

$$\varphi = \bigwedge_{i=1}^{n} c_i \text{ where each } c_i \text{ is } c_i = \bigvee_{j=1}^{m_i} l_{ij}$$

c_i are called **clauses**.
Each l_{ij} is either a propositional variable or a negated propositional variable.
l_{ij} are called **literals**.

Same as product of sums form.

Converting to CNF

To convert any formula to CNF:
Replace $A \to B$ by $\neg A \vee B$
$\neg(A \vee B)$ by $(\neg A \wedge \neg B)$
$\neg\neg A$ by A

Resolution Rule (for clauses)

Clause: set of literals
CNF formula: set of clauses

Resolution Rule

$$\frac{\{p\} \cup A \quad \{\neg p\} \cup B}{A \cup B}$$

Lemma

Let φ be a formula in CNF considered as a set of clauses.
Then φ is unsatisfiable if and only if there is a derivation via resolution of the clauses of φ

Resolution in First Order Logic

Quantifiers: Move to front
$\neg\exists x.\varphi$ to $\forall x.\neg\varphi$
$\neg\forall x.\varphi$ to $\exists x.\neg\varphi$
$(\exists x.\varphi) \wedge \eta$ to $\exists y.\left(\varphi_x^y \wedge \eta\right)$ where $y \notin FV(\eta)$ and $y \notin FV(\varphi)$

Yields:
$\forall x_1, \exists x_2, \ldots \varphi$ where φ has no quantifiers can then convert φ to CNF

Skolem Functions

$\forall x$ discard (Free variables implicitly universally qualified)
$\exists x$ replace by $\varphi^x_{f(y_1,\ldots,y_n)}$

Instead of $\{p, \ldots, \}$ and $\{\neg p, \ldots\}$ we get relational terms.

Suppose we have a formula $(p \vee \varphi) \wedge (\neg p \vee \eta)$ where p does not appear in φ or η
This formula is satisfiable iff $\varphi \vee \eta$ is satisfiable.

Example of CNF

$p \to (q \to r)$
$(\neg p \vee (q \to r))$
$\neg p \vee \neg q \vee r$

To prove $p \to (q \to p)$ equivalent to show $\neg(p \to (q \to p))$ is unsatisfiable
$\neg(\neg p \vee \neg q \vee p)$
$(p \wedge q \wedge \neg p)$

$$\frac{\{p\} \quad \{\neg p\}}{\emptyset}$$

Derived $\emptyset$, so derived False, so $\neg(p \to (p \to q))$ is unsatisfiable.

Proof of Lemma

$\Leftarrow$
Is "obvious"

$\Rightarrow$
Induction on the number of variables in φ. Consider φ with variable p.
Categories of clauses:

- $\{p, \ldots\}$
- $\{\neg p, \ldots\}$
- $\{\ldots\}$
- $\{p, \neg p, \ldots\}$ Discard these, they are valid.

Apply the resolution rule in all possible ways to clauses of the first and second type.
Now discard all clauses of the first and second type.

Now p appears nowhere. Do this for all variables in φ

To finish, show that if φ was unsatisfiable, the new formula is unsatisfiable.

Example of FOL Resolution

$R(x, y)$
$\neg R(z, f(z))$
Want to **unify** these two terms.

A **unifier** is a substitution θ of terms for variables that makes the terms identical.

Example

θ: $x \mapsto z, \quad y \mapsto f(z), \quad z \mapsto z$
$\theta(R(x, y)) = R(z, f(z)) = \theta\left(R(z, f(z))\right)$
Use the **Most General Unifier** (MGU)

Example

$R(x, f(y))$ and $R(g(z), z)$
$x \mapsto g(z)$
$z \mapsto f(y)$
No good since $R(g(z), f(y)), R\left(g(f(y)), f(y)\right)$

$z \mapsto f(y)$
$y \mapsto y$
$x \mapsto g(f(y))$
Works

Computations in First Order Logic

November-17-11 10:07 AM

Functions vs. Relations

Functions need equality of terms, whereas for relations the terms stay 'separate'. Do not need to modify or deal with terms as non-atom when using relations.

Recall: Natural Numbers

Constant symbol: 0
Unitary function symbol: S

$\forall x. S(x) \neq 0$
$\forall x. \forall y. \left(\left(S(x) = S(y)\right) \rightarrow (x = y) \right)$
$\left(\varphi(0) \wedge \left(\forall x. \left(\varphi(x) \rightarrow \varphi(x+1)\right)\right)\right) \rightarrow \left(\forall x. \varphi(x)\right)$

Add a function '+'
$\forall x. x + 0 = x$
$\forall x. \forall y. x + S(y) = S(x + y)$

Add a function $\times$
$\forall x. x \times 0 = 0$
$\forall x. x \times S(y) = (x \times y) + x$

Using relations instead of functions:
$\text{Plus}(x, y, z) = \{(a, b, c) | a + b = c\}$
$\forall x. \text{Plus}(x, 0, x)$
$\forall x. \text{Plus}(x, y, z) \rightarrow Plus\left(x, S(y), S(z)\right)$
"Plus represents a function"
$\forall x. \forall y. \forall z. \forall w. \left(\text{Plus}(x, y, z) \wedge \text{Plus}(x, y, w)\right) \rightarrow (z = w)$

Lists as Domain Elements

Constant symbol: $\emptyset$
Binary function symbol: Cons

$\forall x. \forall y. Cons(x, y) \neq 0$
$\forall x. \forall y. \forall z. \forall q. \left(Cons(x, y) = Cons(z, w)\right) \rightarrow \left((x = z) \wedge (y = w)\right)$

$\forall x. \forall y. (First(x, y) \rightarrow \left(\exists z. x = Cons(y, z)\right)$
$\forall x. Append(\emptyset, x, x)$
Type equation here.

Lists are built out of cons as
$[1\ 2] = cons\left(1, cons(2, \emptyset)\right)$
$f_{Append}(cons(a, b), y) = cons\left(a, f_{Append}(b, y)\right)$ as a function
$\forall x. \forall y. \forall z. \forall w. (Append(x, y, z) \rightarrow Append\left(Cons(w, x), y, Cons(w, z)\right)$

Example

Can we prove $Append([a\ b], [c\ d], [a\ b\ c\ d])$?

$Append(Cons\left(a, Cons(b, \emptyset)\right), Cons\left(c, Cons(d, \emptyset)\right), Cons\left(a, Cons\left(b, Cons\left(c, Cons(d, \emptyset)\right)\right)\right)$
$\leftarrow Append\left(Cons(b, \emptyset), Cons\left(c, Cons(d, \emptyset)\right), Cons\left(b, Cons\left(c, Cons(d, \emptyset)\right)\right)\right)$
$\leftarrow Append\left(\emptyset, Cons\left(c, Cons(d, \emptyset)\right), Cons\left(c, Cons(d, \emptyset)\right)\right)$, $[Axiom]$

Resolution

Refute
$\neg Append(Cons\left(a, Cons(b, \emptyset)\right), Cons\left(c, Cons(d, \emptyset)\right), Cons\left(a, Cons\left(b, Cons\left(c, Cons(d, \emptyset)\right)\right)\right)$

CNF of rules:
1: $Append(\emptyset, x, x)$
2: $\neg Append(x, y, z) \vee Append\left(Cons(w, x), y, Cons(w, z)\right)$

Need to unify terms
$\neg Append([a\ b], [c\ d], [a\ b\ c\ d])$
$\neg Append([b], [c\ d], [b\ c\ d] \vee Append([a\ b], [c\ d], [a\ b\ c\ d]$
$\Rightarrow \neg Append([b], [c\ d], [b\ c\ d])$
$\neg Append(\emptyset, [c\ d], [c\ d]) \vee Append([b], [c\ d], [b\ c\ d])$
$\Rightarrow \neg Append(\emptyset, [c\ d], [c\ d])$
$Append(\emptyset, [c\ d], [c\ d])$
$\emptyset$

Impossible Computations

November-22-11 10:16 AM

Countable

A set S is countable if there is a bijection between S and ℕ.

Halting Problem

Program: P
Input: I
Can we inspect P and I to determine whether P with input I will halt?

Want a function 'halts?' such that (halts? P I) returns true if (P I) halts and false if (P I) does not.
Note that 'halts?' must always halt.

There does not exists such a program.

Decidability

A **decision problem** is one which asks for an answer 'yes' or 'no' to each input. Each input has only one correct answer.

Equivalently,
A set of possible inputs, which is $\{I \mid I$ has answer yes$\}$

A decision problem is **decidable** iff there is a program that for any input, gives the correct answer in a finite number of steps.

Church's Thesis

Every programming method is equivalent to (or weaker than) a Turing machine.

Decidability

A decision problem (i.e. a set) D is decidable iff ∃ a program P such that $\forall I, P(I) = \begin{cases} true\ if\ I \in D \\ false\ if\ I \notin D \end{cases}$

Acceptable, Semi-Decidable

A decision problem D is acceptable or semi-decidable iff $\exists p$ such that $\forall I, P$ on I halts if $I \in D$, loops if $I \notin D$

Claim

For each D,
D is decidable iff both D is semi-decidable and $\bar{D}$ is semi-decidable
$\bar{D} = \{I \mid I \notin D\}$

Proof

Exercise

Note

The halting problem is semi-decidable but not decidable.

Looping Problem

Given P and I, does P on input I loop forever?
As a set, this is $\{P, I \mid P\ on\ I$ loops forever$\}$

Claim

The Looping Problem is undecidable.

Empty-Halt

Let Empty-Halt be the problem: Given P, does P halt with empty input?

Claim

Empty-Halt is undecidable.

Hilbert's Tenth Problem

Given a polynomial $p(x_1, \ldots, x_n)$ in n variables with integer coefficients, are there rational numbers $r_1, \ldots, r_n$ such that $p(r_1, r_2, \ldots, r_n) = 0$?

This problem is undecidable.

Theorem: Valid is Undecidable

Valid $= \{\varphi \in WFF \mid \varphi$ is valid$\}$

Note: The analog for propositional formulas is decidable, just try all possible valuations.

Enumerable

Halting Problem

Suppose we have the function 'halts?'

Consider the function

```
(define (self-halts? P)
     (halts? P P))
```

What happens with

```
(self-halts? self-halts?)
⇒(halts? self-halts? self-halts?)
⇒ True
```

Now consider

```
(define (halt-if-dont P)
     (cond   [(halts? P P) (loop)]
               [else True]))

(halt-if-dont halt-if-dont)
⇒ (cond [(halts? halt-if-dont halt-if-dont) (loop)]
          [else True])
```

$\Rightarrow \begin{cases} \text{(loop) if (halts? halt-if-dont halt-if-dont)} \\ \text{True otherwise} \end{cases}$

So it halts only if (halts?) says it does not. Therefore the halts function fails on this function.

Diagonalization

1. The power set of ℕ is uncountable.
   That is, if $f: \mathbb{N} \to P(\mathbb{N})$, then $\exists T \in P(\mathbb{N})$ s.t. $\forall n \in N, f(n) \neq T$
2. The halting problem is undecidable.

Another way to show halting problem is undecidable

	List of Inputs
List of Programs	0 1 0 0 1 0 ... 1 0 0 1 1 1 ... 1 1 1 1 1 1 ... ⋮

1: Halts, 0: Loops
Take complement of diagonal. No programs halts on P_i iff P_i loops on P_i

Reduction

Suppose we have a program A which uses program B. And if B returns a result then A will halt.

If B is decidable, then A is decidable.
If A is undecidable, then B is undecidable.

Proof of Looping Undecidability

Reduce the halting problem to the looping problem.
halts? is not loops?
∴ halting undecidable ⇒ looping undecidable.

Proof of Empty-Halt Undecidability

Let E be the program that satisfies

$E(P) = \begin{cases} 1\ if\ P\ halts\ on\ \emptyset \\ 0\ if\ P\ loops\ on\ \emptyset \end{cases}$

```
(halts? P I)
     (E (lambda (Q) (P I))))
```

Therefore Empty-Halt is undecidable

Validity Undecidability Proof Sketch

Valid $= \{\varphi \in WFF \mid \varphi$ is valid$\}$

Can't try all valuations/interpretations because there are infinitely many interpretations.

Recall Empty-Halt = {P | P halts on empty input} is undecidable. Will show if Valid is decidable, then Empty-Halt is decidable.

Plan: given P, construct φ_P s.t. $\varphi_P \in$ Valid iff $P \in$ Empty-Halt

List structures: function cons, constant ∅ , define \lambda, *cond*
Take the formulas for cons
$\forall x \forall y \neg cons(x, y) = \emptyset$
$\forall x \forall y \neg cons(x, y) = define$
⋮
$\forall x \forall y \forall x' \forall y' \big(cons(x, y) = cons(x', y')\big) \to \big((x = x') \wedge (y = y')\big)$
Now P is a term over lists (with names being constant terms)
List of pairs $cons(x, d)$ "x has definition d"
Now create a formula $subs(s, D, y, E)$ meaning list x, in context D, after 1 substitution step, produces list y in context E. The context is the dictionary.

Note: The analog for propositional formulas is decidable, just try all possible valuations.

Enumerable
S is enumerable iff ∃ an algorithm A s.t. A outputs a list of items $a_1, a_2, \ldots$ such that A outputs b iff $b \in S$.

Lemma
S is enumerable iff S is semi-decidable.

Proof
Exercise

Theorem
There is a set Σ of satisfiable WFFs s.t.
1) Σ is decidable
2) For each set Σ' of WFF if $\Sigma \cup \Sigma'$ is satisfiable (i.e. consistent) and is decidable, or semi-decidable s.t. neither $\Sigma \cup \Sigma' \vdash \varphi$ nor $\Sigma \cup \Sigma' \vdash \neg\varphi$

Gödel's Incompleteness Theorem
We can take Σ to be the rules of arithmetic.

Proof: "Times" sufficient to implement "cons".

Now P is a term over lists (with names being constant terms)
List of pairs $cons(x, d)$ "x has definition d"
Now create a formula $subs(s, D, y, E)$ meaning list x, in context D, after 1 substitution step, produces list y in context E. The context is the dictionary.

Properties of subs
- Constant values don't substitute
  - $\forall x \forall y \forall D \forall E.Value(x) \rightarrow \neg Subs(x, D, y, E)$
  - Where $Value = \{x | x \text{ is constant}\}$
- A defined name gets replaced by its definition
  - Let Lookup be a formula s.t. $Lookup(x, D, y)$ iff "x has definition y in D"
  - $\forall x. \forall y. \forall D. Lookup(x, D, y) \rightarrow subs(x, D, y, D)$
  - Look up:
    - $\forall x. \forall y. \neg Lookup(x, \emptyset, y)$
    - $\forall x. \forall y. \Big(Lookup(x, [cons(x, y), \ldots], y)$ $\wedge \Big(\forall z. \big(z \neq y \rightarrow \neg Lookup(x, [cons(x, y), \ldots], z)\big)\Big)\Big)$
    - $\forall x. \forall x'. \forall y. \forall y'. \Big(\big(Lookup\ (x, D, y) \wedge (x' \neq x)\big)$ $\rightarrow Lookup(x, cons(cons(x', y'), D), y)\Big)$
    - Handle defines
    - Etc..

Evaluation:
Some chain of substitutions to x produces y
- $\forall x. \forall D. \big(Value(x) \rightarrow Eval(x, D, x)\big)$
- $\forall x. \forall y. \forall z. \forall D. \forall E\ \Big(\big(Subs(x, D, y, E) \wedge Eval(y, E, z)\big) \rightarrow Eval(x, D, z)\Big)$

The conjunction of all these formulas, and the formula $\exists x\ Eval(P, \emptyset, x)$ is valid iff P halts on empty input.

Therefore, validity of formulas is undecidable.

Proof of Theorem
Take Σ to be the formulas of the previous proof describing computation.

If $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$ and an algorithm can decide Γ then an algorithm can find whether $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$

Consider the formula: this algorithm halts/

Review

December-01-11 10:09 AM

Propositional Logic

- Precise definition of well formed formulae.
- Unambiguous syntax.
- Semantics from syntax
  - Based on valuations
  - Valid, satisfiable, unsatisfiable
  - Equivalence, entailment
- Variations on syntax
  - Adequate set of connectives
  - Normal forms
- Syntactic approach: proofs
  - Strict formal proofs
  - Correctness of a proof easily checked step by step
- Key properties
  - Soundness
  - Completeness

Further Study: Weaker proof systems

- Constructive logic
  - To prove 'P or Q' either prove P or prove Q
  - But then can't prove $\vdash P \vee \neg P$ for arbitrary P.
- Linear logic
  - Constrain the number of uses of a formula for implication
  - Length of a proof is bounded by the number of uses of axioms and hypotheses.

Modal Logic

- Consider many valuations simultaneously
  - possibility, necessity, etc.
- Syntax: $\square, \lozenge$
- Semantics: set of related valuations.

Further Study

- AI, planning
- Systems with time, time dependent behaviour
- Formal software engineering

First - Order (Predicate) Logic

- 'Things' and their properties
  - Interpretations: what are the actual things/relations/etc.
  - Valuations: Current meaning of variables.
- Ideas from propositional logic still work
- Syntactic additions: qualifiers, functions
- Soundness, Completeness

Related Notions

- (Abstract) Data types $\Leftrightarrow$ specified functions, relations, and constants
- Terms (of functions) as domain objects(things)
  - Interpreter, compiler
  - Programming languages with powerful type systems

Further study

- How complex must a proof be?

Decidability

- "This statement is false"
- "This statement has no proof"
- "Statement P has no proof" is statement P