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 / $\neg p$

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.

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

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

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

$v(\phi \to \psi) = \top \Leftrightarrow v(\neg\phi) = \top$ 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 symbol set (relation) P^n (n-ary)

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

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

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

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

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

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

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

$(fm \wedge fm)$

$(fm \to fm)$

$\forall x \phi$ 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, fncs, predicates

I_C maps constant symbols in C to elements of D

I_f — — n-ary fncs $f \in F$ 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. $I_P(=) = \{(d,d) : d \in D\}$

interpretation

domain

$S = [D, I]$

structure

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

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

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

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

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

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

$S, A \models (\phi \vee \phi') \Leftrightarrow S, A \models \phi$ 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.t. $A \models \phi$.
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.

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

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

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

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

$\Downarrow$

ϕ_i

If such a proof exists, ϕ_n is a theorem.
$\underset{\sim}{\vdash \phi_n}$.

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

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

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

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

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

$(p \to p)$.

Proofs with other conn

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

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

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 from T.
$\phi_0 \cdots \phi_n$.
ϕ_i is either * axiom.
* an assumption.
* obtained from $\phi_j \phi_k$.
$(j, k < i)$

Ex. $\bot \vdash p$
$\{\neg(q \to q)\} \vdash p$.
- $q \to q$.
- $(q \to q) \to \neg\neg(q \to q)$ Ax. IV.
- $\neg\neg(q \to q)$. MP.

$\because\ \neg\neg(q\to q)\to(p\to\neg\neg(q\to q))$ 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 II.

$\therefore\ \neg(q\to q)\to\neg p$

$\because\ \neg(q\to q)$ Assumption.

$\therefore\ \neg p$

$\because\ \neg\neg(q\to q)\to(\neg p\to\neg\neg(q\to q))$

$\therefore\ \neg p\to\neg\neg(q\to q)$

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

$\therefore\ \neg(q\to q)\to p$

$\because\ \neg(q\to q)$ Assum.

$\therefore\ p$

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

$\neg\phi$ ⊕ $\neg\psi$ ⊕

closes both $\longrightarrow$ 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 brach 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.

- $\phi\vee\psi$
  - ϕ
  - ψ.

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

- $\phi\to\psi$
  - $\neg\phi$
  - ψ.

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

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

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

$((p\wedge q)\to r)$ ✓

$\neg((p\to r)\vee(q\to r))$ ✓

Left branch:

$\neg(p\wedge q)$ ✓

$\neg(p\to r)$ ✓

$\neg(q\to r)$ ✓

- $\neg p$, p, $\neg r$, $\otimes$
- $\neg q$, p, $\neg r$, q, $\neg r$, $\otimes$

Right branch:

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.

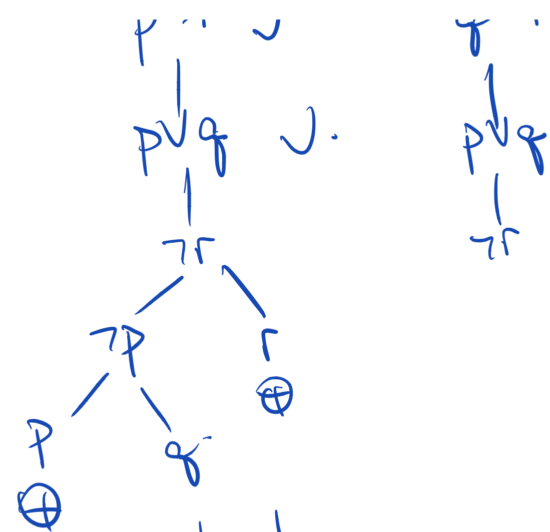
$\Leftrightarrow$ is $\neg(((p\to r)\vee(q\to r))\to((p\vee q)\to r))$ ✓ satisfiable.

$((p\to r)\vee(q\to r))$ ✓ α.

$\neg((p\vee q)\to r)$ ✓

β

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



open branch

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

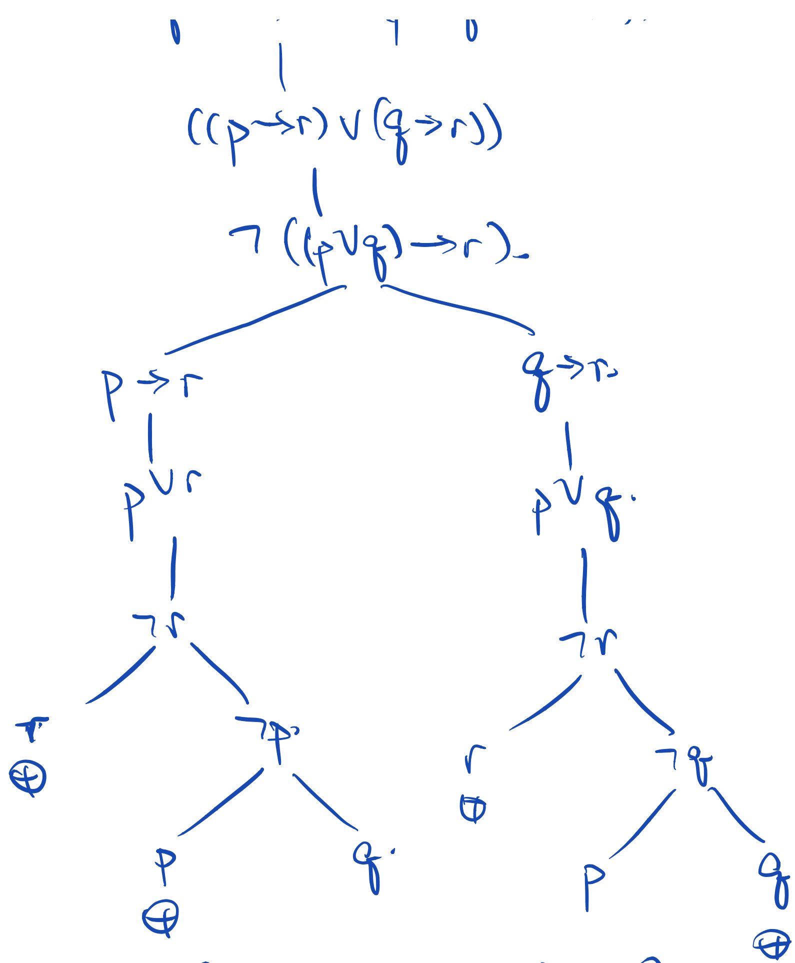
truth table / logical equivalences / tableau

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

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

then $\phi\equiv\bigvee C_\theta$

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



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 fula of each leaf below the node.

$$\exists x\, \phi \qquad \neg \forall x\, \phi$$
$$| \qquad\qquad |$$
$$\phi(p/x). \qquad \neg \phi(p/x)$$

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

γ formula

pick any closed term t.

don't tick node.

until you used every available const/closed term

$$\forall x\, \phi. \qquad \neg \exists x\, \phi$$
$$| \qquad\qquad |$$
$$\phi(t/x). \qquad \neg\phi(t/x).$$

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

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

$\neg(G(a) \wedge F(a))$ β.

$\exists y\, p(y)$. δ.

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.

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

$\neg(Ga \wedge Fa)$ ✓.

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

$\oplus$

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

$\oplus$

$Ga \to Ha$

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

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

∴ $\neg(\forall x(Gx \to Hx) \wedge \forall x(Hx \to Fx) \wedge Ga \wedge \neg\exists x(Gx \wedge Fx))$ is valid

Infinite Tableau.

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

$\neg\forall x \neg q(x)$ ✓

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

$\neg\neg q(c)$ ✓

$q(c)$

$\neg\forall y \neg(c<y)$ ✓

$\neg\neg(c<d).$ ✓

for infinite tableau it's satisfiable.

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

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

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

$c < d$

$\neg \forall y \neg (d < y) \checkmark$

$\neg\neg (d < e) \checkmark$

$d < e$

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). — S = Tab.dequeue();

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

SATISFIABLE

else.

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

swith (φ)

case α:

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

make an α expansion on φ

$\Sigma[x/y]$,

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 (φ)

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 notC(Σ) 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 $\mathcal{E} \in Tab$, $\theta \in \mathcal{E} \Rightarrow v(\theta) = T$.

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

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

$v(\psi) = T$.

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

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

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

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

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

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.

... is closed.

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 $n=0$.

I Hypothesis $\psi \in p^n(\mathcal{E})$

$\Rightarrow v(\psi) = T$.

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

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, it 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 expand.
* let X be a set of sub fmlas of ϕ and single negations of sub fmlas $2|\phi|$ 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 $I = (I_c, I_f, I_p)$.

$I_c(c) = c$.

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

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 AI)

Fairness

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

you should, in a <u>fair</u> 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 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

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. $\Sigma_0, \Sigma_1, \ldots \in Tab$
where $\Sigma_n = P(\Sigma_{n+1})$

Let $\Sigma = \bigcup_{n<\infty} \Sigma_n$.

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

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

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

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

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

$\neg \theta(p)$

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

Let H be Herbrandt structure.

D {closed terms of Σ}

$I(t) = t$.

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

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.

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

i.e. $H \models \phi$.

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.

$x = x$.

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

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

$(x = y) \to (A(x) \to A(y/x))$.

why don't we need

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

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

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

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

Inference Rules

Modus Ponens

$$\frac{A, \; A \to B}{B}$$

Universal Generalisation.

$$\frac{A(x)}{\forall x A(x)}$$

$\leftarrow$ $A(x)$ valid.