COMP0009 Revision Notes

Formal Logic : Syntax. how to write | Semantics,, what it means | proof system (Inference) 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 / $\neg p$

Main Connective : $(p \wedge q) \vee \neg(q \to r)$
the connective with the largest scope (most outside, highest level)

Semantics

v → evaluation. $\top$ truth $\bot$ 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 non-empty predicate (relation) symbol set P^n (n-ary)

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

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

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

$+, x \in f^2.$

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

$x + y < 2 \times y - 1 \quad < \text{ is a } p^2$

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

$(fm \wedge fm)$

$(fm \to fm)$

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

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)

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

domain interpretation

$S = [D, I]$

structure

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

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

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

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

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

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

$S, A \models (\phi \vee \phi') \Leftrightarrow S, A \models \phi$ 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. $(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$

$\Updownarrow$

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

$\Updownarrow$

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

$\Updownarrow$

$$\Updownarrow$$

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

$$\Updownarrow$$

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

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

$$\Updownarrow$$

true.

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

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

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

$$\Updownarrow$$

true.

Inference Rule.

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

$$\frac{\phi \quad (\phi \to \psi)}{\psi}$$

modus tollens. if proved $\neg q$ and $p \to q$. then deduce $\neg p$

$$\frac{\neg q \quad (p \to q)}{\neg p}$$

Proof.

a proof is a seq. of fmlas.

$\phi_0 \cdots \phi_n$.

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

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

$$\Downarrow$$

ϕ_i

If such a proof exists, ϕ_n is a theorem.

$\vdash \phi_n$.

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

$\Gamma \vdash \phi$

if there's a proof of ϕ using assumptions from Γ.

$\phi_0 \cdots \phi_n$.

ϕ_i is either * axiom.

* an assumption.

* obtained from ϕ_j ϕ_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

$\Updownarrow$

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

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

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

ϕ $\quad$ ψ

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

$\oplus$ $\oplus$

closes both $\rightarrow$ not satisfiable.

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

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

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.

$\neg\neg\phi$ | ϕ

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

$\neg(\phi\rightarrow\psi)$ | ϕ | $\neg\psi$.

$\phi\wedge\psi$ | ϕ | ψ.

β formula. $\phi\vee\psi$

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

$\phi\vee\psi$ / ϕ \ ψ.

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

$\phi\rightarrow\psi$ / $\neg\phi$ \ ψ.

Example. $p\vee r$ β.

$\neg q$ X

$\neg((p\vee q)\rightarrow r)$ α.

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

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

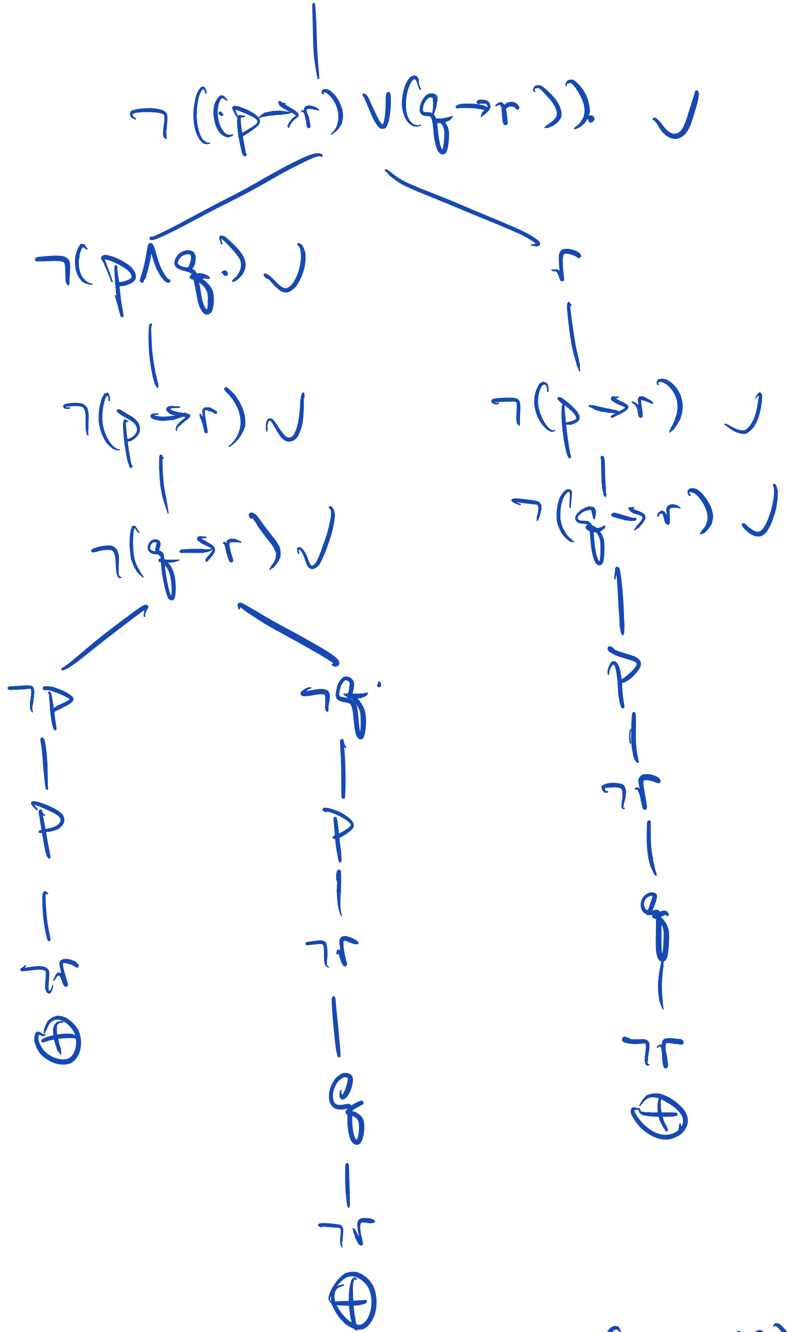
test ϕ is valid $\Leftrightarrow$ test $\neg\phi$ is not satisfiable

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.

$((p \wedge q) \to r)$ ✓

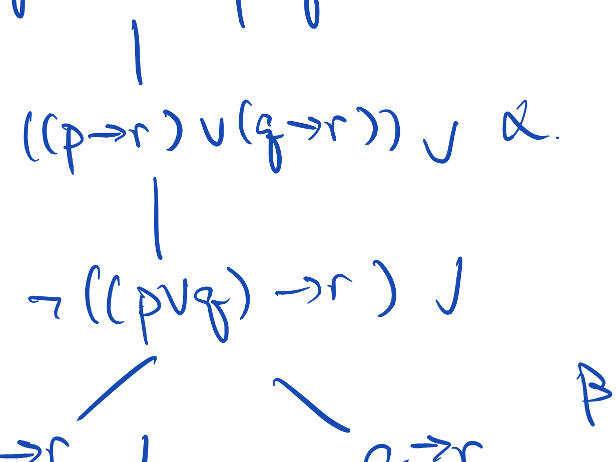


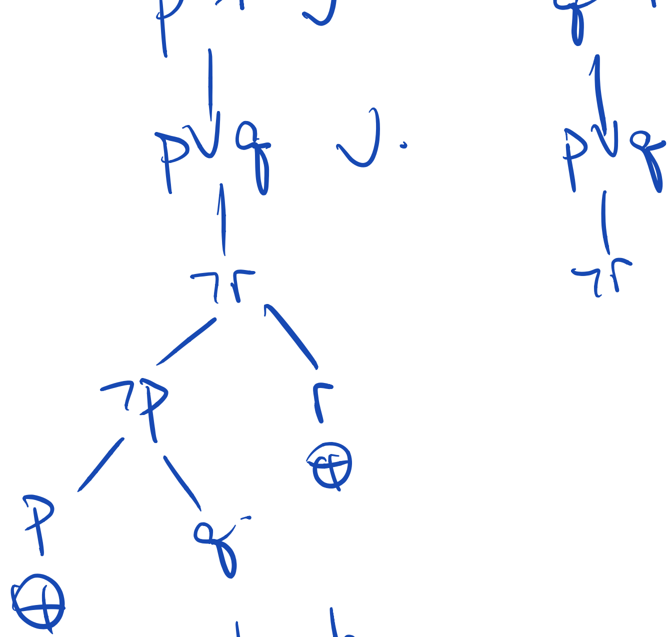
$\therefore\ (((p\wedge q)\to r)\to((p\to r)\vee(q\to r)))$

$\quad\therefore$ 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.





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.

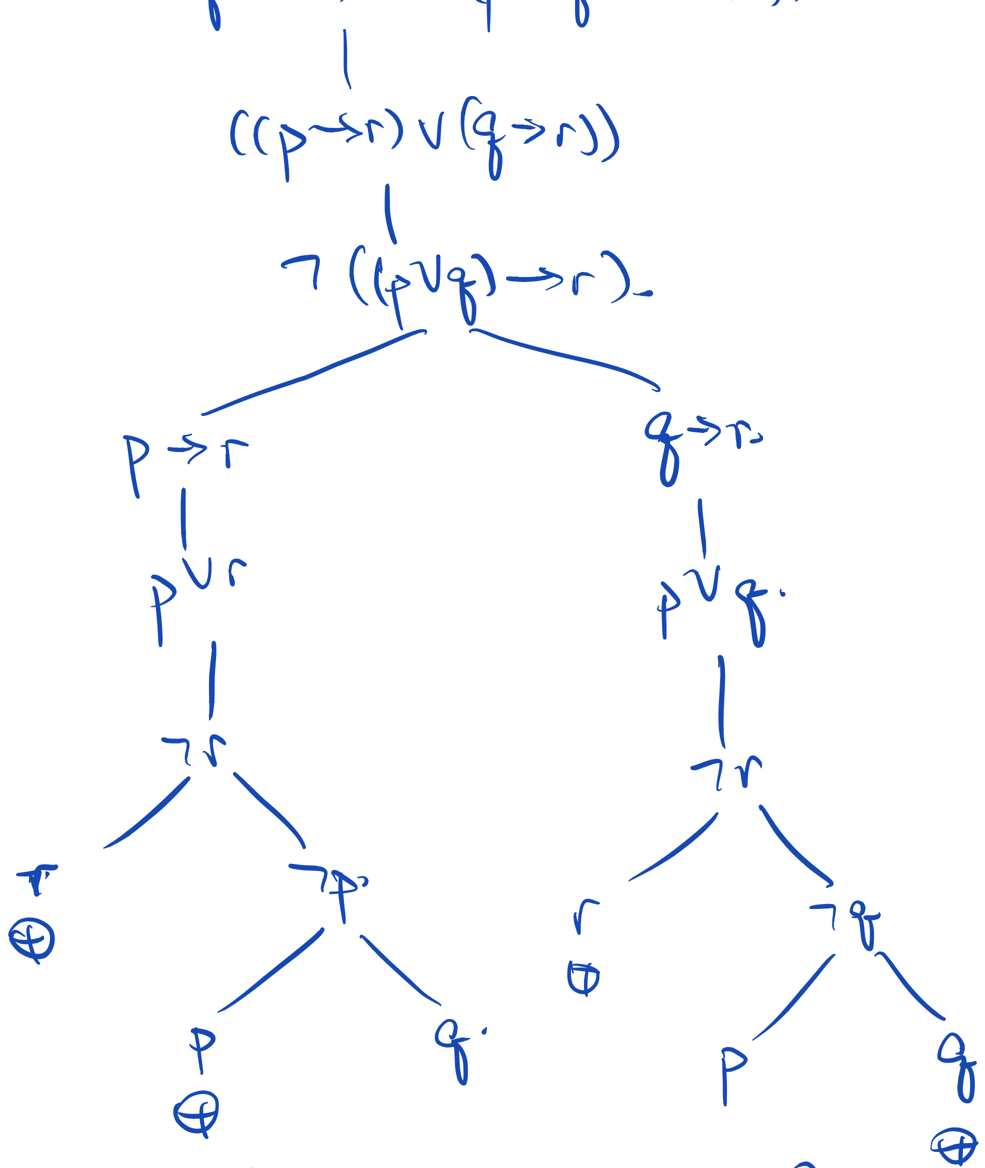
truth table / logical equivalences / tableau

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

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

then $\phi\equiv\vee C_\theta$

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



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.

$\exists x \phi$ $\qquad$ $\neg \forall x \phi$

$|$ $\qquad$ $|$

$\phi(p/x)$. $\qquad$ $\neg \phi(p/x)$

$$\underbrace{\exists x \neg \phi \equiv \neg \forall x \phi}$$

γ formula

pick any closed term t.

don't tick node.

until you used every available const/closed term

$\forall x \phi$. $\qquad$ $\neg \exists x \phi$

$|$ $\qquad$ $|$

$\phi(t/x)$. $\qquad$ $\neg \phi(t/x)$.

$$\underbrace{\neg \exists x \phi \equiv \forall x \neg \phi}.$$

$\forall x \neg p(x)$. $\quad \gamma$

$H(a) \to F(a)$ $\quad \beta$.

$\neg\neg \exists y\, p(y)$. $\cancel{\delta}$ (double negation first) α rule

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

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

$G(a) \to H(a)$. $\quad \beta$

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

$\neg \exists x (G(x) \wedge F(x))$ $\quad \gamma$

$\neg \forall y \neg (c < y)$ $\quad \delta$.

$\forall x (H(x) \to F(x))$ $\quad \gamma$.

$G(a)$ $\quad$ X,

$\neg p(c)$ $\quad$ X.

$\forall x (G(x) \to H(x))$ $\quad \gamma$.

$\neg H(a)$. $\quad$ X -

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

$\neg G(a)$. $\quad$ X

$\neg(G(a) \wedge F(a))$ 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. $\neg(\forall x(Gx \to Hx) \wedge \forall x(Hx \to Fx) \wedge Ga \wedge$
$\neg\exists x(Gx \wedge Fx)$ valid?

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

$\forall x(Gx \to Hx)$

$\forall x(Hx \to Fx)$

Ga.

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

$\neg(Ga \wedge Fa)$ ✓.

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

$\oplus$

$\neg Ha$ $\quad$ $Fa.$

$\oplus$

$Ga \to Ha$

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

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

$\therefore \neg(\forall x(Gx \to Hx) \land \forall x(Hx \to Fx) \land Ga \land \neg\exists x(Gx \land Fx))$ 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 $\phi_1 \wedge \phi_2$ $\neg(\phi_1 \vee \phi_2)$ $\neg(\phi_1 \to \phi_2)$, $\neg\neg\phi$

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

Tab = [{ϕ}].

while ! Tab.empty() {.

branch (set). —— ξ = Tab.dequeue();

if ξ is fully expanded. and. ξ doesn't have contradiction,

SATISFIABLE

else.

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

swith (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,

swith (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

— — . $\neg \exists x \theta(x)$

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

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

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 $S, A \models \exists x\, \theta(x)$

$\exists A' =_x A$.

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.

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 $\forall$ 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 $v(\alpha_1) = v(\alpha_2) = T$.

$\alpha_1, \alpha_2 \in P^n(\mathcal{E})$.

$\therefore v(\theta) = v(\alpha_1) \wedge v(\alpha_2) = T$.

if a fmla is valid,

if can be proved.

$\Updownarrow$

if a $\neg$ fmla is not satisfiable,

then the tableau of $\neg$fmla must close

$\Updownarrow$

if a $\neg$fmla is open (SATISFIABLE)

the $\neg$ fmla is satisfiable.

$\theta = \beta$.

β_1 or $\beta_2 \in P^n(\mathcal{E})$.

$v(\beta_1) = T$ or $v(\beta_2) = T$.

$\therefore v(\theta) = v(\beta_1) \vee v(\beta_2) = T$.

$\therefore v(\beta) = T$.

Termination of propositional tableau algo.

* when running tableau ϕ, only new theories are expanded.
* let X be a set of subfmlas of ϕ and single negations of subfmlas. $2|\phi|$ 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 $\underset{\sim}{t}$ 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 $r \le k$ is countable, 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

King'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 $\beta_2 \in S$

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

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

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

$\neg\theta(p)$

$\neg\exists x\,\theta(x) \quad \text{---} \quad \neg\theta(t)$

Let H be Herbrandt structure.

D {closed terms of Σ}

$I(t) = t$.

$(t_0 \ldots t_{k-1}) \in I(R) \iff R(t_0 \ldots t_{n-1}) \in S$

Try to show $\theta \in S \Rightarrow H \models \theta$ and $\neg\theta \in S \Rightarrow H \models \neg\theta$

atomic fmla. — base case

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

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

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

p is a const.

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

c. $H \models \phi$

Eq. rules.

$$\begin{array}{l} A(t) \\ t = s \end{array} \Rightarrow A(s)$$

$$\begin{array}{l} A(t) \\ s = t \end{array} \Rightarrow A(s)$$

$\neg(t = t) \Rightarrow x$. ← 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

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

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

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

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

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

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

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

Substitution $z \to x$

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

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

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

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

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

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

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

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

Entailment

$S \models \Gamma$.

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

$\Gamma \models \phi$

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 completness 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 guranteed 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