

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

## Propositional Logic.

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

$f_m := \text{prop} \mid \neg f_m \mid f_m \circ f_m$   
 an operand  $\wedge, \vee, \rightarrow$

Literal: prop or its negation  $p \mid \neg p$

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

### Semantics

$v \rightarrow$  evaluation. Truths  $\top$  false.

$$v(\neg \phi) = T \Leftrightarrow v(\phi) = \perp$$

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

$$v(\phi \vee \psi) = T \Leftrightarrow v(\phi) = T \text{ or } v(\psi) = T.$$

$$v(\phi \rightarrow \psi) = T \Leftrightarrow v(\neg \phi) = T \text{ or } v(\psi) = T$$

### Validity, Satisfiability, Equivalence.

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

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

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

All valid formulae 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  $P^n$  ( $n$ -ary)  
(relation)

$tm ::= \nu; \nu \in \text{Var} \mid c: c \in C \mid f^i(tm \dots tm) : f \in F$ .

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

$2,3 \in C$   $x \in \text{Var}$ .  
 $+, \times \in f^2$ .

$\text{atom} := P^n(t_1, \dots, t_m) : P \in \mathcal{P}$

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

$\text{fm} := \text{atom} \mid \neg \text{fm} \mid (\text{fm} \vee \text{fm}) \mid \exists v \text{ fm} : v \in \text{Var}$

$(\text{fm} \wedge \text{fm})$

$(\text{fm} \rightarrow \text{fm})$

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

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  $D$

If  $\underline{\quad}$  many funcs  $f \in I$  to  $n$ -ary functions over  $D$

$I_P$   $\underline{\quad}$  many predicate symbols  $P \in \mathcal{P}$  to  $n$ -ary relations over  $D$

domain  $\downarrow$   $A : \text{Var} \rightarrow D$  is a variable assignment. / (subset of  $D^n$ )  
 $\uparrow$  structure  $S = [D, I]$  e.g.  $I_p(=) = \{(d, d) : d \in D\}$

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

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

$[f(t_0 \dots t_{n-1})]^{S,A} = I(f)([t_0]^{S,A} \dots [t_{n-1}]^{S,A})$

Repl.  $S, A \models R(t_0 \dots t_{n-1}) \Leftrightarrow ([t_0]^{S,A} \dots [t_{n-1}]^{S,A}) \in I(R)$

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

$S, A \models (\phi \vee \psi) \Leftrightarrow S, A \models \phi \text{ or } S, A \models \psi$

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

Validity

$S = (D, I)$  be a L-structure,  $\phi$  a formula.

$\phi$  is valid in  $S$ , for all  $A : \text{Var} \rightarrow D$  we have  $S, A \models \phi$

$S \models \phi$ .

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

$S \models \phi$ .

Satisfiability.

$\phi$  is satisfiable in  $S$  if  $\exists A : \text{Var} \rightarrow D$  such that  $S, A \models \phi$ .

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

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

## Propositional Proof System.

a system for determining validity of formula.

Obvious one: write down truth table for  $\phi$ .  
problem - exponential time.

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

$\geq 2^{50}$  possibilities.

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

Problem: how to make sure syntactical changes

make semantic sense?  
make sure the proof system is sound and complete.

$\vdash \phi \Leftrightarrow \phi$  is valid.

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

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

completeness  $\vdash \phi \Rightarrow \vdash \neg \phi$

if sth. is valid.

the system can prove it

$\vdash \phi \Leftrightarrow \vdash \neg \phi$

## Axiomatic Proof System.

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

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

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

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

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



$$\begin{aligned}
 & (\neg p \wedge \neg q \wedge r) \rightarrow ((\neg p \wedge q) \wedge (\neg p \wedge r)) \\
 & \quad \text{D} \\
 & \neg(\neg p \wedge \neg q \wedge r) \wedge (\neg p \vee \neg q) \wedge \neg p \wedge r \\
 & \quad \text{C} \\
 & (\neg p \vee q \vee r) \wedge (\neg p \vee \neg q) \wedge \neg p \wedge r \\
 & \quad \text{true.}
 \end{aligned}$$

$$\begin{aligned}
 \text{III. } & (\neg p \rightarrow \neg q) \rightarrow (q \rightarrow p). \\
 & \neg(p \wedge \neg q) \wedge (\neg q \wedge p) \\
 & (\neg p \vee q) \wedge (\neg q) \wedge p \\
 & \quad \text{II} \\
 & \text{true.}
 \end{aligned}$$

Inference Rule:

$$\begin{array}{c}
 \text{Modus Ponens. if proved } \phi \text{ and } \phi \rightarrow \psi. \\
 \frac{\phi \quad (\phi \rightarrow \psi)}{\psi}. \quad \text{then deduce } \psi.
 \end{array}$$

$$\begin{array}{c}
 \text{Modus tollens. if proved } \neg q \text{ and } p \rightarrow q. \\
 \frac{\neg q. \quad (\neg q \rightarrow \neg p)}{\neg p}. \quad \text{then deduce } \neg p.
 \end{array}$$

Proof.

a proof is a seq. of fmlas.

$$\phi_0 \cdots \phi_n.$$

s.t. for  $i \leq n$ .  $\phi_i$  is either <sup>\*Axiom</sup>  
<sup>\* obtained by  
modus ponens</sup>

$$\begin{array}{c}
 \boxed{\phi_j.} \\
 \text{e.g. } \boxed{\phi_k = \phi_j \rightarrow \phi_i} \quad (j, k < i) \\
 \downarrow \quad \downarrow
 \end{array}$$

if  $\vdash \phi_0 \cdots \vdash \phi_i$

If such a proof exists,  $\phi_n$  is a theorem.  
 $\vdash \phi_n$ .

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

$$\text{AxII. } (p \rightarrow ((p \rightarrow p) \rightarrow p)) \rightarrow (((p \rightarrow (p \rightarrow p)) \rightarrow (p \rightarrow p)))$$

$$\text{AxI. } (p \rightarrow ((p \rightarrow p) \rightarrow p))$$

$$1. \quad (p \rightarrow (p \rightarrow p)) \rightarrow (\neg p \rightarrow p)$$

$$\text{Ax I. } (\neg p \rightarrow (p \rightarrow p)).$$

$$2. \quad \neg p \rightarrow p.$$

Proofs with other conn

$$\text{IV. } p \rightarrow \neg \neg p \text{ and } \neg \neg p \rightarrow p$$

$$\vee (p \vee q) \Leftrightarrow (\neg p \rightarrow q). \xrightarrow{\text{def}} p \vee q$$

$$\text{VI. } (p \wedge q) \Leftrightarrow \neg (\neg p \vee \neg q) \rightarrow \neg (\neg p \vee \neg q) \\ \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \Leftrightarrow p \wedge q.$$

Proofs with assumptions

$\vdash \phi$

if there's a proof of  $\phi$  using assumptions from  $T$ .

$\phi_0, \dots, \phi_n$ .

$\phi_i$  is either  $\star$  axiom.

$\star$  an assumption.

$\star$  obtained from  $\phi_j, \phi_k$ .

( $j, k < i$ )

Ex.  $\vdash p$

$$\vdash (\neg q \rightarrow q) \vdash p$$

$$\neg q \rightarrow q.$$

$$\neg (\neg q \rightarrow q) \rightarrow \neg \neg (q \rightarrow q) \text{ Ax.IV.}$$

$$\neg \neg (q \rightarrow q). \text{ MP.}$$

$\neg \neg (q \rightarrow q) \rightarrow (\neg p \rightarrow \neg(q \rightarrow q))$   
 L.  $\neg p \rightarrow \neg(q \rightarrow q)$   
 $\neg (\neg p \rightarrow \neg(q \rightarrow q))$   
 $\rightarrow (\neg(q \rightarrow q) \rightarrow p)$  Ax III.  
 L.  $\neg(q \rightarrow q) \rightarrow p$   
 $\neg(q \rightarrow q)$  Assum.  
 L. P

$\neg \neg (q \rightarrow q) \rightarrow (p \rightarrow \neg(q \rightarrow q))$  Ax I.  
 $\therefore p \rightarrow \neg(q \rightarrow q)$   
 $\therefore (p \rightarrow \neg(q \rightarrow q)) \rightarrow (\neg(q \rightarrow q) \rightarrow \neg p)$  Ax III.  
 L.  $\neg(q \rightarrow q) \rightarrow \neg p$   
 $\neg(q \rightarrow q)$  Assumption.  
 L.  $\neg p$

Soundness J.

check all axioms

check if  $\phi$  and  $\neg\phi$ . then  $\phi$  is valid. Modus Ponens

all provable formulas are valid.

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

Completeness.

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

## Propositional Tableaux.

Given formula  $\phi$ , tableau. can tell us whether it is satisfiable or not.

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

↓  
atomic formula.

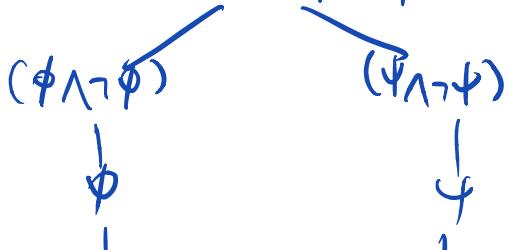
or its negation.

\* tableau. (closed or not).

if tableau closes  $\phi$  is unsatisfiable

tableau never closes  $\phi$  is satisfiable

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



$\neg\phi$   
 $\oplus$   
 closes both  
 $\neg\psi$   
 $\oplus$   
 not satisfiable.

Tableau T is a BT. (node labelled by formula)  
 Every formula 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.

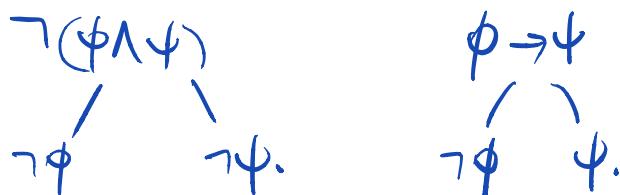
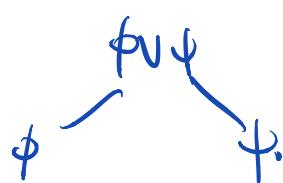
a formula.  $\phi \wedge \psi$ .

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

$\neg\neg\phi$	$\neg(\phi \vee \psi)$	$\neg(\phi \rightarrow \psi)$	$\frac{\phi \wedge \psi}{\phi}$
$\frac{\phi}{\neg\phi}$	$\frac{\neg\phi}{\neg\psi}$	$\frac{\neg\phi}{\phi}$	$\frac{\psi}{\neg\psi}$

B formula.  $\phi \vee \psi$

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



Example.  $p \vee r$  B.  
 $\neg q$  X  
 $\neg((\neg p \rightarrow r) \rightarrow s)$  A.

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

$$((p \rightarrow r) \vee (q \rightarrow r)). \quad \beta.$$

$$\neg(((p \rightarrow r) \vee (q \rightarrow r)) \rightarrow ((p \vee q) \rightarrow r)) \quad \alpha.$$

$$\neg(p \rightarrow r) \quad \alpha.$$

$$q \rightarrow r \quad \beta.$$

$$\neg r \quad x$$

$$\neg(p \wedge q) \quad \beta.$$

$$\neg((p \rightarrow r) \vee (q \rightarrow r)) \quad \alpha.$$

$$p \vee q. \quad \beta.$$

$$\neg(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r))) \quad \alpha.$$

$$\begin{array}{c} p \\ p \rightarrow r \end{array} \quad x$$

$$p \rightarrow r \quad \beta.$$

$$\neg(q \rightarrow r). \quad \alpha$$

$$\neg(((p \wedge q) \rightarrow r)). \quad \alpha.$$

\* a tableau is complete if either ticked (expanded)

\* if  $\phi$  is at the root or is a literal, of a complete open tableau,  $\phi$  is satisfiable.

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

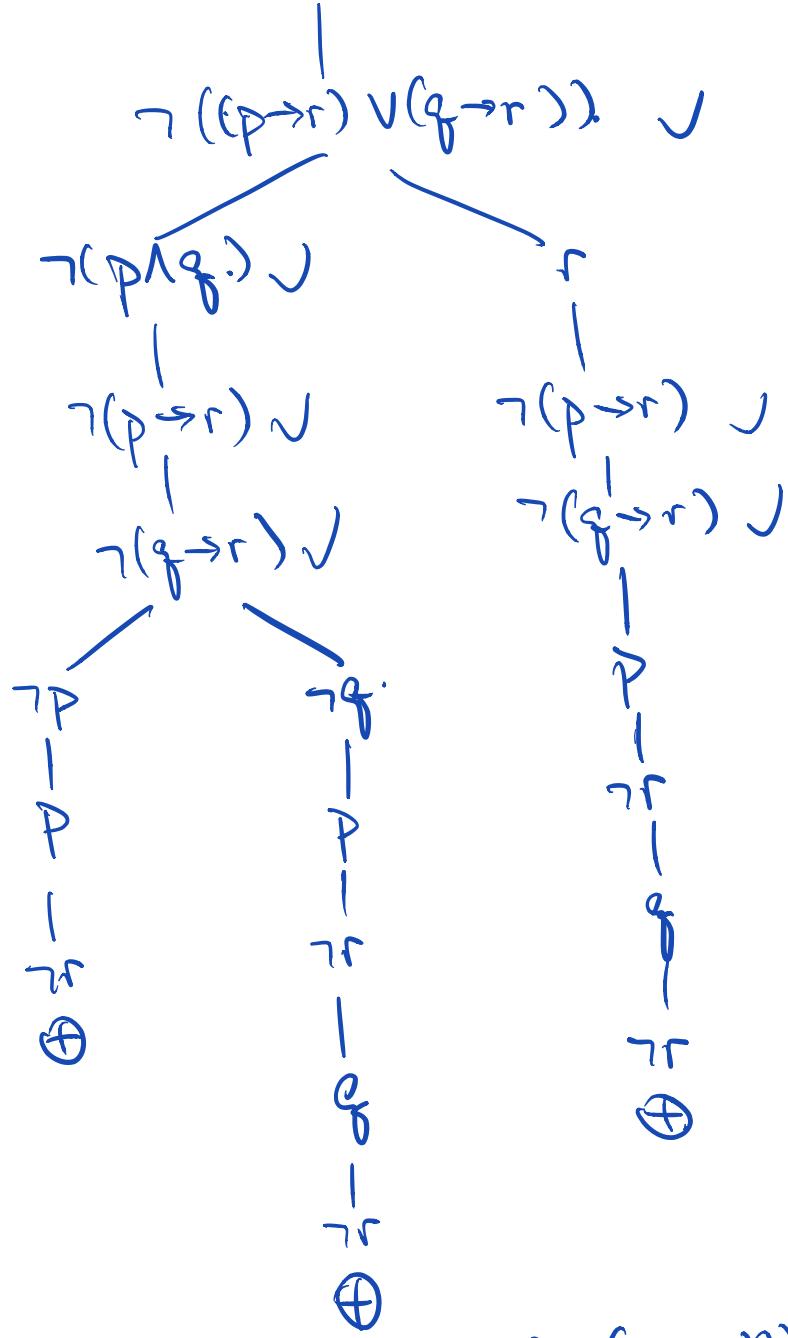
$\phi$  is valid  $\Leftrightarrow \neg\phi$  is not satisfiable.

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

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

is  $\neg(((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r)))$  not satisfiable?

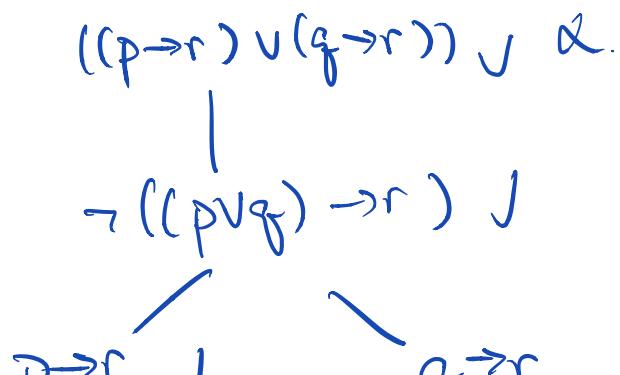
$$\begin{array}{c} | \\ ((p \wedge q) \rightarrow r) \quad \checkmark \end{array} \quad \alpha.$$

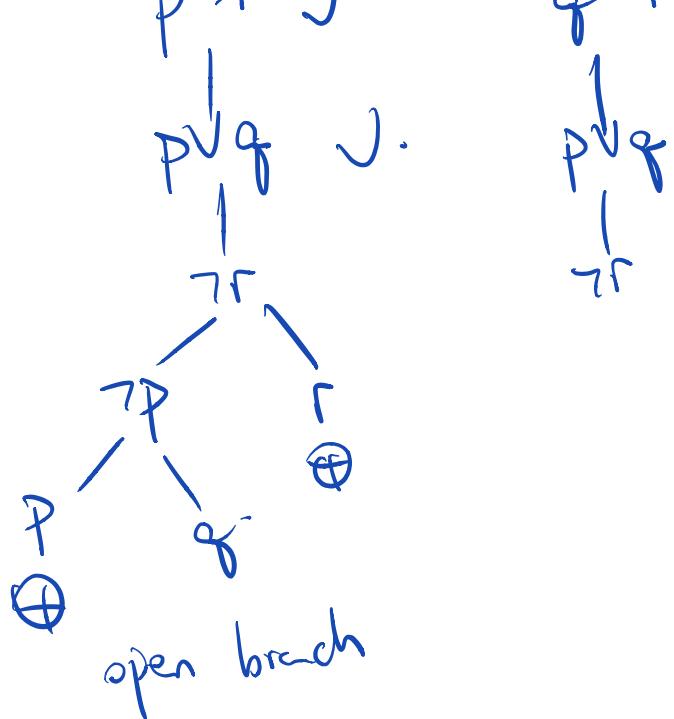


∴  $((\neg p \wedge \neg q) \rightarrow r) \rightarrow ((\neg p \rightarrow r) \vee (\neg q \rightarrow r))$   
     ... is valid.

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

↪ Is  $\neg((p \rightarrow r) \vee (q \rightarrow r)) \rightarrow ((p \vee q) \rightarrow r)$  satisfiable.





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

DNF

Disjunctive Normal form.

disjunctions of conjunction literals.

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

Each. fmla. has an equivalent fmla in DNF.

Conjunctive Normal Form.

$$(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r) \wedge \dots \wedge (x \vee y \vee z)$$

Converting to DNF.

truth table / logical equivalences / tautology

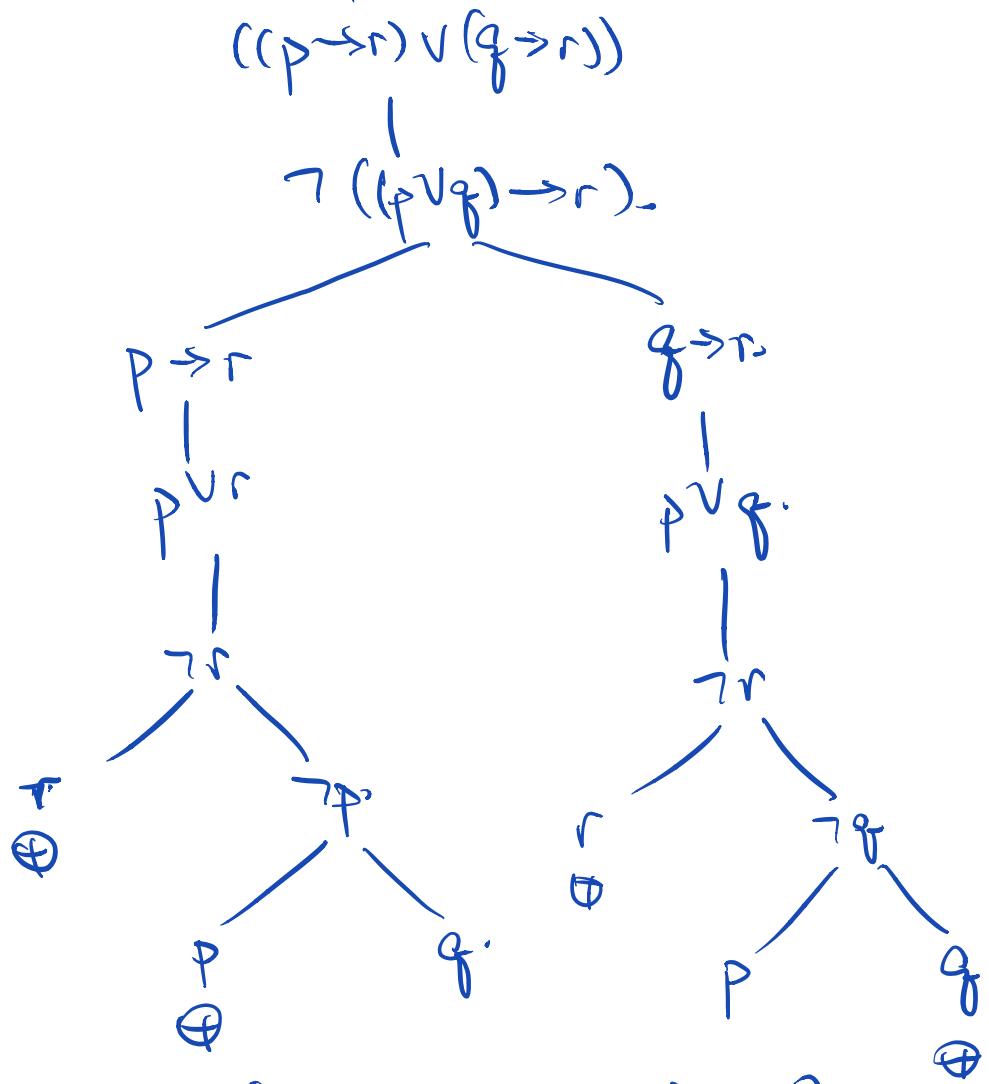
- $\phi$  at root of T.
- Expanded until T is completed.
- open branch  $\Theta$  of T let,

$$C_\Theta = \{ \text{literals in } \Theta \}.$$

then

$$\phi \equiv \bigvee C_\Theta$$

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



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

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

## First Order Tableaux

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

2.  $\beta$  remain the same.  
8 formulae.

choose a const.  $p_i$

add rule at each leaf below the node.

$$\begin{array}{c} \exists x \phi \quad \neg \forall x \phi \\ | \quad | \\ \phi(p/x). \quad \neg \phi(p/x) \\ \underbrace{\exists x \neg \phi \equiv \neg \forall x \phi.}_{\text{ }} \end{array}$$

$\gamma$  formula

pick any closed term t.  
don't tick node.

$$\begin{array}{c} \forall x \phi. \quad \neg \exists x \phi \\ | \quad | \\ \phi(t/x). \quad \neg \phi(t/x). \end{array}$$

until you used  
every available const/closed term

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

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

$\neg \exists y p(y).$  ~~X~~ (double negation first)  $\alpha$  rule

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

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

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

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

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

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

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

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

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

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

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

$$\forall x (G(x) \rightarrow H(x)) \wedge \forall x (H(x) \rightarrow F(x)) \wedge G(a) \wedge \neg \exists x (G(x) \wedge F(x))$$

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

$$\neg(G(a) \wedge F(a)) \quad \text{p.}$$

$$\exists y P(y). \quad \text{8.}$$

Ex.3.

$$(\forall x \neg P(x) \rightarrow \exists y P(y)) \text{ valid?}$$

$$\neg(\forall x \neg P(x) \rightarrow \exists y \neg P(y))$$

$$\forall x \neg P(x)$$

$$\exists y P(y). \quad \text{J}$$

$$\neg P(c).$$

$$\neg P(c).$$

$$\oplus$$

$$\text{Ex4. } \neg(\forall x (Gx \rightarrow Hx) \wedge \forall x (Hx \rightarrow Fx) \wedge Ga \wedge \\ \neg \exists x (Gx \wedge Fx) \text{ valid?}$$

$$\forall x (Gx \rightarrow Hx) \wedge \forall x (Hx \rightarrow Fx) \wedge Ga \wedge$$

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

$$|$$

$$\forall x (Gx \rightarrow Hx)$$

$$|$$

$$\forall x (Hx \rightarrow Fx)$$

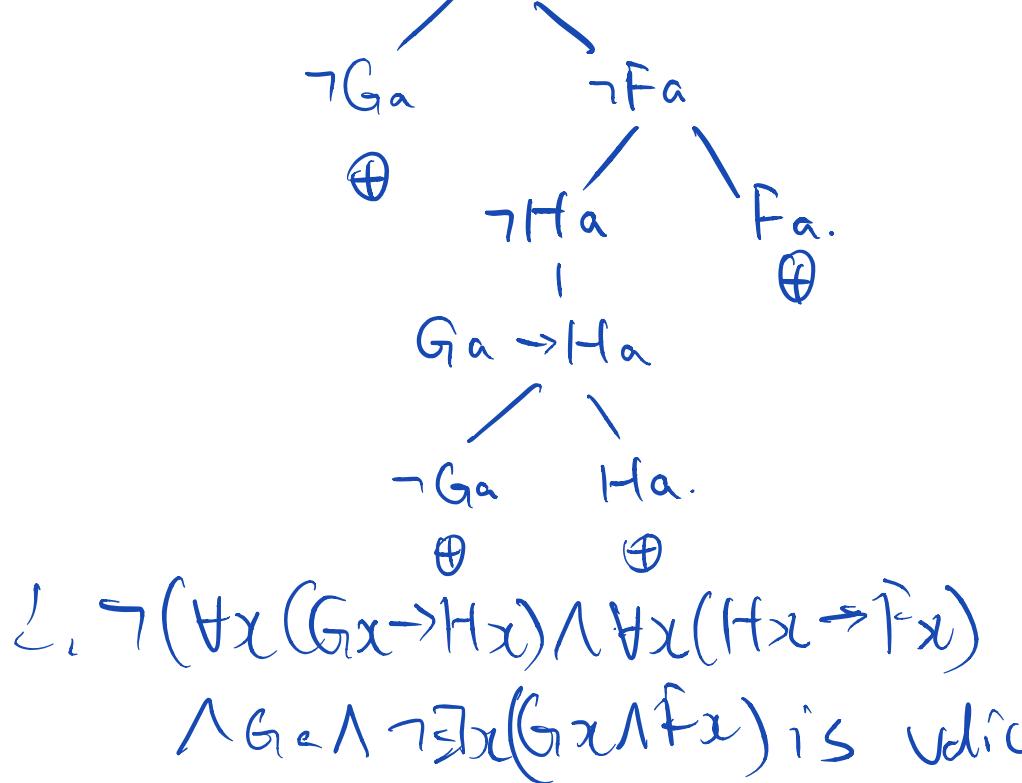
$$|$$

$$Ga.$$

$$|$$

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

$$\neg(Ga \wedge Fa) \quad \text{J.}$$



Infinite Tableau.

$$\text{Ex.5. } \neg(\forall x \neg g(x) \vee \exists x \forall y \neg(x < y))$$

for infinite  
tableau it's  
satisfiable.

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

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

$$I(<) = \{c, d, (d, e), \neg \forall y \neg(c < y) \vee \dots\}.$$

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

$$\neg \neg g(c) \vee$$

$$g(c)$$

$$\neg \neg (c < d) \vee$$

$$\begin{array}{c}
 c \in d. \\
 | \\
 \neg \forall y \neg (d \subset y). \\
 | \\
 \neg \forall (d \subset e) \vee \\
 | \\
 d \subset e. \\
 | \\
 |
 \end{array}$$

Alternative.

Tchkean as lists.

Recd	literal	$p, \neg p$
$\alpha$	$\phi_1 \wedge \phi_2$	$\neg(\phi_1 \vee \phi_2)$ $\neg(\phi_1 \rightarrow \phi_2), \neg \phi_1$
$\beta$	$\phi_1 \wedge \phi_2$	$\neg(\phi_1 \wedge \phi_2)$ $\phi_1 \rightarrow \phi_2$

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

while ! Tab.empty() S.

branch. —  $S = \text{Tab.dequeue}();$

(set). if  $S$  is fully expanded. and.  $S$  doesn't have

contradiction

## SATISFIABLE

else.

pick non-lit var  $\psi \in \Sigma$ .

switch ( $\psi$ )

case  $\alpha$ :

$$\Sigma = \Sigma[\alpha / f(\alpha_1, \alpha_2)]$$

make an  $\alpha$  expansion

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

if  $\Sigma$  doesn't have contradiction  
and  $\Sigma \notin \text{Tab}$ . enqueue  $\Sigma$ .

case  $\beta$ :

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

if — — . enqueue  $\Sigma_1$

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

— —  $\Sigma_2$ .

Output unsatisfiable (after all  $\Sigma$  seen).

if Predicate Tableau,

switch ( $\psi$ )

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

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

case  $\delta = \forall x \theta(x)$

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

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

$$\Sigma = \Sigma \cup (\theta(t)) \quad t \text{ for closed term}$$

$\neg \exists x \theta(x)$

$$\Sigma = \Sigma \cup \neg \exists x \theta(x) \quad -- --$$

If  $\Sigma \notin \text{Tab}$  and  $\neg \exists x \theta(x)$  enqueue  $\Sigma$ .

contradiction

## Proof of soundness (Induction)

Assume valuation  $v(\phi) = T$

$n = \text{number of iterations.}$

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

Assume  $n$  iteration  $\Sigma \in \text{Tab. } \theta \in \Sigma \rightarrow v(\theta) = T.$

for a new iteration  $\theta \in \Sigma \rightarrow v(\theta) = T.$

If  $\Sigma$  is defined and  $\psi \in \Sigma$  is picked.

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

if  $\psi$  is  $\alpha. v(\alpha_1) = v(\alpha_2) = T.$

$$\vdash \theta \in \Sigma \rightarrow v(\theta) = T.$$

if  $\psi$  is  $\beta. v(\beta_1) = T \text{ or } v(\beta_2) = T.$

$$\theta \in \Sigma_1 \rightarrow v(\theta) = T.$$

$$\text{or } \theta \in \Sigma_2 \rightarrow v(\theta) = T.$$

◻

## Soundness of Predicate Tableau

Similar approach.  $\Sigma \in \text{Tab}, S, A \models \Sigma$

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

$$\exists A' \exists x A.$$

$A'$  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  $\Sigma_{Tab}$  is degenerated  
and  $\Sigma_1, \Sigma_2$  are enlarged.  
 $\Sigma$  is parent of  $\Sigma_1, \Sigma_2$ .

$$P(\Sigma) = \Sigma'$$

$$P^0(\Sigma) = \Sigma$$

$$P^{n+1}(\Sigma) = P(P^n(\Sigma))$$

$\Sigma'$  is the ancestor of  $\Sigma'$  if  
 $n > 0$  and  $P^n(\Sigma) = \Sigma$

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

L.  $S'; A \models \Sigma$ .  
if  $\psi$  is  $\gamma$ .  $\psi = \forall x \theta(x)$ .  $S, A \models x \theta(x)$   
 $S, A \models \theta(t)$  for all closed term  
L.  $\forall x \theta(x)$  is replaced  $\theta(t)$  in  $\Sigma$   
 $S, A \models \Sigma$  still true.

Completeness of Prop. Tableau.

Proof: SATISFIABLE output  $\Rightarrow \Sigma_{Tab}$  degenerated.

Def.  $v(p) = T \Leftrightarrow p \in \Sigma$ . (anything that's included in  $\Sigma$ ).  
Prove by induction over  $n$  that

$$\begin{aligned} \psi \in P^n(\Sigma) \\ \Rightarrow v(\psi) = T. \end{aligned}$$

True for  $n=0$ .

$$\begin{aligned} \text{IHypothesis } \psi \in P^n(\Sigma) \\ \Rightarrow v(\psi) = T. \\ \theta \in P^{n+1}(\Sigma). \end{aligned}$$

Either  $\theta$  is in  $P^n(\Sigma)$   
or  $\theta$  is expanded in  
 $P^n(\Sigma)$   
if  $\theta$  is in  $P^n(\Sigma)$   
 $v(\theta) = T$ .  
if  $\theta$  is expanded by  $\alpha$ .

$$\begin{aligned} \theta = \beta. \\ \beta_1 \text{ or } \beta_2 \in P^n(\Sigma) \\ v(\beta_1) = T \text{ or } v(\beta_2) = T. \\ \therefore v(\theta) = v(\beta_1) \vee v(\beta_2) = T. \end{aligned}$$

$$\begin{aligned} \theta = \alpha \Rightarrow \alpha_1, \alpha_2 \in P^n(\Sigma) \\ \text{by IH } v(\alpha_1) = v(\alpha_2) = T. \\ \alpha_1, \alpha_2 \in P^n(\Sigma). \end{aligned}$$

$$\begin{aligned} \therefore v(\theta) = v(\alpha_1) \wedge v(\alpha_2) = T. \end{aligned}$$

Termination of propositional tableau does.

- \* When running tableau  $\phi$ , only new theories
- \* let  $X$  be a set of sub formulas of  $\phi$  <sup>are erased.</sup>
- and single negations of sub formulas  $\supset 2^{|\phi|}$  of these.
- \* a theory is a subset of  $X$ .  $\supset 2^{|\phi|}$  of these.
- \* Algo. stops in  $\supset 2^{|\phi|}$  steps at most

### Herbrand Structures:

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

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

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

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  $\infty$  const symbols,  
and no equality predicate

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

(some A)

## Fairness

Suppose you have  $P_1, \dots, P_k$  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 processes 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  $\phi$  never closes and expanded by a fair schedule. (can be infinite)  $\phi$  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.  $S_0, S_1, \dots, S_{\infty}$   
where  $S_n = P(\Sigma_{n+1})$ .

Let  $S = \bigcup_{n<\infty} S_n$ .  
 $\alpha \in S \Rightarrow \alpha \in S_n, \alpha \in S$   
 $\beta \in S \Rightarrow \beta \in S_n \text{ or } \beta \in S$   
 $\exists x \theta(x) \in S \Rightarrow \theta(t) \in S$ .

$\forall x \theta(x) \in S \Rightarrow \theta(t) \text{ for all closed terms } t$   
 $\neg \forall x \theta(x) \in S \Rightarrow \neg \theta(t) \text{ for all closed terms } t$   
 $\neg \exists x \theta(x) \in S \Rightarrow \neg \theta(t) \text{ for all closed terms } t$

Let  $H$  be Herbrand structure.

$D \setminus \{\text{closed terms of } S\}$

$I(t) = t$ .

$I(S^n) \rightarrow D^n(t_1, \dots, t_n)$

$\{t_0 \dots t_{k-1}\} \subseteq L(R) \Leftrightarrow R(t_0 \dots t_{k-1}) \in L$   
 Try to show  $\Theta \in S \Rightarrow H \models \Theta$  and  $\neg \Theta \in S \Rightarrow H \not\models \Theta$   
 atomic formula - base case  
 $\alpha_1 \in S, \alpha_2 \in S \Rightarrow H \models \alpha$ :  
 $\beta_1 \in S \text{ or } \beta_2 \in S \Rightarrow H \models \beta$ .  
 $\theta(p) \in S \Rightarrow H \models \exists x \theta(x)$   
 $p$  is a const.

$H, \theta(t) \in S \Rightarrow H \models Hx\theta(x)$   
 i.e.  $H \models \phi$ .

Eq. rules.

$A(t)$   
 $t = s \Rightarrow A(s)$ .

$A(t)$   
 $s = t \Rightarrow A(s)$ .

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

Other proof systems

Tableau — Tests satisfiability

Easy to implement

Axiomatic — Easy to define  
hard to use.

?? Resolution Theorem Provers. — Checks validity. Problem

Natural Deduction — Easy to read,

Truth-table — Exponential time.

Doesn't work for predicate logic

# Theorem Proving for Predicate Logic. Axiomatic.

Quantifier.

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

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

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

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

$$x = x.$$

$$(x = y) \rightarrow (y = x)$$

$$(x = y) \rightarrow (t(x) = t(y))$$

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

Sub.  $y$  for  $x$   
but must do so  
for all  $x$

Why don't we need

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

$$(x = y) \rightarrow (y = x) \quad \text{Ax.}$$

$$(y = x) \rightarrow (y = z \rightarrow x = z) \quad \underline{A(y = z)} \quad \text{Ax.}$$

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

Inference Rules

Modus Ponens

$$\frac{A, A \rightarrow B}{B}$$

Universal Generalisation.

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

$A(x)$  valid.

## Proofs

A proof of  $\phi$  is a finite set.

$$\phi_0, \phi_1, \phi_2, \dots, \phi_n = \phi.$$

$\phi_i$  is either axiom

or obtained from  $\phi_j$  and  $\phi_k$ .

where  $j, k < i$ .

by inference rules

(modus, ponens)

$$\vdash \phi$$

Proving from hypothesis.

(not consider  
all possible  
first order  
structures)

hypotheses are formulas which are valid in the type of formula 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

$\Gamma_g$  is a set of hypotheses  
gamma

$$\Gamma_g \vdash \phi.$$

$\phi_0, \dots, \phi_n = \phi$  proved based on  $\Gamma$

$\phi_i$  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  
 $z \rightarrow x$

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

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

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

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

$$1. \neg(x < x) \rightarrow \neg(x < y \wedge y < x),$$

$$\neg \exists \forall \neg (x < x) \quad (\text{Hypothesis})$$

$$2. \neg(x < x). \quad \text{Ax IV.}$$

$$3. \neg(x < y \wedge y < x). \quad \text{Modus Ponens.}$$

$$4. \forall x \forall y \neg(x < y \wedge y < x) \quad (\text{Universal Generalization})$$

Enfaldment

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

$$\Gamma \models \phi$$

if every model  $S$  of  $\Gamma$  is a model of  $\phi$

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

Strong Completeness.

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

hypotheses  $\vdash^S$  finite  $\Leftrightarrow$  All  $S$  that proves  $\Gamma$ .  
proves  $\phi$

Tableau Summary.

\* Tableau is sound and complete for first order logic.

Gödel's completeness theorem.

predicate logic

- \* Sound All provable  $\phi$  is valid.  $\Leftrightarrow$  If  $\neg\phi$  closes,  $\neg\phi$  is not satisfiable
- \* complete  $\phi$  is valid, it's provable.  
if  $\neg\phi$  is not satisfiable, tableau should close, if fair seg. is used

Recursive Lang (Turing decidable language)

A language  $L$  is set of strings over finite alphabets  $\Sigma$

$L$  is recursive. if  $\exists$  program.

determine if  $a \in \Sigma^*$ ,  $s \in L$  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 formulas is recursively enumerable.

Let  $\phi_0, \phi_1, \dots$  be an enumeration of formulas  
for ( $i=0, i < \infty, i++$ ).  $\Sigma$ .

Start tableau  $T_i$  with  $\phi_i$  at root.

for ( $j=0, j < i, j++$ )  $\Sigma$ .

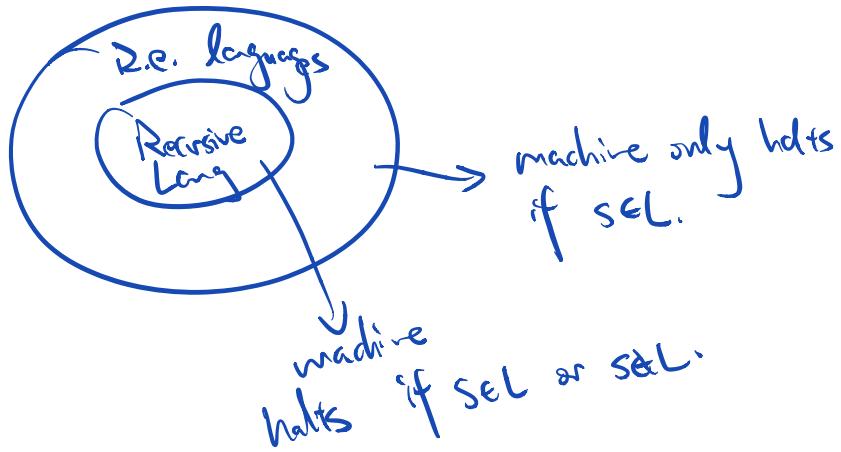
Expand  $T_j$  once with a fair schedule (queue)

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

}, }

program only outputs valid formula

any valid formula will get output (not guaranteed to terminate)



true statements of arithmetic isn't even r.e.

Gödel's incompleteness theorem

Proof: Gödel's Number

$$\forall x \ (x > 0)$$

1 2 3 2 4 5 6

$$G = 2^1 \times 3^2 \times 5^3 \times 7^4 \dots$$

Fixed Point Theory       $A(x) \quad B(x)$   
 $\text{Apply } ((A(x), B(x))) = f(A(B(x)))$

$$\begin{aligned} \forall F(x). \text{ Def } G(x) &= F(\text{apply}(x, x)) \\ A &= G((G(x))) = F(\text{apply}(|G(x)|, |G(x)|)) \\ &= F(|G|G(x)|) \\ &= F(|A|) \end{aligned}$$

for all  $F(x)$

$\exists A$  as a fixed point.

provable(n) n as a Gödel number.

formula describing provable(n) is PROVABLE(FN)

with fixed-point theory, unprovable( $\neg A$ ) = A.  
Cycle.

proved A is formula that's unprovable.

1. Arithmetic system is not complete.

Proving from Assumptions:

WTS.  $\phi$  is valid in a model or a type of model.

Write  $\Sigma$  assumptions set.

\* Add any  $\phi \in \Sigma$  at the leaf of  $T$  at any time.

\* if prove  $\Sigma \vdash \phi$ .

$$\Sigma = \{ \forall x \forall y (x < y \rightarrow x + 1 < y) \}$$

$$\forall x \forall y (\neg(x < y \vee y < x \vee x = y))$$

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

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

$$\neg \forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z) \quad \perp$$

$$\neg \forall y \forall z ((y < z \wedge z < y) \rightarrow y = z) \quad \perp$$

| 8

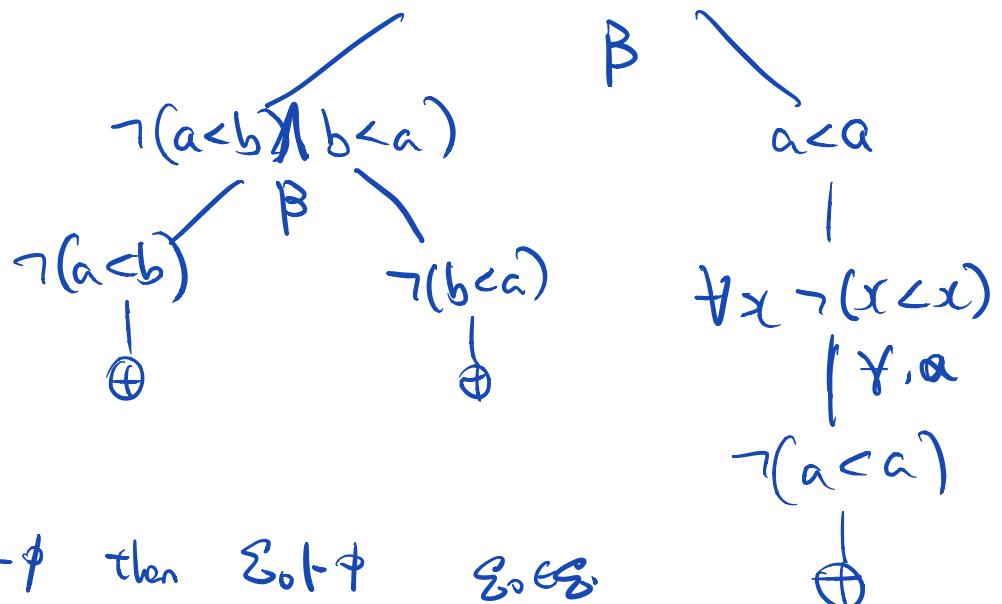
$(a < b \wedge b < a) \vdash$

$$\begin{array}{c} | \\ a < b \\ | \\ b < a. \\ | \end{array}$$

$\forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z)$   
 $| \gamma, a$

$\forall y \forall z ((a < y \wedge y < z) \rightarrow a < z)$   
 $| \gamma, b$

$| \gamma, a$   
 $((a < b \wedge b < a) \rightarrow a < a)$



Compactness.

if  $\Sigma \vdash \phi$  then  $\exists \text{ finite } \Sigma_0 \text{ s.t. } \Sigma_0 \vdash \phi$

Inconsistency.

$\Sigma$  is inconsistent if  $\vdash \perp$

by compactness if  $\Sigma$  is inconsistent  
 $\Sigma_0$  is inconsistent

By strong completeness consistent has a model

compactness if every finite subset of  $\Sigma$  has a model, there is a model for  $\Sigma$

FOL cannot have Connectedness (graph)

$$\begin{array}{ll} L & C = \{c, d\} \\ & F = \emptyset \quad \text{Edge.} \\ & D = \{=, E\} \end{array}$$

Assume  $G \models \exists \underline{\beta} \rightarrow G$  is connected.

$\downarrow$   
a set of hypotheses that ensures  
a graph.

$$\text{Let } \phi_1(x, y) = E(x, y)$$

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

$\phi_k$  means there is a connection length  
of  $k$ .

Consider  $\exists U(\phi_1(c, d), \neg \phi_2(c, d) \dots)$ .

a finite subset would certainly have  
a gap  $\exists U(\neg \phi_1 \dots \neg \phi_k)$

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

$$G \models \exists U(\neg \phi_n(c, d) : n = 1, 2, 3 \dots)$$

$G$  is connected, but no path  $c \text{---} d$   
 $\exists : \neg \phi_n(c, d)$

FOL has limitation in Expressiveness

Compactness theorem and non-standard analysis

$\Sigma = \{ \text{all valid statements of } N \}$   
e.g.  $2+2=4 \in \Sigma$

$\Sigma^* = \Sigma \cup \{ w \neq 0, w \neq 1, \dots, w \neq n, \dots \}$

Every finite subset of  $\Sigma^*$  has a model.  
 $w \in n+1$

i.  $\Sigma^*$  has a model.

Similarly if  $\Sigma = \{ \text{all valid statements of } R \}$

$\Sigma^*$  has a model. ( $\alpha$ )

↑  
infinitely  
large real.

i.  $\frac{1}{\alpha}$  is a infinitely small  
every real number + real number.  
(ordinary one)

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

↑  
standard  
real ↑  
Infinitary

i.  $f(x) = St\left(\frac{f(x+\delta x) - f(x)}{\delta x}\right)$  small  
real.

## Paradoxes

- Liar Paradox.

Russell set

- Russell's Paradox.

$R = \{ S : S \notin S \}$  (Barber)

$R \in R$

- 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.

$$\begin{aligned}L &\Rightarrow C = \{0, 1, 2, \dots\} \\F &= \{+, \times\} \\P &= \{=, <\}.\end{aligned}$$

Theorem (Gödel).

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

Gödel Number is a unique number  
assigned to every formula.

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

$$\begin{aligned}G_1(\phi) &= n. & G(\bar{\phi}) &= m \\F(n) &= \phi & \uparrow & \text{a proof, a.k.a a list} \\P(m) &= \phi & & \text{of } \phi.\end{aligned}$$

Proof.

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

$\lambda(n) = F(n)$  is a formula with one var  $x$ .

Let.

$A_0(x) \dots A_n(x) \dots$

be an enum. of formula with variable  $x$ .

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

Consider formula.

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

this is a formula with one free var  $x$

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

$\text{NF } A_{n_0}(x)$  iff. there is no proof of  $Ax(x)$

$\text{NF } Ax(x)$ .

$\exists A_{n_0}(n_0)$

$\text{NF } A_{n_0}(n_0)$  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-

Grammer.

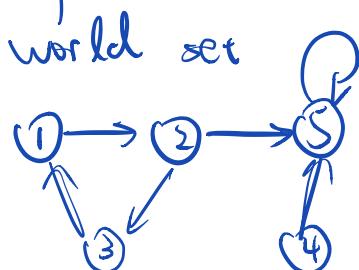
$$\text{prop} ::= p \mid q \mid r \mid \dots$$

$$\phi ::= \text{prop} \mid \neg \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid p \rightarrow \phi \mid \Diamond \phi \mid \Box \phi$$

$$\Diamond g \rightarrow \Box \Diamond p$$

Frames

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



Valuation.

$$V: \text{prop} \rightarrow P(W). \quad P(W).$$

$V(p) = \{1, 3, 5\}$ . Power Set of  $W$ .

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

valuations

Frame with Valuation = Model.  
 let  $w \in W$ , ( $w$  is a world)

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

$$M, w \models \neg \phi \Leftrightarrow M, w \not\models \phi.$$

$$M = (F, V) \downarrow$$

$$= (W, R, V)$$

set of worlds relations

$$M, w \models \phi \wedge \theta \Leftrightarrow M, w \models \phi \text{ and } M, w \models \theta$$

$$M, w \models \Diamond \phi \Leftrightarrow \exists w_0 \in W \quad (w, w_0) \in R \quad M, w_0 \models \phi.$$

$$M, w \models \Box \phi \Leftrightarrow \forall w_0 \in W \quad \text{if } (w, w_0) \in R \text{ then } M, w_0 \models \phi$$

Validity:

$$M = (W, R, V).$$

$M \models \phi \Rightarrow$  for all  $w \in W \quad w \models \phi.$

Valid in a Frame

$W, R \models \phi$     all valuations  $V \quad W, R, V \models \phi.$   
 $F \models \phi.$

Valid over a class of frames.

$K \models \phi \Leftrightarrow FCK, \quad F \models \phi.$

Valid formulas.

$$\Box \Diamond p \rightarrow \Box \Diamond p.$$

$$\Box(p \wedge q) \rightarrow (\Box p \wedge \Box q)$$

$$\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$$

Not valid.

$$\Box(p \vee q) \rightarrow (\Diamond p \vee \Diamond q)$$

Classes of Frames.

$\phi$  defines  $K$     " $F \models \phi \Leftrightarrow FCK$ "

Reflexive     $\forall w R(w, w).$      $\Box p \rightarrow p.$

Transitive     $\forall u \forall v \forall w (Ruv \wedge Rvw \rightarrow Ruw), \quad \Box p \rightarrow \Box \Box p$

Symmetric     $\forall u \forall v Ruv \rightarrow Rvu \quad p \rightarrow \Box \Diamond p$

Dense     $\forall u \forall v Ruv \rightarrow \exists w (Ruw \wedge Rvw)$

$$\Box \Diamond p \rightarrow \Diamond p$$

Labelled Frame

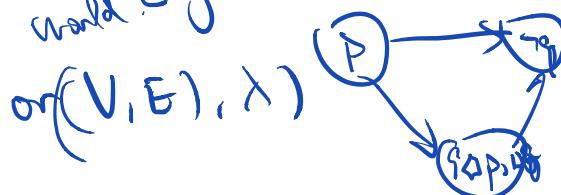
Each world is labelled by a set of formulas.

$$((w, E), \lambda)$$

↑  
World Edges.

$$\text{or}(V, E), \lambda$$

PowerSet  
 $\lambda: W \rightarrow \wp(\text{formulas}).$



## Modal Tableau

check satisfiability of  $\Phi$ , make a queue of labelled frames.

one free in queue, one word  $\{w\}$   $\lambda(w) \in \{*\}$

while not empty  $Q_i$ , dequeue a word,  $E \rightarrow$

$\dots (\omega, \dot{\theta}), \lambda \rangle$

else. i. if there are only  $\square$ ,  $\neg \square$   
and blocks in range  $\phi$   
 $\rightarrow$  YES, model found.

else pick new and  $\theta^*$  such  
that it's non-linear not a  $\square$   
not a  $\neg \square$ .

if  $\theta$  is a.

$$\delta'(w) := \delta(w) \setminus \{\theta\} \cup \{a_1, a_2\}$$

for every  $\lambda^i(v) \rightarrow \lambda(v)$ ,  $v \neq w$ . remain  
else see

$\text{Eng}_> ((V, E) \lambda')$

If  $\theta$  is  $\beta$

$$\lambda_1(\omega) = \chi(\omega) \setminus \{\theta\} \cup \{\beta_1\}$$

$$\lambda_2(w) = \lambda(w) \setminus \{\theta\} \cup \{\beta^2\}$$

eng.  $((V, E, \lambda_1), (U, E, \lambda_\Sigma))$

if  $\theta$  is OA or  $\pi - \theta$

let  $V' = V \cup \{v\}$ .  
 $E' = E \cup \{(w, v)\}$ .

$\lambda'$  be the same as  $\lambda$  except  $\lambda'(v) = \{A\}$   
 $\cup \{B: \Box B \in \lambda(w)\}$   
 $\cup \{\neg B: \neg \Box B \in \lambda(w)\}$

Eng.  $((V', E'), \lambda')$

if q.empty() return NOT satisfiable

Tableau for Reflexive frms

$E = \{(w, w)\}$  Initially

when new world  $v$  is added.  
 add  $(v, v)$ .

$\Box A \in \lambda(w) \rightarrow A \in \lambda(w)$ .

$\neg \Box A \in \lambda(w) \rightarrow \neg A \in \lambda(w)$

$\vdash$

Symmetric frms

when adding edge  $(w, v)$

add  $(v, w)$

Anything box or negated diamonds in  $v$   
 goes back to  $w$

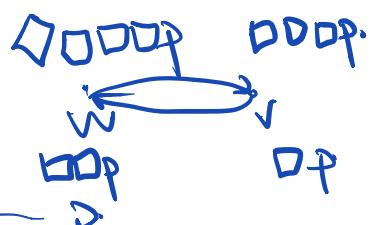
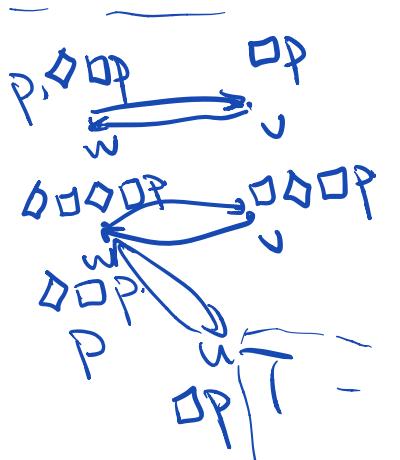
transitive frms.

When adding  $v$  and edge  $(w, v)$

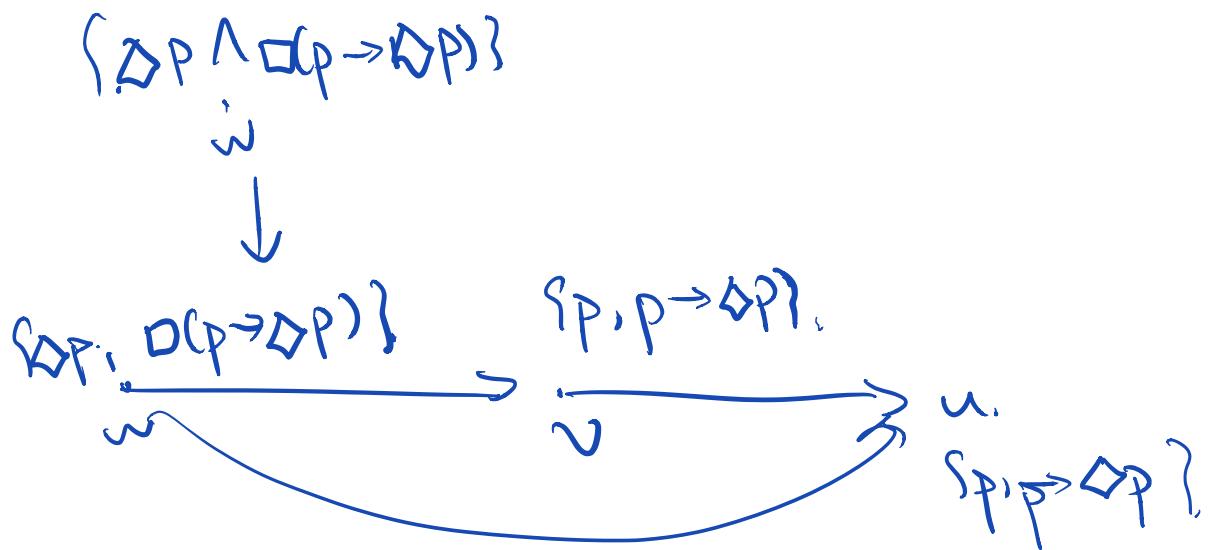
add  $(w, v)$ . forall  $(u, w) \in E$ .

include  $A$  in  $\lambda(v)$  whenever  $\Box A \in \lambda(u)$

$\neg A$



(May not terminate in transitive frames)



Frame and node morphism

let  $(W, R)$  be frame.

p-morphism  $f: W \rightarrow W'$ .  $R \rightarrow R'$  st.

$\forall v, w \in W$  if  $(v, w) \in R$

$(f(v), f(w)) \in R'$

connection remains

$\forall v \in W$  and all  $u \in W'$ .

if  $(f(v), u) \in R'$

$\exists f(w) = u$   $(v, w) \in R$ .

pairing remains

(no more of them)

Let a model (frame

$N = (N, <)$

reflexive.

$$F = (\{x\}, \{x, x\}).$$

Not reflexive  $f: N \rightarrow \{x\}$ .

$f(n)=x$  is a  $\varphi$ -morphism.

extreme case.

$F$  is single-point reflexive

$N$  is not reflexive?

Contradicts.

s.t. modal logic cannot define  
irreflexive

## Temporal Logic.

$$\text{frla} := \text{prop} \mid \neg \phi \mid (\phi \vee \psi) \mid F\phi \mid P\phi,$$

$F\phi$  will be true at some point in future.

$P\phi$  was true at some time in past.

$G\phi \Leftrightarrow \neg F\neg\phi$ .  $\neg\phi$  will not be true  
in the future.

$\phi$  will be true  
in the future.

$H\phi \Leftrightarrow \neg P\neg\phi$ .  $\phi$  was always true

Basic temporal axioms:

$$G(p \rightarrow q) \rightarrow (Gp \rightarrow Gq).$$

$$H(p \rightarrow q) \rightarrow (\neg p \rightarrow Hq).$$

$P \rightarrow GP$

$P \rightarrow HF P$

and Ax. from prop. logic

Assume Time is Linear.

K<sub>t</sub>.

$GGP \rightarrow GP$

$F_P \wedge F_Q \rightarrow (F(P \wedge F_Q) \vee F(P \wedge Q) \vee F(Q \wedge F_P))$

Multi-modal logic.

$\Diamond_i \phi \quad \Box_i \phi \quad i = 0, 1, 2, \dots, k-1$ .

multimodal frame.

$(V, R_0, R_1, \dots, R_{k-1}) \quad R_i \subseteq V \times V$ .

$(V, R_0, \dots, R_{k-1}), v, x \models \Diamond_i \phi \Leftrightarrow$

$\exists y \in V \forall (x, y) \in R_i \wedge (V, R_0, \dots, R_{k-1}), v, y \models \phi$

Propositional Dynamic Language.

$P = \{ p_0, p_1, \dots \}$ .

Atomic Programs  $\Pi = \{ \pi_0, \dots \}$ .

Program  $\pi' := \pi_i \| (\pi; p) \| (\pi + p) \| \pi^* \| \phi$ .

$\pi, p$  are programs,  $\phi$  is a formula.

$f \vdash \phi \iff \vdash f \rightarrow \phi$

Ind  $\psi := p \wedge \psi_1 \mid \psi_1 \wedge \psi_2 \mid \langle \pi \rangle \psi_1 \mid L^\pi \psi$ .

if  $p$  then  $\pi$  else  $\pi'$ .

$(p?; \pi \vdash ((\neg p)?); \pi)$ .

While  $p$  do  $g$ .

$((p?; \pi)^*; (\neg p)?)$

Ex.

$F, v, x \models p \wedge g \wedge [\pi_0^*] (p \vee (g \wedge \langle \pi_1 \rangle (\neg p \wedge \neg g)))$

