Logic

3 components (language formalising arguments)

- syntax. – formula.
- semantics – explain / understand formula.
- inference rules

Classical principles

$\vdash$ ← define.

$\vdash \neg(A \wedge \neg A)$ → $\underline{A \wedge \neg A}$ false

$\vdash A \vee \neg A$. true

A logic is inconsistent if $A \wedge \neg A$.

Set theory

Set can be finite / infinite

set with property P. $\{x \mid P(x)\}$.

$x \in Y$. object x is a member of Y.

$X \subseteq Y$ set X is a subset of Y.

Russell's Paradox.

$R = \{x \mid x \notin x\}$

If $R \in R$ then $R \notin R$.

If $R \notin R$ then $R \in R$

Given P and input I, whether P holds on I. is undecidable

logic verification.

$\{A\}\, C\, \{B\}$. if C run in state A. It end with state B

Propositional logic

$\neg$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
	conjuction	disjunction.		equivalence.
not	and	or	implies	iff.

propositions are smallest factual. statements

connectives

$\neg$ has a greater precedence.

e.g. $\neg A \vee B = (\neg A) \vee B$

Valuations

$v: L \to \{0, 1\}$.

function that interprets proposition letters to 0 (false) or 1 (true).

$\neg A$ is true iff A is false.

$A \wedge B$ iff A and B.

$A \vee B$ iff A or B.

$A \to B$ if A then B.

$A \to B = \neg A \vee B$

$\neg(A \to B) = A \wedge \neg B$

$v \models A$. A is true with valuation v.

$v \not\models A$ — — false — —

A satisfiable. if $v \models A$ for some v.

A valid. $v \models A$ for all v.

A is invalid $v \models A$ not for all v.

iff. $\neg A$ is satisfiable

A is unsatisfiable iff $\neg A$ is valid.

Easy way to tell whether valid/satisfiable

truth table.

$(A \wedge \neg A) \to B$

$A \wedge \neg A$	B	$A \wedge \neg A \to B$
0	0	1
0	1	1

not satisfiable

valid.

Paradoxes of material implication.

$B \to (A \to B)$. valid.

$\neg A \to (A \to B)$ valid

$(P \to Q) \vee (Q \to R)$.

$Q = 1$, $P \to Q$ valid

$Q = 0$ $Q \to R$ valid.

Minimal set of connectives.

$\neg$ $\to$.

$A \vee B = \neg A \to B$.

$A \wedge B = \neg(A \to \neg B)$

$A \leftrightarrow B = (A \to B) \wedge (B \to A)$

$= \neg((A \to B) \to \neg(B \to A))$.

DNF.

Disjunctive $L ::= P \mid \neg P$ (literal)

Normal Form.

$C ::= L \mid L \wedge L$ (conjuction)

$D ::= (C) \mid D \vee D$ ("DNF").

every formula can be converted to DNF

Applications of DNF.

$A \to A_{DNF} = C_0 \vee C_1 \cdots \vee C_{n-1}$.

A satisfiable iff C_k is satisfiable

$C_k = L_{k,0} \wedge L_{k,1} \cdots$

C_k is satisfiable iff not contain P and ¬P

decides satisfiability

Also used to decide invalidity A is valid iff ¬A is unsatisfiable.

Get DNF.

Example $P \leftrightarrow (\neg Q \to R)$.

Truth table

P	Q	R	$\neg Q \to R$	$P \leftrightarrow (\neg Q \to R)$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
1	0	0	0	0
0	1	1	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

$$(\neg P \wedge \neg Q \wedge \neg R) \vee (P \wedge \neg Q \wedge R) \vee (P \wedge Q \wedge \neg R) \vee (P \wedge Q \wedge R)$$

Using logic equivalences

$$P \leftrightarrow (\neg Q \to R) = P \leftrightarrow (Q \vee R) = (P \to (Q \vee R)) \wedge ((Q \vee R) \to P)$$

$$= (\neg P \vee (Q \vee R)) \wedge (\neg (Q \vee R) \vee P)$$

$$= [(\neg P \vee (Q \vee R)) \wedge (\neg (Q \vee R))] \vee$$
$$[(\neg P \vee (Q \vee R)) \wedge P]$$

$$= [(\neg P \vee (Q \vee R)) \wedge (\neg Q \wedge \neg R)] \vee$$
$$[(\neg P \vee (Q \vee R)) \wedge P]$$

$$= ((\neg Q \wedge \neg R) \wedge \neg P) \vee (\cancel{(\neg Q \wedge \neg R) \wedge (Q \vee R)})$$
$$\vee (\cancel{\neg P \wedge P}) \vee ((Q \vee R) \wedge P))$$

$= (\neg P \wedge \neg Q \wedge \neg R) \vee ((P \wedge R) \vee (P \wedge Q))$

$= (\neg P \wedge \neg Q \wedge \neg R) \vee (P \wedge Q \wedge R)$
$\quad \vee (P \wedge \neg Q \wedge R)$
$\quad \vee (P \wedge Q \wedge \neg R)$

CNF
Conjuctive normal form.

$L ::= P \mid \neg P$
$D ::= L \mid L \vee L$.
$C ::= (D) \mid C \wedge C$.

Every formula $\rightarrow$ equivalent CNF.
CNF useful for proving.
A is valid iff $\forall C_i$ is valid.

Boolean Algebra
Propositional logic equivalent.

$(B, 0, 1, +, -)$.

B is non-empty set, $0, 1 \in B$.

$+ : B \times B \rightarrow B$ binary function.
$- : B \rightarrow B$ unary function.

for $+$:

$(a+b)+c = a+(b+c)$ Assoc.
$a+b = b+a$ Commutavity.
$a+a = a$. idempotency
$0+a = a$. zero law

for $-$:

$--a = a$.
$a + \bar{a} = 1$.
$-1 = 0$

Distribution $a \cdot (b+c) = a \cdot b + a \cdot c$.

$\{\{0,1\}, 0, 1, + -\}$.

$+ \rightarrow$ disjunction $(\vee)$ or.
$- \rightarrow$ negation
$\cdot \rightarrow$ Conjunction $(\wedge)$.

Another Example of Boolean Algebra.

Power Set $\quad Pow(X) = \{Y \mid Y \subseteq X\}$.

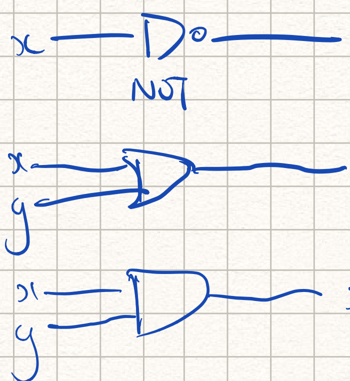
$$Y \in Pow(X) \iff Y \subseteq X$$

$$\langle Pow(X), \phi, X, \cup, | \rangle$$
$$0, \; 1, \; +, \; -$$

Boolean set algebra.

Stone's Theorem.

Boolean algebra. $\cong$ Boolean set algebra.



$\bar{x}$.

$x \vee y \,/\, x+y$.

$x \wedge y \,/\, x \cdot y$.

De Morgan's Laws

$$-(a+b) = \bar{a} \cdot \bar{b}. \qquad \neg(A \vee B) = \neg A \wedge \neg B$$
$$-(a \cdot b) = \bar{a} + \bar{b}.$$

$$-(a+b) = -(\bar{\bar{a}} + \bar{\bar{b}}) = \bar{a} \cdot \bar{b}.$$
$$-(a \cdot b) = -(\underbrace{-(\bar{a}+\bar{b})}_{\text{def}}) = \bar{a} + \bar{b}.$$

def

$$a \cdot 0 = -(\bar{a} + 1 = -1 = 0$$

$$a + 1 = (\bar{\bar{a}} + \bar{\bar{1}}) = -\bar{a} \cdot \bar{1} = -\bar{a} \cdot 0 = -0 = 1.$$

Duality $\quad$ if $A = B \quad A^* = B^*$

swap $+/\cdot$, $0/1$.

$$a \cdot (b+c) = a \cdot b + a \cdot c$$

$a+(b\cdot c) = (a+b)\cdot(a+c)$

Sum of Products (DNF)

$$\neg(\overline{a}+b) \rightarrow \overline{a}\cdot\overline{b}$$
$$\neg(a\cdot b) \rightarrow \overline{a}+\overline{b}$$
$$\neg\neg a \rightarrow a.$$

$$(a+b)\cdot(c+d) = a\cdot c + a\cdot d + b\cdot c + b\cdot d.$$

$$(A\wedge C)\vee(A\wedge D)\vee(B\wedge C)\vee(B\wedge D)$$

Propositional logic / Boolean algebra

not very expressive

no internal structure for propositions

↙

first-order logic.

propositions → predicate formulas

$P(t_1, \dots, t_n)$

$t_i \rightarrow$ terms from variables and function symbols.

two quantifiers $\forall x$ $\exists x$

$\forall x.\ \mathrm{Man}(x) \rightarrow \mathrm{Mortal}(x).$

predicates / adj

terms

predicate being T/F depends on their arguments

explain by
1. terms (arguments)
2. predicate symbols

A model of FOL is a interpretation for all of its non-logical symbols

Assume inf set of variables $V = \{x, y, z, \dots$

first order language Σ defined by.

1. A set of function symbols.

$F = \{f_1, f_2 \cdots\}$. each with an arity $k \geq 0$

constant symbol $f()$

2. A set of predicate symbols

$P = \{P_1, P_2 \cdots\}$. predicate symbols, each. with associated arity

A proposition letter is a predicate symbol of arity 0.

$\Sigma = (F, P)$.

Peano arithmetic.

$$\Sigma_{PA} =_{def} (\{\underbrace{0}_{\text{constant}}, s, +, \times\}, \{=, <\})$$

function ... predicate

$+(x, y) \Rightarrow$ or infix $x+y$

Syntax of FOL: terms

Terms : 1. a variable $x \in V$.

2. a function symbol in Σ of arity k.

$f(t_1 \cdots t_k)$ is a term.

NB: predicate symbols are not allowed in terms

Formulas of a FOL Σ is give by.

$$\begin{aligned} A ::= \; & P(t_1 \cdots t_k) \quad \text{(predicate formulas)} \\ & \mid \neg A \\ & \mid A \wedge A \\ & \mid A \vee A \\ & \mid A \to A \\ & \mid A \leftrightarrow A \end{aligned}$$

| $\forall x\,(A)$
| $\exists x\,(A)$

P are predicate symbols in $\mathcal{S}$ with arity $k \geq 0$
$t_1 \cdots t_k$ are terms of Σ

Bound. / free variables

bound of a variable x in a formula A
is within the scope of a quantifier $\forall x$, $\exists x$.

Otherwise the occurence is free.

$$\underbrace{\forall y\,(\underbrace{\exists x(x = s(y))}_{\text{bound } x} \vee x + y = 0)}_{\text{bound } y}$$

A formula with no free variables is a sentence

Syntax Sugar.

$\forall x.\, A = \forall x\,(A)$.

| ¬.
| ∨ and ∧
↓ → and ⟺

There is a haggis larger than any other.

$\exists x.\ Haggis(x) \wedge$.
$\forall y\,(\neg(y = x) \wedge Haggis(y) \rightarrow LargerThan(x, y))$

First order logic Semantics.

Interpret $\mathcal{S} = (F, P)$ in first order structure $\underline{M}$.

D — domain.

$f \rightarrow f^{n}$ ⇐ interpreted in domain D.

$P \rightarrow P^{M}$ ⇐ relations

first order structure.

$M = (D, F^M, P^M)$

Example. $\Sigma_{PA} =_{def} (\{0\ s, +, \times\}, \{=, <\})$

first order structure. $M =_{def} (\mathbb{N}, \{0^M, s^M, +^M, \times^M\}, \{=^M, <^M\})$

domain

first order structure.

$M = \{D, F^M, P^M\}$

Non-empty set.

Valuation. is a function. interpreting variable in $\mathcal{V}$ as an element in the domain. $\rho : V \to D$

$\rho(f(t_1 \cdots t_k)) =_{def} f^M(\rho(t_1), \cdots \rho(t_k))$

$$\rho[x \mapsto d](y) =_{def} \begin{cases} d & \text{if } x = y \\ \rho(y) & \text{otherwise} \end{cases}$$

$\rho[x \mapsto d]$ is like ρ, for x it maps to d.

Satisfaction relation.

give FO, Σ, FO structe M, and valuation ρ

$M \models_\rho A$ $\to$ A is satisfied in M under ρ.

sentence.

$M \models_\rho P(t_1 \cdots t_k) \Leftrightarrow (\rho(t_1) \cdots \rho(t_k)) \in P^M$

$M \models_\rho x = y \times z \Leftrightarrow (\rho(x), \rho(y \times z)) \in =^M$

$M \models_\rho \neg A \Leftrightarrow M \not\models_\rho A$.

$M \models_\rho \forall x (A) \Leftrightarrow M \models_{\rho[x \mapsto d]} A \ \forall d \in D$

$M \models_\rho \exists x (A) \Leftrightarrow M \models_{\rho[x \mapsto d]} A$ for some $d \in D$

Example $M \models_\rho \exists x.\ 0 < x$.

$M \models_{\rho[x \mapsto d]} 0 < x$ for some $d \in \mathbb{N}$.

$(\rho[x \mapsto d](0), \rho[x \mapsto d](x)) \in <^M$ for some $d \in \mathbb{N}$

$(0^M, d) \in <^M$ for some $d \in \mathbb{N}$

$0^M < d$ for some $d \in \mathbb{N}$.

TRUE

A valid in M $\quad M \models A$ (with all possible ρ) $\quad M \models_\rho A$.

just valid $\forall M$ $\quad \models A$.

A valid in M for certain ρ $\quad M \models_\rho A$

just satisfiable if satisfiable in some M

Definition of $M \models_\rho A$.

$$M \models_\rho P(t_1 \cdots t_k) \Leftrightarrow (\rho(t_1) \cdots \rho(t_k)) \in P^M$$
$$M \models_\rho \neg A \Leftrightarrow M \not\models_\rho A$$

$$M \models_\rho \forall x (A) \Leftrightarrow M \models_{\rho[x \mapsto d]} A \text{ for all } d \in D$$
$$\exists \quad \cdots \quad \text{for some } d \in D$$

Example $\quad L \models_\rho \forall x \; Blues(x) \to x \triangleright y$. dependent on valuation ρ.

$L \models_\rho \forall x \; V(x) \to \exists y (x \sim y \land \neg V(y))$

$\Leftrightarrow$ for all songs s, if $s \in V^L$ $\; L \models_{\rho[x \mapsto s][y \mapsto s']} x \sim y \land \neg V(y)$ for some songs s'.

Whether it is true depends on L not ρ

If A is a formula $A[t/x]$ means substitute all free x with t.

$(\exists x. P(x,y) \land Q(y, x)) [t/x] = \cdots$ remain same

$(\exists x. P(x,y)) \land Q(y,x)) [t/x] = \exists x (P(x,y)) \land Q(y, t)$

$(\exists x. x \neq y)[x/y]$ Wrong!

$\Downarrow$

$\exists x \; x \neq x$ $\quad$ free variable capture.

$\Uparrow$

Not what we wanted

FOL Equivalences

$A \equiv B$ (equivalent if for any M and ρ) $\quad M \models_\rho A \Leftrightarrow M \models_\rho B$

$$\forall x A \equiv \forall z (A[z/x])$$
$$\exists x A \equiv \exists z (A[z/x])$$

$$\neg \forall x A \equiv \exists x \neg A$$
$$\neg \exists x A \equiv \forall x \neg A$$
$$\forall x (A \land B) \equiv \forall x A \land \forall x B$$
$$\exists x (A \lor B) \equiv \exists x A \lor \exists x B.$$

Prenex formulas

Prenex formulas are in special form.

$Q_0 x_0 \, Q_1 x_1 \cdots A$

Q_i is either a $\exists$ or $\forall$.

A is a quantifier free formula.

Every formula can be converted to Prenex formula.

Example.

1. $\forall x(Jx \to (\exists y(x \sim y \wedge \neg J(y))))$
$= \forall x(\neg Jx \vee (\exists y(x \sim y) \wedge \neg J(y)))$

$= \forall x \, \exists y(\neg Jx \vee ((x \sim y) \wedge \neg J(y)))$

2. $\neg \exists x(Px \to \forall y(Py \vee (\exists z Qz)))$
$= \neg \exists x(\neg Px \vee (\forall y(Py \vee (\exists z Qz))))$

$= \neg \exists x(\neg Px \vee (\forall y \exists z(Py \vee Qz)))$
$= \neg \exists x \forall y \exists z(\neg Px \vee Py \vee Qz))$

$= \forall x \forall y \exists z \neg(\neg Px \vee Py \vee Qz))$
$= \forall x \forall y \exists z. \, Px \wedge \neg Py \wedge \neg Qz$

Proof by Induction

Example $\quad \forall n \in N \quad \text{even}(n) \vee \text{odd}(n)$

Base Case. $\quad 0$ is even $\quad P(0)$ holds.
Hypo : $\quad k$ is either even or odd.
Step: $\quad P(s(k))$ if k is even $s(k)$ is odd.
$\qquad k$ odd -- even
in both cases $s(k)$ is even or odd
$P(s(k))$ holds.

Example. $\quad \forall y \in N \quad y + 0 = y$

Def for +.
of $y = y$
$s(x) + y = s(x+y)$

Base case $\quad 0 + 0 = 0$
Induction $\quad k + 0 = k.$
$s(k) + 0 = s(k+0) = s(k).$

$\forall y, k \in N, \quad s(y) + k = y + s(k)$

$s(0) + k = s(0+k) = s(k) + 0$
$\forall k \in N \quad s(z) + k = z + s(k)$

$s(s(z)) + k = s(s(z)+k) = s(z + s(k))$
$= s(z) + s(k).$

$\forall x, y \quad x + y = y + z.$

$0+y=y+0$

Assue $\forall y \in \mathbb{N}$ $\quad k+y=y+k$.

$$s(k)+y = s(k+y) = s(y+k) = s(y)+k = y+s(k).$$

Ordinary induction on $\mathbb{N}$.

$n ::= 0 \mid s(n)$.

$P(0)$ holds.

Assue $P(k)$ holds

show $P(s(k))$ holds

Inducive definition of lists

$l ::= [\,] \mid x :: l$

$[\,]$ empty list.

l is a list $\quad x :: l$ is a list.

$\text{len}\,[\,] = 0$

$\text{len}\; x :: xs = 1 + \text{len}(xs)$

$[\,] @ ys = ys$.

$(x :: xs) @ ys = x :: (xs @ ys)$

$\text{rev}\,[\,] = [\,]$

$\text{rev}(x :: xs) = (\text{rev}(xs) @ (x :: [\,]))$

$\forall xs, ys \in \text{list}.\ \text{len}(xs @ ys) = \text{len}(xs) + \text{len}(ys)$.

$\text{len}(xs @ [\,]) = \text{len}(xs) = \text{len}([\,]) + \text{len}(xs)$

Assue $\text{len}(zs @ ys) = \text{len}(zs) + \text{len}(ys)$

WTS $\quad \text{len}((z :: zs) @ ys) = \text{len}(z :: zs) + \text{len}(ys)$

$$\begin{aligned} &\text{len}(z :: (zs @ ys)) \\ &= 1 + \text{len}(zs @ ys) \\ &= 1 + \text{len}(zs) + \text{len}(ys) \\ &= \text{len}(z :: zs) + \text{len}(ys) \end{aligned}$$

$\forall xs \in \text{List}.\ \text{len}(\text{rev}(xs)) = \text{len}(xs)$

$\text{len}(\text{rev}([\,])) = \text{len}([\,]) = 0$

Assue $\text{len}(\text{rev}(ys)) = \text{len}(ys)$

WTS $\quad \text{len}(\text{rev}(y :: ys)) = \text{len}(y :: ys)$

$$len(rev(ys) @ (y::[])) =$$
$$\overset{\shortparallel}{len(rev(ys))} + len(y::[])$$
$$\overset{\shortparallel}{len(ys)} + 1 = len(y::ys)$$

Inductive. Definition of trees.

$$t ::= Lf \mid Br(\underset{\text{root}}{x}, \underset{\text{tree}}{t}, \underset{\text{tree.}}{t})$$

1. $P(Lf)$ holds.
2. Assume $P(t_1)$, $P(t_2)$
3. Prove $P(x, t_1, t_2)$ holds

$size(Lf) = 0$
$size(Br(x, t_1, t_2)) = 1 + size(t_1) + size(t_2)$

$refl(Lf) = Lf$
$refl(Br(x, t_1, t_2)) = Br(x, refl(t_2), refl(t_1))$

Example.
$\forall t \in Tree \quad refl(refl(t)) = t$
$refl(refl(Lf)) = Lf.$

Assume. $refl(refl(t_1)) = t_1$ and $refl(refl(t_2)) = t_2.$

RTP: $refl(refl(Br(x, t_1, t_2))) = Br(x, t_1, t_2)$
$$\overset{\shortparallel}{refl(Br(x, refl(t_2), refl(t_1)))}$$
$$\overset{\shortparallel}{Br(x, refl(refl(t_1)), refl(refl(t_2)))} = Br(x, t_1, t_2)$$

$\forall t \in Tree.\ size(refl(t)) = size(t)$
RTP: $size(refl(Lf)) = size(Lf) = 0$

Assume. $size(refl(t_1)) = size(t_1)$
$size(refl(t_2)) = size(t_2)$

$size(refl(Br(x, t_1, t_2)))$
$= size(Br(x, refl(t_2), refl(t_1)))$
$= 1 + size(refl(t_2)) + size(refl(t_1))$
$= 1 + size(t_2) + size(t_1)$
$= size(Br(x, t_1, t_2))$

Induction for formulas

$(\vee\ \neg)$ formulas $F = Q \mid \neg F \mid F \vee F$

P(Q) holds for Q.

P(A) (IH) → show P(¬A)

P(A), P(B) (IH) → show P(A∨B)

Every (∨, ¬) formula A has one more occurence of proposition letters than of ∨.

¬A has as many prop. as ∨.

∨. A, B

m $m-1$ n $n-1$

$A \vee B \Rightarrow \underline{m+n}$ prop

$m-1+n-1+1 = m+n-1$ ∨.

Two step induction.

show P(0) ∨.

P(S(0)) ∨. (base)

Assume P(k) and P(S(k)) ∨.

show P(S(S(k)))

Infinite descent.

$n_0 > n_1 \cdots$

$$\frac{\neg P(k) \rightarrow (\exists k' < k \in \mathbb{N}.\ \neg P(k'))}{\forall n \in \mathbb{N}.\ P(n)}$$

Infinit descent.

$\sqrt{2}$ not rational. $\neg \exists x, y \in \mathbb{N},\ \sqrt{2} = x/y$

$\sqrt{2} = x/y$ for $x, y \in \mathbb{N}$

$x^2 = 2y^2$. $x(x-y) = y(2y-x)$

$\frac{2y-x}{x-y} = \frac{x}{y} = \sqrt{2}$.

$x' = 2y - x$.

$y' = x - y$.

Also $2y > x > y$

$y' = x - y$

$y > y' > 0$

infinite descent on y

Re-write logic premise.

$\neg P(k) \rightarrow (\exists k' < k \in \mathbb{N}.\ \neg P(k'))$

$\Leftrightarrow \neg(\exists k' < k \in \mathbb{N}. \neg P(k') \to \neg\neg P(k)$

$\Leftrightarrow (\forall k' < k \in \mathbb{N}.\ P(k')) \to P(k)$

if $P(k)$ for all $k' < k$.

$\underline{P(k) \text{ holds}}$

For every N. $n \geq 2$. can be written as a produce of prime numbers

Hyp. every $n' < n$ $(n \geq 2)$ can be - - .

n is a prime ✓.

$n = \underline{m} \times \underline{k}.$ $(m, k < n)$

can be written as — —.

s.t. n — —.

Ackermann function.

$A(0, y) := y + 1.$

$A(s(x), 0) = A(x, 1)$

$A(s(x), s(y)) = A(x, A(s(x), y)).$

Assume $A(x', y')$ is defined for all $(x', y') < (x, y)$

$A(x, 0) = A(s(x)$

$A(s(x), s(y)) \to$ depend on $A(x, A(s(x), y)$

$A(s(x), 0)$

$A(x, \underbrace{1)}$ defined

$A(\underbrace{0}, x)$

define.

Induction on data structure

Infinite descend — depend on well-founded order of N.

Class modal logic

extend propositional logic.

with modalities

possibility

necessity

Assume infinite set $\mathcal{V}$ of propositional letters

$A ::= P$

$\mid \neg A$

$\vdots$

$\mid \Diamond A$ possibly

$\mid \Box A$ necessarily

A frame (Kripke model) is a pair

$M = (W, R)$

worlds

binary relation

$R \subseteq W \times W$

Valuation $\rho : \mathcal{V} \to Pow(W)$

$\rho(P)$ is the set of worlds P true

$M, w \vDash_\rho A$ $\quad w \in \rho(A)$

$M, w \vDash_\rho \Diamond A$ $\quad \exists w' \in W$ $\underline{Rww'}$ and $M, w' \vDash_\rho A$

$w \Leftrightarrow w'$

$\rho \Box A$ $\quad \forall w' \in W$ if Rww' then $M, w' \vDash_\rho A$

A is valid in M. $\quad M, w \vDash_\rho A$ for all $w \in W$ and ρ for M.

A is valid. $\quad \vDash A$

A is satisfiable in M. $\underline{M, w \vDash_\rho A}$

A is satisfiable if satisfiable in some frame.

$(\Box A \wedge \Box(A \to B)) \to \Box B$

$M = (W, R)$ $\quad \rho$ a evaluation

$M, w \vDash_\rho (\Box A \wedge \Box(A \to B)) \to \Box B$

$M, w \vDash_\rho \Box A$

$M, w \vDash_\rho \Box(A \to B)$

Rww'

$M, w' \models_\rho A.$

$M, w' \models_\rho A \to B.$

$M, w' \models_\rho B$

$\therefore M, w \models_\rho \Box B$

$\neg\Box A \leftrightarrow \Diamond\neg A$. valid.

$M, w \models_\rho \neg\Box A.$

$\exists w' \in W,\ Rww'$ and $M, w' \models_\rho \neg A.$

$\therefore M, w \models_\rho \Diamond\neg A.$

$\Box A \to \Diamond A$ invalid.

Take $R = \phi$.

$M, w \models_\rho \Box A$ always true (no w')

$\Diamond A$ always false.

$\Box A \to A$ is not valid

$M.$ $M = (W, \phi)$ $\rho(P) = \phi.$

$M, w \models_\rho \Box P$ always true.

$M, w \not\models_\rho P$ $w \notin \phi. = \rho(P)$

M reflexive $Rww.$

$\Box A \to A.$

$R \subseteq W \times W$ is transitive $Rxy.\ Ryz \to Rxz$

$M, w \models_\rho \Box A \to \Box\Box A$

Rww' $M, w' \models_\rho A.$ $Rw'w'' \to Rww''$

To prove $M, w \models_\rho \to \Box\Box A$

$M, w'' \models_\rho \to A$

$\Downarrow$

$M, w' \models_\rho \to \Box A$

$M \models \Box A \to \Box\Box A$ M is transitive

$\Diamond R \to \Box U$

$\Box F \to \Diamond L.$

$\Diamond\phi \leftrightarrow \neg\Box\neg\phi.$

$M, w \models_p \Diamond\phi \leftrightarrow \neg\Box\neg\phi.$

Rww' $\quad M, w' \models_p \phi$

$M, w' \not\models_p \neg\phi$

$\therefore M, w \not\models_p \Box\neg\phi$

$M, w \models_p \neg\Box\neg\phi$

$M, w \quad \Box(\phi \wedge \psi) \leftrightarrow \Box\phi \wedge \Box\psi.$

$M, w \models_p \Box(\phi \wedge \psi).$

$\forall w' \; Rww', \quad M, w \models_p \phi \wedge \psi$

$M, w \models_p \phi$ and $M, w \models_p \psi.$

$W = \{w_0, w_1, w_2, w_3\}$

$Pow(W) = \quad \emptyset, \{w_1\}, \{w_2\} \{w_3\} \{w_1, w_2\} \ldots$