COMP0009 Revision Notes

Formal Logic : Syntax, Semantics, proof system (Inference)
how to write / what it means / deduction. proofs. (how to reason)

Propositional Logic.

prop ::= p | q | r |

fm ::= prop | ¬fm | fm ∘ fm

an operand ∧, ∨ or →

Literal: prop or its negation p / ¬p

Main Connective: $(p \wedge q) \vee \neg(q \to r)$

the connective with the largest scope (most outside, highest level)

Semantics

v → evaluation. ⊤ truth ⊥ false.

$v(\neg\phi) = \top \iff v(\phi) = \bot$

$v(\phi \wedge \psi) = \top \iff v(\phi) = v(\psi) = \top$.

$v(\phi \vee \psi) = \top \iff v(\phi) = \top$ or $v(\psi) = \top$.

$v(\phi \to \psi) = \top \iff v(\neg\phi) = \top$ or $v(\psi) = \top$

Validity, Satisfiability, Equivalence.

ϕ is valid if $v(\phi) = \top$ for all types of valuations v. (always true)

ϕ is satisfiable if $\exists v$ $v(\phi) = \top$. (true at least once)

ϕ and ψ is logically eq. iff $\forall v$ $v(\phi) = v(\psi) \Rightarrow \phi \equiv \psi$

All valid fmla are satisfiable

Predicate Logic

Language $L(C, F, P)$

C constant symbols.

F function symbols f^n (n-ary)

P a nonempty predicate (relation) symbol set P^n (n-ary)

$tm ::= v : v \in Var \mid c : c \in C \mid f^n(tm \ldots tm) : f^n \in F$.

$3 + (x \times 2) \iff +(3, \times(x, 2))$.

$2, 3 \in C$ $x \in Var.$

$+, \times \in f^2.$

atom $:= p^n(tm, \ldots, tm)$: $p^n \in P.$

$x+y < 2 \times y - 1$ $<$ is a p^2

$fm ::=$ atom $| \neg fm | (fm \vee fm) | \exists v fm : v \in Var.$

$(fm \wedge fm)$

$(fm \to fm)$

$\forall x \phi$ as abbr. $\neg \exists x \neg \phi$ $\forall x \phi \Leftrightarrow \neg \exists x \neg \phi.$

L-structure.

(D, I) D is any non-empty set (domain)

I constants, funcs, predicates

I_C maps constant symbols in C to elements of D

I_f — — n-ary funcs $f \in F$ to n-ary functions over D

I_P — — n-ary predicate symbols $p \in P$ to n-ary relations over D (subset of D^n)

interpretation

domain

$S = [D, I]$

structure

A: $Var \to D$ is a variable assignment.

e.g. $I_P(=) = \{(d, d) : d \in D\}.$

$[c]^{S,A} = I(c).$

$[x]^{S,A} = A(x)$

$[f(\tau_0 \cdots \tau_{n-1})]^{S,A} = I(f)([\tau_0]^{S,A} \ldots [\tau_{n-1}]^{S,A})$

REP. $S, A \models R(\tau_0 \cdots \tau_{n-1}) \Leftrightarrow ([\tau_0]^{S,A} \ldots [\tau_{n-1}]^{S,A}) \in I(R)$

$S, A \models \neg \phi \Leftrightarrow S, A \not\models \phi.$

$S, A \models (\phi \vee \phi') \Leftrightarrow S, A \models \phi$ or $S, A \models \phi'$

$S, A \models \exists x \phi \Leftrightarrow S, A[x \mapsto d] \models \phi$ for some $d \in D.$

Validity

$S = (D, I)$ be a L-structure, ϕ a formula

ϕ is valid in S. for all A: $Var \to D$ we have $S, A \models \phi$

$S \models \phi.$

ϕ is valid. for all L-structures S we have $S \models \phi.$

$\models \phi.$

Satisfiability.

ϕ is satisfiable in S if $\exists$ A: $Var \to D.$ such that $S, A \models \phi.$

ϕ is satisfiable if $\exists A, S$ $S, A \models \phi$.

ϕ is not valid iff $\neg\phi$ is satisfiable.

Propositional Proof System.

a system for determining validity of fmla.

Obvious one: write down truth table for ϕ.

problem - exponential tie.

$\phi: p_1 \vee p_2 \cdots \vee p_{50}$

2^{50} possibilities.

Better: manipulate and analyse the syntax of formula to see if anything can falsify it.

problem: how to make sure syntactical changes $\Updownarrow$ make semantic sense?

make sure the proof ~~system~~ is sound and complete.

$\models \phi \Leftrightarrow \phi$ is valid.

$\vdash \phi \Leftrightarrow$ there is a proof of ϕ

Soundness $\vdash \phi \Rightarrow \models \phi$ System can prove only valid things

completeness $\models \phi \Rightarrow \vdash \phi$. if sth. is valid. the system can prove it.

$\models \phi \Leftrightarrow \vdash \phi$

Axiomatic Proof System.

Fix a prop. lang with only $\rightarrow$ and $\neg$. (no double $\neg$).

I. $p \rightarrow (q \rightarrow p))$

II. $(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$.

$\Updownarrow$

$\neg p \wedge (q \rightarrow r)$

$\Updownarrow$

$\neg p \wedge (\neg q \wedge r)$

$\Updownarrow$

$$\Updownarrow$$

$$(\neg p \wedge \neg q \wedge r) \rightarrow ((\neg p \wedge q) \wedge (\neg p \wedge r))$$

$$\Updownarrow$$

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

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

$$\Updownarrow$$

true.

III. $(\neg p \rightarrow \neg q) \rightarrow (q \rightarrow p)$.

$$\neg(p \wedge \neg q) \wedge (\neg q \wedge p)$$

$$(\neg p \vee q) \wedge (\neg q) \wedge p$$

$$\Updownarrow$$

true.

Inference Rule.

Modus Ponens. if proved ϕ and $\phi \rightarrow \psi$, then deduce ψ.

$$\frac{\phi \qquad (\phi \rightarrow \psi)}{\psi}$$

Modus tollens. if proved $\neg q$ and $p \rightarrow q$, then deduce $\neg p$.

$$\frac{\neg q. \qquad (p \rightarrow q)}{\neg p}$$

Proof.

a proof is a seq. of fmlas.

$$\phi_0 \cdots \phi_n.$$

s.t. for $i \leq n$. ϕ_i is either
* axiom
* obtained by modus ponens from ϕ_j ϕ_k $(j, k < i)$

e.g. $\boxed{\begin{array}{l} \phi_j. \\ \phi_k = \phi_j \rightarrow \phi_i \end{array}}$

$$\Downarrow$$

$$\phi_i$$

If such a proof exists, ϕ_n is a theorem.

$\underset{\sim}{\vdash \phi_n}$.

Ex. $\vdash (p \to p)$

AxII. $(p \to ((p \to p) \to p)) \to ((p \to (p \to p)) \to (p \to p))$

AxI. $(p \to ((p \to p) \to p))$

$\therefore$ $(p \to (p \to p)) \to (p \to p)$

Ax I. $(p \to (p \to p))$.

$\therefore$ $\vdash p \to p$.

Proofs with other conn

IV. $p \to \neg\neg p$ and $\neg\neg p \to p$

V $(p \vee q) \Leftrightarrow (\neg p \to q)$. $\Leftrightarrow$ $p \vee q$

VI $(p \wedge q) \Leftrightarrow \neg(p \to \neg q)$ | $\to \neg(\neg p \vee \neg q)$

$\Leftrightarrow p \wedge q$.

Proofs with assumptions

$T \vdash \phi$

if there's a proof of ϕ using assumptions for T.

$\phi_0 \dots \phi_n$.

ϕ_i is either * axiom.

* an assumption.

* obtained from $\phi_j\ \phi_k$.

$(j, k < i)$

Ex. $\bot \vdash p$

$\{\neg(q \to q)\} \vdash p$

$\therefore$ $q \to q$.

$\therefore$ $(q \to q) \to \neg\neg(q \to q)$ Ax.IV.

$\therefore$ $\neg\neg(q \to q)$. MP.

$\because\ \neg\neg(q\to q)\to(\neg p\to\neg\neg(q\to q))$
$\therefore\ \neg p\to\neg\neg(q\to q)$
$\because\ (\neg p\to\neg\neg(q\to q))\to(\neg(q\to q)\to p)$ Ax III.
$\therefore\ \neg(q\to q)\to p$
$\because\ \neg(q\to q)$ Assum.
$\therefore\ p$

$\because\ \neg\neg(q\to q)\to(p\to\neg\neg(q\to q))$ Ax. I.
$\therefore\ p\to\neg\neg(q\to q)$
$\because\ (p\to\neg\neg(q\to q))\to(\neg(q\to q)\to\neg p)$ Ax III.
$\therefore\ \neg(q\to q)\to\neg p$
$\because\ \neg(q\to q)$ Assumption.
$\therefore\ \neg p$

Soundness ✓.

check all axioms

check if ϕ and $\phi\to\psi$. then ψ is valid. Modus Ponens

$\Downarrow$

all provable fmlas are valid.

$\vdash\phi\Rightarrow\models\phi$.

Completeness.

$\models\phi\Rightarrow\vdash\phi$. (all. valid fmla are provable)

Propositional Tableaux.

given fmla ϕ, tableau can tell us whether it is satisfiable or not.

* decomposing fmla according to certain rules to the point only literals are left.

$\Downarrow$

atomic fmla. or its negation.

* tableau. (closed or not).

if tableau closes ϕ is unsatisfiable

tableau never closes ϕ is satisfiable

$((\phi\wedge\neg\phi)\vee(\psi\wedge\neg\psi))$

$(\phi\wedge\neg\phi)$ $(\psi\wedge\neg\psi)$

ϕ ψ

$\neg\phi$ $\qquad\qquad$ $\neg\psi$

$\oplus$ $\qquad\qquad$ $\oplus$

closes both $\to$ not satisfiable.

Tableau T is a BT. (node labelled by fmla)

Every fmla in tab, except literals.
gets expanded. and ticked.

if branch T_B contains p and $\neg p$.
it's closed.

if every branch of T is closed then
T is closed.

α formula. $\phi \wedge \psi$.

$\phi \wedge \psi$ is true iff ϕ and ψ is true.

$$\begin{array}{c} \phi\wedge\psi \\ | \\ \phi \\ | \\ \psi . \end{array}$$

$$\begin{array}{c} \neg\neg\phi \\ | \\ \phi \end{array} \qquad \begin{array}{c} \neg(\phi\vee\psi) \\ | \\ \neg\phi \\ | \\ \neg\psi \end{array} \qquad \begin{array}{c} \neg(\phi\to\psi) \\ | \\ \phi \\ | \\ \neg\psi . \end{array}$$

β formula. $\phi \vee \psi$

$\phi \vee \psi$ is true iff ϕ or ψ is true.

$$\begin{array}{ccc} & \phi\vee\psi & \\ / & & \backslash \\ \phi & & \psi . \end{array}$$

$$\begin{array}{ccc} & \neg(\phi\wedge\psi) & \\ / & & \backslash \\ \neg\phi & & \neg\psi . \end{array} \qquad \begin{array}{ccc} & \phi\to\psi & \\ / & & \backslash \\ \neg\phi & & \psi . \end{array}$$

Example. $p \vee r$ $\qquad$ β.

$\neg q$ $\qquad$ X

$\neg((p \vee q) \to r)$ $\qquad$ α.

$\neg(p \vee q$ ...

$((p \to r) \vee (q \to r))$. β.

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

$\neg(p \to r)$ α.

$q \to r$ β.

$\neg r$ $\times$

$\neg(p \wedge q)$ β.

$\neg((p \to r) \vee (q \to r))$ α.

$p \vee q$. β.

$\neg(((p \wedge q) \to r) \to ((p \to r) \vee (q \to r)))$ α.

p $\times$

$p \to r$ β.

$\neg(q \to r)$. α

$\neg(((p \wedge q) \to r)$. α.

* a tableau is complete is either ticked (expanded) or is a literal.
* if ϕ is at the root of a complete open tableau, ϕ is satisfiable.

ϕ is satisfiable. $\Leftrightarrow$ $\neg\phi$ is not valid.

ϕ is valid $\Leftrightarrow$ $\neg\phi$ is not satisfiable.

test ϕ is valid $\Leftrightarrow$ test $\neg\phi$ is not satisfiable

Ex.1. is $((p \wedge q) \to r) \to ((p \to r) \vee (q \to r)))$ valid?

is $\neg(((p \wedge q) \to r) \to ((p \to r) \vee (q \to r)))$ not satisfiable? α.

$((p \wedge q) \to r)$ ✓

$\neg((p\to r)\vee(q\to r))$ ✓

- $\neg(p\wedge q)$ ✓
  - $\neg(p\to r)$ ✓
  - $\neg(q\to r)$ ✓
    - $\neg p$
      - p
      - $\neg r$
      - $\otimes$
    - $\neg q$
      - p
      - $\neg r$
      - q
      - $\neg r$
      - $\otimes$
- r
  - $\neg(p\to r)$ ✓
  - $\neg(q\to r)$ ✓
  - p
  - $\neg r$
  - q
  - $\neg r$
  - $\otimes$

$\therefore\ (((p\wedge q)\to r)\to((p\to r)\vee(q\to r)))$ is valid.

Ex 2. Is $(((p\to r)\vee(q\to r))\to((p\vee q)\to r))$ valid.

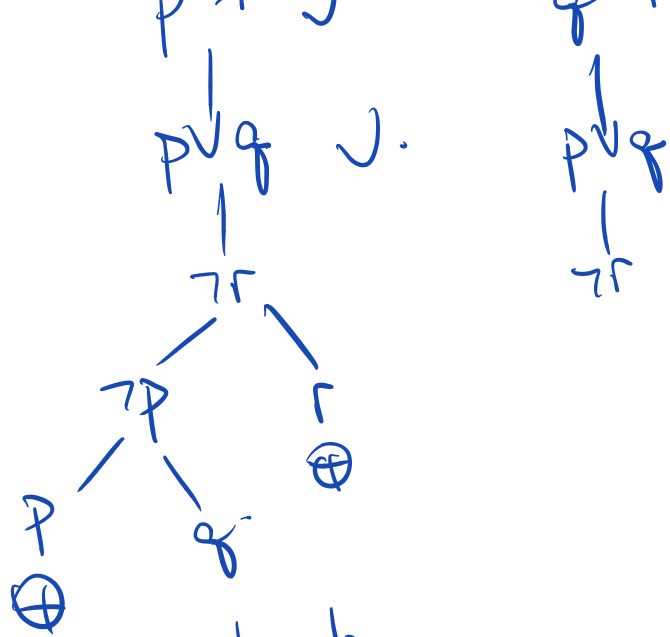
$\Leftrightarrow$ is $\neg(((p\to r)\vee(q\to r))\to((p\vee q)\to r))$ ✓ satisfiable.

$((p\to r)\vee(q\to r))$ ✓ α.

$\neg((p\vee q)\to r)$ ✓

β

- $p\to r$
- $q\to r$



open branch

$\therefore$ $((p\to r)\vee(q\to r))\to((p\vee q)\to r))$ is not valid.

DNF

Disjunctive Normal form.

disjunctions of conjunction literals.

$$(p\wedge\neg q\wedge r)\vee(p\wedge q)\ \cdots\vee\cdots(x\wedge x\wedge x)$$

Each. fmla. has an equivalent fmla in DNF.

Conjunctive Normal. Form.

$$(p\vee q)\wedge(\neg p\vee\neg r)\wedge q\cdots\wedge\cdots\wedge(x\vee x\vee x)$$

Converting to DNF.

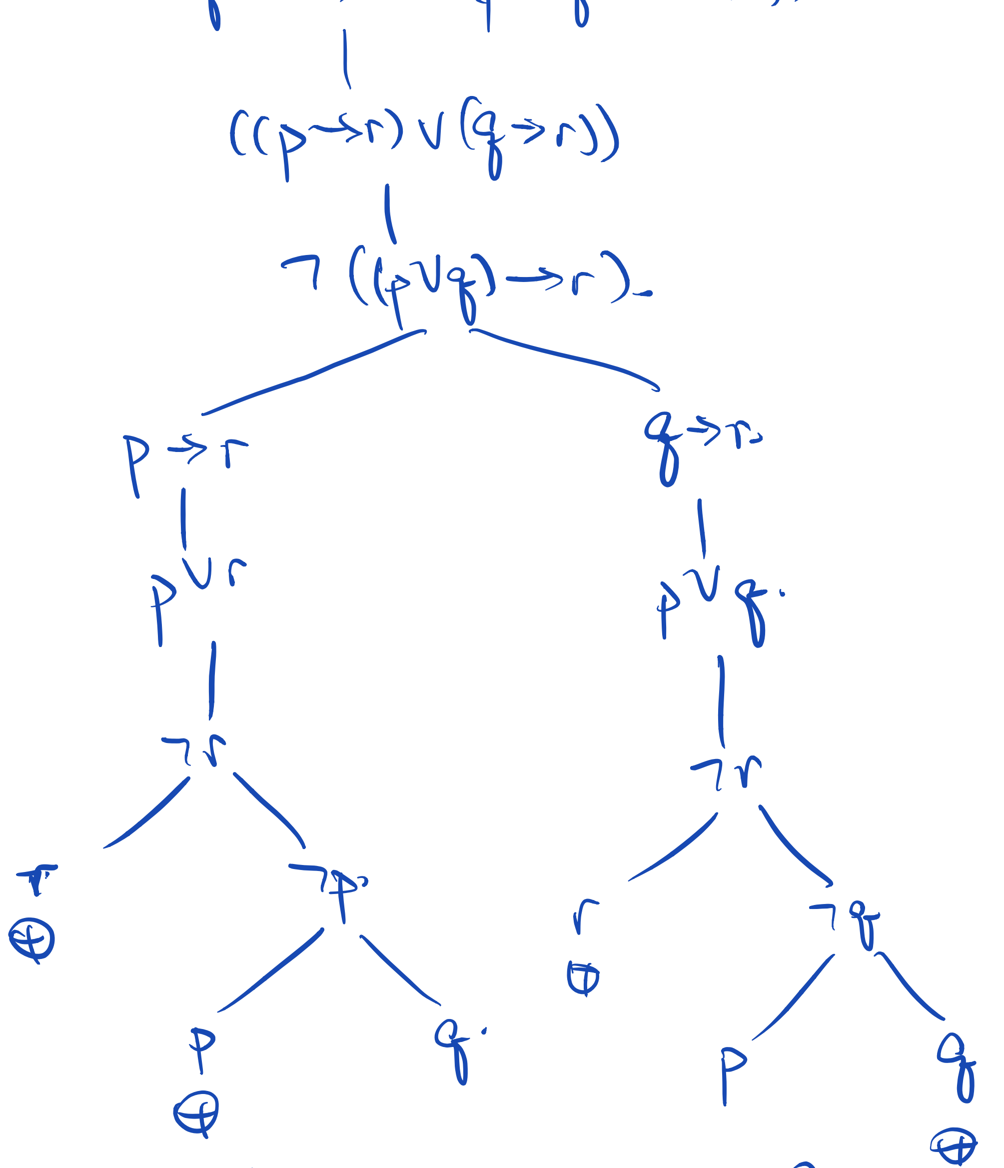
truth table / logical equivalences / tableau

- φ at root of T. T is completed.
- Expand until
- open branch θ of T let.

$$C_\theta=\bigwedge\{\text{literal in } \theta\}.$$

then $\varphi\equiv\bigvee C_\theta$

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



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

$$(\neg p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge \neg r).$$

First Order Tableaus

- A literal is an atom or its negation.
- closed term is a term contains no vars.
- Same kind of tableau construction.

α, β remain the same.

δ formula.

choose a const. p.

add fmla at each leaf below the node.

$$\exists x \phi \qquad \neg \forall x \phi$$
$$\phi(p/x). \qquad \neg \phi(p/x)$$
$$\underbrace{\exists x \neg \phi \equiv \neg \forall x \phi.}$$

γ formula

pick any closed term t.

don't tick node.

until you used every available const/closed term

$$\forall x \phi. \qquad \neg \exists x \phi$$
$$\phi(t/x). \qquad \neg \phi(t/x).$$
$$\underbrace{\neg \exists x \phi \equiv \forall x \neg \phi.}$$

$\forall x \neg p(x).$ γ

$H(a) \to F(a)$ β.

~~$\neg\neg \exists y\ p(y).$ δ~~ (double negation first) α rule

$\neg(\forall x \neg p(x) \to \neg \exists y\ p(y))$ α.

$\neg(\forall x \neg q(x) \vee \exists x \forall y \neg(x < y))$ α.

$G(a) \to H(a).$ β

$\neg\neg(\forall x(G(x) \to H(x)) \wedge \forall x(H(x) \to F(x)) \wedge G(a) \wedge \neg \exists x(G(x) \wedge F(x))).$ α.

$\neg \exists x(G(x) \wedge F(x))$ γ

$\neg \forall y \neg (c < y)$ δ.

$\forall x(H(x) \to F(x))$ γ.

$G(a)$ X.

$\neg p(c)$ X.

$\forall x(G(x) \to H(x))$ γ.

$\neg H(a).$ X.

$\forall x(G(x) \to H(x)) \wedge \forall x(H(x) \to F(x)) \wedge G(a) \wedge \neg \exists x(G(x) \wedge F(x))$ α.

$\neg G(a).$ X

$\neg(G(a) \land F(a))$ β.

$\exists y\, p(y)$. 8.

Ex. 3.

$(\forall x \neg p(x) \to \neg \exists y\, p(y))$ valid?

$\neg(\forall x \neg p(x) \to \neg \exists y\, p(y))$

|

$\forall x \neg p(x)$ ✓

|

$\exists y\, p(y)$. ✓

|

$p(c)$.

|

$\neg p(c)$.

|

$\otimes$

Ex4. $\neg(\forall x(Gx \to Hx) \land \forall x(Hx \to Fx) \land Ga \land \neg \exists x(Gx \land Fx)$ valid ?

$\forall x(Gx \to Hx) \land \forall x(Hx \to Fx) \land Ga \land \neg \exists x(Gx \land Fx)$.

|

$\forall x(Gx \to Hx)$

|

$\forall x(Hx \to Fx)$

|

Ga.

|

$\neg \exists x(Gx \land Fx)$

|

$\neg(Ga \land Fa)$ ✓.

$\neg Ga$ $\quad$ $\neg Fa$

$\oplus$

$\neg Fa$ branches to $\neg Ha$ $\quad$ $Fa.$

$Fa.$ $\oplus$

$\neg Ha$

$Ga \to Ha$

$\neg Ga$ $\quad$ $Ha.$

$\oplus$ $\quad$ $\oplus$

$\therefore \neg(\forall x(Gx \to Hx) \land \forall x(Hx \to Fx) \land Ga \land \neg\exists x(Gx \land Fx))$ is valid

Infinite Tableau

Ex.5. $\neg(\forall x \neg q(x) \lor \exists x \forall y \neg(x<y))$

$\neg\forall x \neg q(x)$ ✓

$\neg\exists x \forall y \neg(x<y).$

$\neg\neg q(c)$ ✓

$q(c)$

$\neg\forall y \neg(c<y)$ ✓

$\neg\neg(c<d).$ ✓

for infinite tableau it's satisfiable:

$D = \{c, d, e \ldots\}$

$I(q) = \{c\}.$

$I(<) = \{(c,d),(d,e) \ldots\}.$

$c < d$

|

$\neg \forall y \neg (d < y)$ ✓

|

$\neg\neg(d < e)$ ✓

|

$d < e$

|

⋮

Alternative.

Tableau as lists.

Recall literal $p, \neg p$

α $\quad \phi_1 \wedge \phi_2 \quad \neg(\phi_1 \vee \phi_2) \quad \neg(\phi_1 \to \phi_2), \neg\neg\phi$

β $\quad \phi_1 \wedge \phi_2, \neg(\phi_1 \wedge \phi_2) \quad \phi_1 \to \phi_2$

$Tab = [\{\phi\}]$.

while ! Tab.empty() {

branch (set). — Σ = Tab.dequeue();

if Σ is fully expanded and Σ doesn't have contradiction,

SATISFIABLE

else.

pick non-literal $\psi \in \Sigma$.

switch (ψ)

case α:

$\Sigma = \Sigma[\alpha / \{\alpha_1, \alpha_2\}]$

make an α expansion on ψ

$\Sigma[x/y]$, delete x, put y in.

if Σ doesn't have contradictory and $\Sigma \notin$ Tab. enqueue Σ.

Case β.

$\Sigma_1 = \Sigma(\beta / \beta_1)$

if —— . . . enq Σ_1

$\Sigma_2 = \Sigma(\beta / \beta_2)$

—— —— Σ_2.

Output unsatisfiable (after all Σ sets).

if Predicate Tableaux,

switch (ψ)

case $\delta = \exists x \theta x$.

$\Sigma = \Sigma[\exists x \theta x / \theta(p)]$ new const p

case $\delta = \neg \forall x \theta x$

$\Sigma = \Sigma[\neg \forall x \theta x / \neg \theta(p)]$ —— p

Case $\delta = \forall x \theta(x)$.

$\Sigma = \Sigma \cup (\theta(t))$ t for closed term

—— —— . $\neg \exists x \theta(x)$

$\Sigma = \Sigma \cup \Sigma \cup (\theta(t))$ —— . . .

If $\Sigma \notin$ Tab and not $C(\Sigma)$ enqueue Σ.

contradictory

Proof of soundness (Induction)

Assume valuation $v(\phi)=T$

$n=$ number of iterations.

$n=0$. $v(\phi)=T$. by assumption.

Assume n iteration $\mathcal{E}\in Tab$, $\theta\in\mathcal{E}\to v(\theta)=T$.

for a new iteration $\theta\in\mathcal{E}\to v(\theta)=T$.

if $\mathcal{E}$ is dequeued and $\psi\in\mathcal{E}$ is picked.

$v(\psi)=T$.

if ψ is α. $v(\alpha_1)=v(\alpha_2)=T$.

$\theta\in\mathcal{E}\to v(\theta)=T$.

if ψ is β $v(\beta_1)=T$ or $v(\beta_2)=T$.

$\theta\in\mathcal{E}_1\to v(\theta)=T$.

or $\theta\in\mathcal{E}_2\to v(\theta)=T$.

$\blacksquare$

Soundness of Predicate Tableau

Similar approach. $\mathcal{E}\in Tab$, $S,A\models\mathcal{E}$

if ψ is δ, $\psi=\exists x\,\theta(x)$. then $S,A\models\exists x\theta(x)$

$\exists A'\equiv_x A$.

A' interprets x to a valid value p

$S,A'\models\theta(x)$.

$\exists x\theta(x)$ replaced by $\theta(p)$

let S' be same as S. except

$I(p)=A'(x)$

Ancestors.

... is dequeued ...

if $\mathcal{E} \in Tab$ is dequeued and $\mathcal{E}_1, \mathcal{E}_2$ are enqueued,
$\mathcal{E}$ is parent of $\mathcal{E}_1 \mathcal{E}_2$.

$p(\mathcal{E}) = \mathcal{E}'$.
$p^0(\mathcal{E}) = \mathcal{E}$
$p^{n+1}(\mathcal{E}) = p(p^n(\mathcal{E}))$
$\mathcal{E}$ is the ancestor of $\mathcal{E}'$ if $n \geq 0$ and $p^n(\mathcal{E}') = \mathcal{E}$

Init tableau $T[\{\phi\}]$ and $\{\phi\}$ is ancestor of every theory in tableau.

$\therefore S; A \models \mathcal{E}$.

if ψ is γ. $\psi = \forall x\,\theta(x)$. $S, A \models \forall x\,\theta(x)$
$S, A \models \theta(t)$ for all closed term

$\therefore \forall x\,\theta(x)$ ~~is~~ replaced $\theta(t)$ in $\mathcal{E}$

$S, A \models \mathcal{E}$ still true.

Completeness of Prop. Tableau.

Proof: SATISFIABLE output $\Rightarrow \mathcal{E} \in Tab$ dequeued and $Exp(\mathcal{E})$ and $\neg C(\mathcal{E})$

Def. $v(p) = T \Leftrightarrow p \in \mathcal{E}$ (anything that's included in $\mathcal{E}$).

Prove by induction over n that
$\psi \in p^n(\mathcal{E}) \Rightarrow v(\psi) = T$.

True for $n = 0$.

IHypothesis $\psi \in p^n(\mathcal{E}) \Rightarrow v(\psi) = T$.

$\theta \in p^{n+1}(\mathcal{E})$.

Either θ is in $p^n(\mathcal{E})$ or θ is expanded in $p^n(\mathcal{E})$

if θ is in $p^n(\mathcal{E})$
$v(\theta) = T$.

if θ is expanded by α.

$\theta = \alpha \Rightarrow \alpha_1, \alpha_2 \in p^n(\mathcal{E})$
by IH $v(\alpha_1) = v(\alpha_2) = T$.
$\alpha_1, \alpha_2 \in p^n(\mathcal{E})$.
$\therefore v(\theta) = v(\alpha_1) \wedge v(\alpha_2) = T$.

$\theta = \beta$.
β_1 or $\beta_2 \in p^n(\mathcal{E})$.
$v(\beta_1) = T$ or $v(\beta_2) = T$.
$\therefore v(\theta) = v(\beta_1) \vee v(\beta_2) = T$.
$\therefore v(\beta) = T$.

if a fmla is valid,
if can be proved.
⇕
if a $\neg$ fmla is not satisfiable,
then the tableau of $\neg$fmla must close
⇕
if a $\neg$fmla is open (SATISFIABLE)
the $\neg$ fmla is satisfiable.

Termination of propositional tableau algo.

* when running tableau ϕ, only new theories are expanded.
* let X be a set of subformulas of ϕ and single negations of subformulas $2|\phi|$ of these.
* a theory is a subset of X. $2^{2|\phi|}$ of these
* Algo. stops in $2^{2|\phi|}$ steps at most

Herbrand Structures

A closed term t is built from const. and func. no vars.

Herbrand structure $H = (D, I)$ has.

Domain $D = \{$ closed terms $\}$.

Interpretation $I = (I_c, I_f, I_p)$.

$I_c(c) = c.$

$I_f(f^n) : (d_1, \dots d_n) \mapsto f^n(d_1, \dots d_n)$

I_p can be chosen freely

$[t]^{H,A} = t.$

Herbrand structure's purpose is to make a interpretation as simple as possible.

symbol terms as their values

Herbrand Theorem.

Let L be a lang. with ∞ const symbols and no equality predicate

if ϕ is satisfiable, $S, A \models \phi$, ϕ is satisfiable in a Herbrand structure $H, A \models \phi$

(some A)

Fairness

Suppose you have $P_1 \cdots P_k$ waiting for input.

you should, in a fair schedule, any P_i waiting for input at time t then eventually at t' $(t' > t)$ P_i will get input.

If process are always waiting, each process will get input infinitely often.

Since tot. requests for input $r \leq k$ is contable, it's possible to find a fair schedule

Completeness of predicate tableau.

if tableau for ϕ never closes and expanded by a fair schedule. (can be infinite) ϕ is satisfiable

König's Tree Lemma.
Let T be a tree each node has a finite branching factor. If every branch is of finite length, the number of nodes in tree is finite

if a tableau never closes,

a seq. $S_0, S_1, \cdots \in Tab$

where $S_n = P(S_{n+1})$

Let $S = \bigcup_{n<\infty} S_n$.

$\alpha \in S \Rightarrow \alpha_1 \in S,\ \alpha_2 \in S$

$\beta \in S \Rightarrow \beta_1 \in S$ or $\beta_2 \in S$

$\exists x\, \theta(x) \in S \Rightarrow \theta(p) \in S$.

$\forall x\, \theta(x) \in S \Rightarrow \theta(t) \in S$. for all closed term T

$\neg\forall x\, \theta(x)$

$\neg\theta(p)$

$\neg\exists x\, \theta(x)$ ---- $\neg\theta(t)$

Let H be Herbrandt structure.

D {closed terms of Σ}

$I(t) = t$.

$(t_1, \ldots, t_n) \in I(R^n) \Leftrightarrow R^n(t_1, \ldots, t_n) \in S$

$(t_0 \dots t_{k-1}) \in I(R) \Leftrightarrow R(t_0 \dots t_{n-1}) \in \mathcal{S}$

Try to show $\theta \in \mathcal{S} \Rightarrow H \models \theta$. and $\neg\theta \in \mathcal{S} \Rightarrow H \models \neg\theta$

atomic fmla. — base case

$\alpha_1 \in \mathcal{S},\ \alpha_2 \in \mathcal{S} \Rightarrow H \models \alpha$:

$\beta_1 \in \mathcal{S}$ or $\beta_2 \in \mathcal{S} \Rightarrow H \models \beta$.

$\theta(p) \in \mathcal{S} \Rightarrow H \models \exists x \theta(x)$

p is a const.

$\forall t, \theta(t) \in \mathcal{S} \Rightarrow H \models \forall x \theta(x)$

$\perp$. $H \models \phi$.

Eq. rules.

$$\begin{matrix} A(t) \\ t = s \end{matrix} \Rightarrow A(s).$$

$$\begin{matrix} A(t) \\ s = t \end{matrix} \Rightarrow A(s).$$

$\neg(t = t) \Rightarrow x$. ← anything is true.

Other proof systems

Tableau — Tests satisfiability
Easy to implement

Axiomatic — Easy to define
hard to use.

?? Resolution Theorem Provers. — Checks validity. Prolog

Natural Deduction — Easy to read.

Truth table — Exponential time.
Doesn't work for predicate logic

Theorem Proving for Predicate Logic. Axiomatic.

Quantifier.

$\forall x \neg A \Leftrightarrow \neg \exists x A$.

$\forall x A(x) \to A(t/x)$ if t is a sub. for x in A.

$\forall x (A \to B) \to (\forall x A \to \forall x B)$

Equality Axioms. $=$ usually can be used without def.

$x = x$.

$(x = y) \to (y = x)$

$(x = y) \to (t(x) = t(y/x))$

$(x = y) \to (A(x) \to A(y/x))$. sub. y for x but must do so for all x

why don't we need

$((x = y) \land (y = z)) \to (x = z)$.

$(x = y) \to (y = x)$ Ax.

$(y = x) \to (y = z \to x = z)$ $\underline{A(y = z)}$ Ax.

$(x = y \land y = z) \to (x = z)$

Inference Rules

Modus Ponens

$$\frac{A, \quad A \to B}{B}.$$

Universal Generalisation.

$$\frac{A(x)}{\forall x A(x)} \leftarrow \underline{A(x) \text{ valid}}.$$

Proofs

A ~~proof~~ proof of ϕ is a finite seq.

$\varphi_0, \phi_1, \phi_2 \cdots, \phi_n = \phi$.

ϕ_i is either axiom

or obtained from ϕ_j and ψ_k.

where $j, k < i$.

by a inference rules

(modus, ponens)

$\vdash \phi$

Proving from hypotheses.

(not consider all possible first order structures)

hypotheses are fmlas which are valid in the type of fmla you want.

Ex. linear order models.

totality of order. $\forall x \forall y (x<y \vee y<x \vee x=y)$

irreflexive. $\forall x \neg(x<x)$

transitivity $\forall x \forall y \forall z ((x<y \wedge y<z) \to x<z)$

Proof with hypothesis

Γ is a set of hypotheses.

gamma

$\Gamma \vdash \phi$.

proved based on Γ

$\phi_0 \cdots \phi_n = \phi$.

ϕ_i is an axiom

by inference rule

$\in \Gamma$.

Ex. $\Gamma_{\text{linear order}} \vdash \forall x \forall y \neg(x<y \wedge y<x)$.

$\forall x \forall y \forall z ((x<y \wedge (y<z) \to (x<z))$

Substitution $z \to x$

$$\forall x \forall y \forall z ((x<y) \wedge (y<z) \to (x<z))$$
$$\to ((x<y) \wedge (y<x) \to (x<x))$$

$$(x<y) \wedge (y<x) \to x<x$$

$$((x<y) \wedge (y<x) \to (x<x)) \to (\neg(x<x) \to \neg(x<y \wedge y<x)). \quad (\text{Ax. III})$$

$\therefore \neg(x<x) \to \neg(x<y \wedge y<x).$

$\vdash \forall x \neg(x<x)$ (Hypothesis)

$\therefore \neg(x<x).$ Ax IV.

$\therefore \neg(x<y \wedge y<x)$ Modus Ponens.

$\therefore \forall x \forall y \neg(x<y \wedge y<x)$ (Universal. Generalisation)

Entailment

$S \models \Gamma.$

if $S \models \phi \ \forall \phi \in \Gamma$ (S is a model of Γ)

Structure.

$\Gamma \models \phi$

if every model S of Γ is a model of ϕ

$S \models \Gamma \Rightarrow S \models \phi$

Strong Completeness.

$\Gamma \vdash \phi \Leftrightarrow \Gamma \models \phi.$

hypotheses $\xrightarrow{\text{proves}}$ finite $\Leftrightarrow$ All S that proves Γ. proves ϕ

Tableau Summary.

* Tableau is sound and complete for first order logic.

predicate logic

Gödel's completeness theorem.

* Sound All provable ϕ is valid. $\Leftrightarrow$ If $\neg\phi$ closes, $\neg\phi$ is not satisfiable.
  If ϕ is satisfiable, ϕ will not close.

* complete ϕ is valid, it's provable.

  if $\neg\phi$ is not satisfiable, $\neg\phi$ tableau should close, if fair seq. is used

Recursive Lang (Turing decidable language)

A language L is set of strings over finite alphabets Σ

L is recursive. if $\exists$ program.
determine if a $s \in \Sigma^*$, $s \in L$ or not
it's guaranteed to terminate (Turing decidable)

(E.g. valid statements of FOL/predicate logic form a lang, but not recursive (turing decidable))

Recursively Enumerable

L is (r.e.) if $\exists$ program outputs any given string from L. (only from L)

Valid Predicate logic/FOL fmlas is recursively enumerable.

Let $\phi_0, \phi_1, \ldots$ be an emeration of fmlas

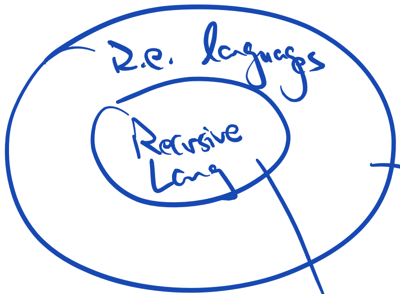
```
for (i=0, i<∞, i++) {
    start tableau Ti with φi at root.
    for (j=0, j<i, j++) {
        Expand Tj once with a fair schedule (queue)
```

if $T(\neg\phi_j)$ close, output
(ϕ_j) is valid.

}

},

program only outputs valid fmla
any valid fmla will get output (not guaranteed to terminate)



machine only halts if $s \in L$.

machine halts if $s \in L$ or $s \notin L$.

$$\underbrace{\text{true statements of arithmetic isn't even r.e.}}_{\text{Gödel's incompleteness theorem}}$$

Proof: Gödel's Number

$$\underset{1\ 2\ \ 3\ 2\ 4\ 5\ 6}{\forall x\ (x > 0)}$$

$$G = 2^1 \times 3^2 \times 5^3 \times 7^2 \text{---}$$

Fixed Point Theory $A(x)$ $B(x)$

$$\text{Apply}(|A(x)|, |B(x)|) = |A(|B(x)|)|$$

$\forall F(x)$. Def $G(x) = F(\text{apply}(x, x))$

$$A = G(|G(x)|) = F(\text{apply}(|G(x)|, |G(x)|))$$
$$= F(|G(|G(x)|)|)$$
$$= F(|A|)$$

forall $F(x)$

$\exists$ A as a fixed point.

provable(n) n as a Gödel number.

formula describing provable(n) is PROVABLE(N)

with fixed-point theory, unprovable(A) = A.

↺ cycle.

proved A is fmla that's unprovable.

1. Arithmetic system is not complete.

Proving from Assumptions.

WTS. ϕ is valid in a model or a type of model.

write $\mathcal{S}$ assumptions set.

* Add any $\phi \in \mathcal{S}$ at the leaf of T at any time.
* if prove $\mathcal{S} \vdash \phi$.

$$\mathcal{S} = \{ \forall x \neg(x<x)$$
$$\forall x \forall y (x<y \vee y<x \vee x=y)$$
$$\forall x \forall y \forall z ((x<y \wedge y<z) \to x<z) \}.$$

$$\mathcal{S} \vdash \forall x \forall y \neg (x<y \wedge y<x)$$

$\neg \forall x \forall y \neg (x<y \wedge y<x)$ ✓

| δ

$\neg \forall y \neg (a<y \wedge y<a)$ ✓

| δ

$(a<b \wedge b<a)$. ✓

α

$a<b$

$b<a$.

$\forall x \forall y \forall z\, ((x<y \wedge y<z) \to x<z)$

γ, a

$\forall y \forall z\, ((a<y \wedge y<z) \to a<z)$

γ, b

γ, a

$((a<b \wedge b<a) \to a<a)$

β

$\neg(a<b \wedge b<a)$

β

$\neg(a<b)$

$\otimes$

$\neg(b<a)$

$\otimes$

$a<a$

$\forall x\, \neg(x<x)$

γ, a

$\neg(a<a)$

$\otimes$

Compactness.

if $\Sigma \vdash \phi$ then $\Sigma_0 \vdash \phi$ $\Sigma_0 \subseteq \Sigma$.

Inconsistency.

Σ is inconsistant if $\Sigma \vdash p \wedge \neg p$

by compactness if Σ is inconsistent Σ_0 is inconsistant

By strong completeness consistent has a model / structure.

Compactness $\Rightarrow$ if every finite subset of Σ has a model, there is a model for Σ

FOL cannot have Connectedness (graph)

L $\quad C = \{c, d\}$
$F = \phi$ $\quad$ Edge.
$P = \{=, E\}$

Assume. $G \models \Sigma \Rightarrow G$ is connected.

a set of hypotheses that ensures a graph.

Let $\phi_1(x, y) = E(x, y)$
$\phi_{n+1}(x, y) = \exists z\, (\phi_n(x, z) \wedge E(z, y))$

ϕ_K means there is a connection length of k.

Consider $\Sigma \cup (\neg\phi_1(c, d), \neg\phi_2(c, d \cdots))$

a finite subset would certainly have a gap $\Sigma \cup (\neg\phi_1 \cdots \neg\phi_K)$
then we can make a $k+1$ length conn between c and d.

Every finite subset has a model, By compactness, the whole set has a model
$G \models \Sigma \cup (\neg\phi_n(c, d) : n = 1, 2, 3 \cdots\}$
G is connected, but no path $c - d$
Σ $\quad \neg\phi_n(c, d)$

FOL has limitation in Expressiveness

Compactness theorem and non-standard analysis

$\mathcal{S} = \{$ all valid statements of $\mathbb{N}\}$.

e.g. $2+2=4 \in \mathcal{S}$

$\mathcal{S}^+ = \mathcal{S} \cup \{w \neq 0, w \neq 1, \cdots, w \neq n, \cdots\}$

Every finite subset of $\mathcal{S}^+$ has a model.

$w = n+1$

$\therefore \mathcal{S}^+$ has a model.

Similarly if $\mathcal{S} = \{$ all valid statements of $\mathbb{R}\}$

$\mathcal{S}^+$ has a model. (α)

infinitely large real

$\therefore \frac{1}{\alpha}$ is a infinitely small + real number.

every real number (ordinary one)

$\forall x (|x| < r) \rightarrow (x = St(x) + Inf(x))$

standard real

Infinitely small real.

$\therefore f'(x) = St\left(\frac{f(x+\delta x) - f(x)}{\delta x}\right)$.

Paradoxes

- Liar Paradox.
- Russell's Paradox. $R = \{S : S \notin S\}$ (Barber)

Russell set

$R \in R$?

— Berry's Paradox   smallest N cannot be defined
by 10 english words
is defined by 9 english
words

Another Proof of Gödel's Incomplete Theorem

$L \Rightarrow C = \{0, 1, 2 \cdots\}$
$F = \{+, \times\}$
$P = \{=, <\}$.

Theorem (Gödel).

If S is any r.e. set of L-sentences.
There is a ϕ which is true
but $\phi \notin S$. incomplete
or ϕ is false and $\phi \in S$ inconsistent.

Gödel Number is a unique number
assigned to every fmla.

Every proof is a list of Gödel Number
can be written as a single
Number.

$G(\phi) = n$. $G(\bar{\phi}) = m$
$F(n) = \underline{\phi}$ a proof, a.k.a a list of ϕ.
$P(m) = \phi$

Proof.

$\mu(n, m) = P(n)$ is a proof of $F(m)$

$\gamma(n) = F(n)$ is a fmla with one var x.

Let. $A_0(x) \ldots A_n(x) \ldots$

be an enum. of fmla with variable x.

$\mu(n, k, q) =$ $P(n)$ is a proof of $A_k(q)$.

Consider fmla.

$$\neg\exists n \ \mu(n, x, x).$$

this is a fmla with one free var x

$\therefore A_{n_0}(x) = \neg\exists n \ \mu(n, x, x)$

$N \models A_{n_0}(x)$ iff. there is no proof of $A_x(x)$

$N \not\models A_x(x)$.

$\exists A_{n_0}(n_0)$

$N \models A_{n_0}(n_0)$ iff there is no proof of $A_{n_0}(n_0)$

either incomplete or inconsistent

Other Logic

model logic

$\Diamond p$. — p is possible
— p might happen.

$\Box p = \neg\Diamond\neg p$ not p is not possible
p is definite
p will happen.

Grammar.

$prop ::= p \mid q \mid r \ldots$

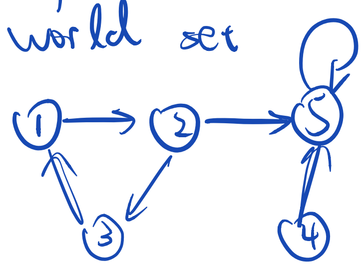
$\phi ::= prop \mid \neg\phi \mid \phi\wedge\phi \mid \phi\vee\phi \mid \phi\to\phi \mid \Diamond\phi \mid \Box\phi$

$\Diamond q \to \Box\Diamond p$

Frames

$F = (W, R)$. $R \subseteq W \times W$.

↑ world set



Valuation.

$V: prop \rightsquigarrow P(W)$. $\wp(W)$.

Power Set of W.

$V(p) = \{1, 3, 5\}$.

$V(q) = \{1, 2\}$

Frame with Valuation = Model.

let $w \in W$, (w is a world)

$M, w \models p \Leftrightarrow w \in V(p)$.

$M = (F, V)$

$= (W, R, V)$

valuations

set of worlds, relations