binary expressions:

theorems:

antitheorems:

binary expressions: represent anything that comes in two kinds

theorems: represent one kind

antitheorems: represent the other kind

binary expressions: represent anything that comes in two kinds

represent statements about the world (natural or constructed, real or imaginary)

theorems: represent one kind

represent true statements

antitheorems: represent the other kind

represent false statements

binary expressions: represent anything that comes in two kinds

represent statements about the world (natural or constructed, real or imaginary)

represent digital circuits

theorems: represent one kind

represent true statements

represent circuits with high voltage output

antitheorems: represent the other kind

represent false statements

represent circuits with low voltage output

binary expressions: represent anything that comes in two kinds

- represent statements about the world (natural or constructed, real or imaginary)
- represent digital circuits
- represent human behavior

theorems: represent one kind

- represent true statements
- represent circuits with high voltage output
- represent innocent behavior

antitheorems: represent the other kind

- represent false statements
- represent circuits with low voltage output
- represent guilty behavior

0 operands $\top$ $\bot$

0 operands	$\top$ $\bot$
1 operand	$\neg x$

0 operands	$\top \quad \bot$
1 operand	$\neg x$
2 operands	$x \land y \quad x \lor y \quad x \Rightarrow y \quad x \Leftarrow y \quad x = y \quad x \neq y$

0 operands	$\top$ $\bot$
1 operand	$\neg x$
2 operands	$x \wedge y$ $x \vee y$ $x \Rightarrow y$ $x \Leftarrow y$ $x = y$ $x \neq y$

0 operands	$\top$ $\bot$
1 operand	$\neg x$
2 operands	$x \land y$ $x \lor y$ $x \Rightarrow y$ $x \Leftarrow y$ $x = y$ $x \neq y$

0 operands $\top$ $\bot$

1 operand $\neg x$

2 operands $x \wedge y$ $x \vee y$ $x \Rightarrow y$ $x \Leftarrow y$ $x = y$ $x \neq y$

0 operands	$\top$ $\bot$
1 operand	$\neg x$
2 operands	$x \wedge y$ $x \vee y$ $x \Rightarrow y$ $x \Leftarrow y$ $x = y$ $x \neq y$

0 operands $\top$ $\bot$

1 operand $\neg x$

2 operands $x \wedge y$ $x \vee y$ $x \Rightarrow y$ $x \Leftarrow y$ $x = y$ $x \neq y$

0 operands	$\top$ $\bot$
1 operand	$\neg x$
2 operands	$x \wedge y$ $x \vee y$ $x \Rightarrow y$ $x \Leftarrow y$ $x = y$ $x \neq y$

0 operands	$\top \quad \bot$
1 operand	$\neg x$
2 operands	$x \wedge y \quad x \vee y \quad x \Rightarrow y \quad x \Leftarrow y \quad x = y \quad x \neq y$
3 operands	**if** *x* **then** *y* **else** *z* **fi**

0 operands	$\top$ $\bot$
1 operand	$\neg x$
2 operands	$x \wedge y$ $x \vee y$ $x \Rightarrow y$ $x \Leftarrow y$ $x = y$ $x \neq y$
3 operands	**if** x **then** y **else** z **fi**

precedence and parentheses

0 operands	$\top \quad \bot$
1 operand	$\neg x$
2 operands	$x \wedge y \quad x \vee y \quad x \Rightarrow y \quad x \Leftarrow y \quad x = y \quad x \neq y$
3 operands	**if** x **then** y **else** z **fi**

precedence and parentheses

associative operators: $\wedge \quad \vee \quad = \quad \neq$

$x \wedge y \wedge z$ means either $(x \wedge y) \wedge z$ or $x \wedge (y \wedge z)$

$x \vee y \vee z$ means either $(x \vee y) \vee z$ or $x \vee (y \vee z)$

0 operands	$\top \quad \bot$
1 operand	$\neg x$
2 operands	$x \wedge y \quad x \vee y \quad x \Rightarrow y \quad x \Leftarrow y \quad x = y \quad x \neq y$
3 operands	**if** x **then** y **else** z **fi**

precedence and parentheses

associative operators: $\wedge \quad \vee \quad = \quad \neq$

$x \wedge y \wedge z$ means either $(x \wedge y) \wedge z$ or $x \wedge (y \wedge z)$

$x \vee y \vee z$ means either $(x \vee y) \vee z$ or $x \vee (y \vee z)$

continuing operators: $\Rightarrow \quad \Leftarrow \quad = \quad \neq$

$x = y = z$ means $x = y \;\wedge\; y = z$

$x \Rightarrow y \Rightarrow z$ means $(x \Rightarrow y) \;\wedge\; (y \Rightarrow z)$

0 operands $\top \quad \bot$

1 operand $\neg x$

2 operands $x \wedge y \quad x \vee y \quad x \Rightarrow y \quad x \Leftarrow y \quad x = y \quad x \neq y$

3 operands **if** x **then** y **else** z **fi**

precedence and parentheses

associative operators: $\wedge \quad \vee \quad = \quad \neq$

$x \wedge y \wedge z$ means either $(x \wedge y) \wedge z$ or $x \wedge (y \wedge z)$

$x \vee y \vee z$ means either $(x \vee y) \vee z$ or $x \vee (y \vee z)$

continuing operators: $\Rightarrow \quad \Leftarrow \quad = \quad \neq$

$x = y = z$ means $x = y \;\wedge\; y = z$

$x \Rightarrow y \Rightarrow z$ means $(x \Rightarrow y) \;\wedge\; (y \Rightarrow z)$

big operators: $\Large = \quad \Large \Rightarrow \quad \Large \Leftarrow$

same as $= \quad \Rightarrow \quad \Leftarrow$ but later precedence

$x \Large = \normalsize y \Large \Rightarrow \normalsize z$ means $(x \Large = \normalsize y) \;\wedge\; (y \Large \Rightarrow \normalsize z)$

value tables (truth tables)

	$\top$	$\bot$
$\neg$	$\bot$	$\top$

	$\top\top$	$\top\bot$	$\bot\top$	$\bot\bot$
$\wedge$	$\top$	$\bot$	$\bot$	$\bot$
$\vee$	$\top$	$\top$	$\top$	$\bot$
$\Rightarrow$	$\top$	$\bot$	$\top$	$\top$
$\Leftarrow$	$\top$	$\top$	$\bot$	$\top$
$=$	$\top$	$\bot$	$\bot$	$\top$
$\ne$	$\bot$	$\top$	$\top$	$\bot$

	$\top\top\top$	$\top\top\bot$	$\top\bot\top$	$\top\bot\bot$	$\bot\top\top$	$\bot\top\bot$	$\bot\bot\top$	$\bot\bot\bot$
if then else fi	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$

value tables (truth tables)

	$\top$	$\bot$
$\neg$	$\bot$	$\top$

	$\top\top$	$\top\bot$	$\bot\top$	$\bot\bot$
$\wedge$	$\top$	$\bot$	$\bot$	$\bot$
$\vee$	$\top$	$\top$	$\top$	$\bot$
$\Rightarrow$	$\top$	$\bot$	$\top$	$\top$
$\Leftarrow$	$\top$	$\top$	$\bot$	$\top$
$=$	$\top$	$\bot$	$\bot$	$\top$
$\neq$	$\bot$	$\top$	$\top$	$\bot$

	$\top\top\top$	$\top\top\bot$	$\top\bot\top$	$\top\bot\bot$	$\bot\top\top$	$\bot\top\bot$	$\bot\bot\top$	$\bot\bot\bot$
if then else fi	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$

value tables (truth tables)

	⊤	⊥
¬	⊥	⊤

	⊤⊤	⊤⊥	⊥⊤	⊥⊥
∧	⊤	⊥	⊥	⊥
∨	⊤	⊤	⊤	⊥
⇒	⊤	⊥	⊤	⊤
⇐	⊤	⊤	⊥	⊤
=	⊤	⊥	⊥	⊤
⧧	⊥	⊤	⊤	⊥

	⊤⊤⊤	⊤⊤⊥	⊤⊥⊤	⊤⊥⊥	⊥⊤⊤	⊥⊤⊥	⊥⊥⊤	⊥⊥⊥
if then else fi	⊤	⊤	⊥	⊥	⊤	⊥	⊤	⊥

value tables (truth tables)

	⊤	⊥
¬	⊥	⊤

	⊤⊤	⊤⊥	⊥⊤	⊥⊥
∧	⊤	⊥	⊥	⊥
∨	⊤	⊤	⊤	⊥
⇒	⊤	⊥	⊤	⊤
⇐	⊤	⊤	⊥	⊤
=	⊤	⊥	⊥	⊤
⧧	⊥	⊤	⊤	⊥

	⊤⊤⊤	⊤⊤⊥	⊤⊥⊤	⊤⊥⊥	⊥⊤⊤	⊥⊤⊥	⊥⊥⊤	⊥⊥⊥
if then else fi	⊤	⊤	⊥	⊥	⊤	⊥	⊤	⊥

value tables (truth tables)

	$\top$	$\bot$
$\neg$	$\bot$	$\top$

	$\top\top$	$\top\bot$	$\bot\top$	$\bot\bot$
$\wedge$	$\top$	$\bot$	$\bot$	$\bot$
$\vee$	$\top$	$\top$	$\top$	$\bot$
$\Rightarrow$	$\top$	$\bot$	$\top$	$\top$
$\Leftarrow$	$\top$	$\top$	$\bot$	$\top$
$=$	$\top$	$\bot$	$\bot$	$\top$
$\neq$	$\bot$	$\top$	$\top$	$\bot$

	$\top\top\top$	$\top\top\bot$	$\top\bot\top$	$\top\bot\bot$	$\bot\top\top$	$\bot\top\bot$	$\bot\bot\top$	$\bot\bot\bot$
if then else fi	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$

value tables (truth tables)

	$\top$	$\bot$
$\neg$	$\bot$	$\top$

	$\top\top$	$\top\bot$	$\bot\top$	$\bot\bot$
$\wedge$	$\top$	$\bot$	$\bot$	$\bot$
$\vee$	$\top$	$\top$	$\top$	$\bot$
$\Rightarrow$	$\top$	$\bot$	$\top$	$\top$
$\Leftarrow$	$\top$	$\top$	$\bot$	$\top$
$=$	$\top$	$\bot$	$\bot$	$\top$
$\neq$	$\bot$	$\top$	$\top$	$\bot$

	$\top\top\top$	$\top\top\bot$	$\top\bot\top$	$\top\bot\bot$	$\bot\top\top$	$\bot\top\bot$	$\bot\bot\top$	$\bot\bot\bot$
if then else fi	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$

value tables (truth tables)

	$\top$	$\bot$
$\neg$	$\bot$	$\top$

	$\top\top$	$\top\bot$	$\bot\top$	$\bot\bot$
$\wedge$	$\top$	$\bot$	$\bot$	$\bot$
$\vee$	$\top$	$\top$	$\top$	$\bot$
$\Rightarrow$	$\top$	$\bot$	$\top$	$\top$
$\Leftarrow$	$\top$	$\top$	$\bot$	$\top$
$=$	$\top$	$\bot$	$\bot$	$\top$
⧧	$\bot$	$\top$	$\top$	$\bot$

	$\top\top\top$	$\top\top\bot$	$\top\bot\top$	$\top\bot\bot$	$\bot\top\top$	$\bot\top\bot$	$\bot\bot\top$	$\bot\bot\bot$
if then else fi	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$

value tables (truth tables)

	$\top$	$\bot$
$\neg$	$\bot$	$\top$

	$\top\top$	$\top\bot$	$\bot\top$	$\bot\bot$
$\wedge$	$\top$	$\bot$	$\bot$	$\bot$
$\vee$	$\top$	$\top$	$\top$	$\bot$
$\Rightarrow$	$\top$	$\bot$	$\top$	$\top$
$\Leftarrow$	$\top$	$\top$	$\bot$	$\top$
$=$	$\top$	$\bot$	$\bot$	$\top$
$\neq$	$\bot$	$\top$	$\top$	$\bot$

⇍

	$\top\top\top$	$\top\top\bot$	$\top\bot\top$	$\top\bot\bot$	$\bot\top\top$	$\bot\top\bot$	$\bot\bot\top$	$\bot\bot\bot$
if then else fi	$\top$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$

value tables (truth tables)

	⊤	⊥
$\neg$	⊥	⊤

	⊤⊤	⊤⊥	⊥⊤	⊥⊥
$\wedge$	⊤	⊥	⊥	⊥
$\vee$	⊤	⊤	⊤	⊥
$\Rightarrow$	⊤	⊥	⊤	⊤
$\Leftarrow$	⊤	⊤	⊥	⊤
$=$	⊤	⊥	⊥	⊤
$\neq$	⊥	⊤	⊤	⊥

	⊤⊤⊤	⊤⊤⊥	⊤⊥⊤	⊤⊥⊥	⊥⊤⊤	⊥⊤⊥	⊥⊥⊤	⊥⊥⊥
if then else fi	⊤	⊤	⊥	⊥	⊤	⊥	⊤	⊥

←

variables are for substitution (instantiation)

variables are for substitution (instantiation)

- add parentheses to maintain precedence

  in $x \land y$ replace x by $\bot$ and y by $\bot \lor \top$ result: $\bot \land (\bot \lor \top)$

variables are for substitution (instantiation)

- add parentheses to maintain precedence

  in $x \wedge y$ replace x by $\bot$ and y by $\bot \vee \top$ result: $\bot \wedge (\bot \vee \top)$

- every occurrence of a variable must be replaced by the same expression

  in $x \wedge x$ replace x by $\bot$ result: $\bot \wedge \bot$

variables are for substitution (instantiation)

- add parentheses to maintain precedence

  in $x \wedge y$ replace x by $\bot$ and y by $\bot \vee \top$ result: $\bot \wedge (\bot \vee \top)$

- every occurrence of a variable must be replaced by the same expression

  in $x \wedge x$ replace x by $\bot$ result: $\bot \wedge \bot$

- different variables can be replaced by the same expression or different expressions

  in $x \wedge y$ replace x by $\bot$ and y by $\bot$ result: $\bot \wedge \bot$

variables are for substitution (instantiation)

- add parentheses to maintain precedence

  in $x \land y$ replace x by $\bot$ and y by $\bot \lor \top$ result: $\bot \land (\bot \lor \top)$

- every occurrence of a variable must be replaced by the same expression

  in $x \land x$ replace x by $\bot$ result: $\bot \land \bot$

- different variables can be replaced by the same expression or different expressions

  in $x \land y$ replace x by $\bot$ and y by $\bot$ result: $\bot \land \bot$

  in $x \land y$ replace x by $\top$ and y by $\bot$ result: $\top \land \bot$

new binary expressions

(the grass is green)

(the sky is green)

(there is life elsewhere in the universe)

(intelligent messages are coming from space)

new binary expressions

(the grass is green)

(the sky is green)

(there is life elsewhere in the universe)

(intelligent messages are coming from space)

$1 + 1 = 2$

$0 / 0 = 5$

new binary expressions

(the grass is green)

(the sky is green)

(there is life elsewhere in the universe)

(intelligent messages are coming from space)

1 + 1 = 2

0 / 0 = 5

---

consistent: no binary expression is both a theorem and an antitheorem

(no overclassified expressions)

new binary expressions

(the grass is green)

(the sky is green)

(there is life elsewhere in the universe)

(intelligent messages are coming from space)

$1 + 1 = 2$

$0 / 0 = 5$

---

consistent: no binary expression is both a theorem and an antitheorem

(no overclassified expressions)

complete: every fully instantiated binary expression is either a theorem or an antitheorem

(no unclassified expressions)

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

$x+y = y+x$ is a mathematical expression

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

$x+y = y+x$ is a mathematical expression

represents a truth in an application such that

when you put quantities together, the total quantity does not depend

on the order in which you put them together

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

$x+y = y+x$ is a mathematical expression

represents a truth in an application such that

when you put quantities together, the total quantity does not depend

on the order in which you put them together

is an axiom

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

$x+y = y+x$ is a mathematical expression

represents a truth in an application such that

when you put quantities together, the total quantity does not depend

on the order in which you put them together

is an axiom

is a theorem

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

$x+y = y+x$ is a mathematical expression

represents a truth in an application such that

when you put quantities together, the total quantity does not depend

on the order in which you put them together

is an axiom

is a theorem

is equivalent to $\top$

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

$x+y = y+x$ is a mathematical expression

represents a truth in an application such that

when you put quantities together, the total quantity does not depend

on the order in which you put them together

is an axiom

is a theorem

is equivalent to $\top$

$x+y = y+x$ is true (not really)

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

axiom: $\top$

antiaxiom: $\bot$

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

axiom: $\top$

antiaxiom: $\bot$

axiom: (the grass is green)

antiaxiom: (the sky is green)

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

axiom: $\top$

antiaxiom: $\bot$

axiom: (the grass is green)

antiaxiom: (the sky is green)

axiom: (intelligent messages are coming from space)

$\Rightarrow$ (there is life elsewhere in the universe)

Proof Rules

Axiom Rule If a binary expression is an axiom, then it is a theorem.

If a binary expression is an antiaxiom, then it is an antitheorem.

axiom: $\top$

antiaxiom: $\bot$

axiom: (the grass is green)

antiaxiom: (the sky is green)

axiom: (intelligent messages are coming from space)

$\Rightarrow$ (there is life elsewhere in the universe)

Evaluation Rule If all the binary subexpressions of a binary expression are classified,

then it is classified according to the value tables.

Proof Rules

Completion Rule If a binary expression contains unclassified binary subexpressions, and all ways of classifying them place it in the same class, then it is in that class.

Proof Rules

Completion Rule If a binary expression contains unclassified binary subexpressions, and all ways of classifying them place it in the same class, then it is in that class.

theorem: (there is life elsewhere in the universe) $\lor$ $\top$

Proof Rules

Completion Rule If a binary expression contains unclassified binary subexpressions, and all ways of classifying them place it in the same class, then it is in that class.

theorem: (there is life elsewhere in the universe) $\lor$ $\top$

theorem: (there is life elsewhere in the universe)
$\lor$ $\neg$(there is life elsewhere in the universe)

Proof Rules

Completion Rule If a binary expression contains unclassified binary subexpressions, and all ways of classifying them place it in the same class, then it is in that class.

theorem: (there is life elsewhere in the universe) $\lor$ $\top$

theorem: (there is life elsewhere in the universe)
$\lor$ $\lnot$(there is life elsewhere in the universe)

antitheorem: (there is life elsewhere in the universe)
$\land$ $\lnot$(there is life elsewhere in the universe)

Proof Rules

Consistency Rule If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

Proof Rules

Consistency Rule If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

We are given that x and $x \Rightarrow y$ are theorems. What is y ?

Proof Rules

Consistency Rule If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

We are given that x and $x \Rightarrow y$ are theorems. What is y ?

If y were an antitheorem, then by the Evaluation Rule, $x \Rightarrow y$ would be an antitheorem.

That would be inconsistent. So y is a theorem.

Proof Rules

Consistency Rule If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

We are given that x and $x \Rightarrow y$ are theorems. What is y ?

If y were an antitheorem, then by the Evaluation Rule, $x \Rightarrow y$ would be an antitheorem.

That would be inconsistent. So y is a theorem.

We are given that $\neg x$ is a theorem. What is x ?

Proof Rules

Consistency Rule If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

We are given that x and $x \Rightarrow y$ are theorems. What is y ?

If y were an antitheorem, then by the Evaluation Rule, $x \Rightarrow y$ would be an antitheorem.

That would be inconsistent. So y is a theorem.

We are given that $\neg x$ is a theorem. What is x ?

If x were a theorem, then by the Evaluation Rule, $\neg x$ would be an antitheorem.

That would be inconsistent. So x is an antitheorem.

Proof Rules

Consistency Rule If a classified binary expression contains binary subexpressions, and only one way of classifying them is consistent, then they are classified that way.

We are given that x and $x \Rightarrow y$ are theorems. What is y ?

If y were an antitheorem, then by the Evaluation Rule, $x \Rightarrow y$ would be an antitheorem.

That would be inconsistent. So y is a theorem.

We are given that $\neg x$ is a theorem. What is x ?

If x were a theorem, then by the Evaluation Rule, $\neg x$ would be an antitheorem.

That would be inconsistent. So x is an antitheorem.

No need to talk about antiaxioms and antitheorems.

Proof Rules

Transparency Rule A binary expression does not gain, lose, or change classification when a classified subexpression is replaced by another expression in the same class.

Proof Rules

Transparency Rule A binary expression does not gain, lose, or change classification when a classified subexpression is replaced by another expression in the same class.

$\neg x \Rightarrow (x \wedge x) \vee y = \quad \top$

Proof Rules

Transparency Rule A binary expression does not gain, lose, or change classification when a classified subexpression is replaced by another expression in the same class.

$$\neg x \Rightarrow (x \wedge x) \vee y = \top$$

$$\neg x \Rightarrow (x \wedge x) \vee y = (y \vee \top)$$

Proof Rules

Transparency Rule A binary expression does not gain, lose, or change classification when a classified subexpression is replaced by another expression in the same class.

$$\neg x \Rightarrow (x \wedge x) \vee y = \top$$

$$\neg x \Rightarrow (x \wedge x) \vee y = (y \vee \top)$$

$$\neg x \Rightarrow x \vee y = (y \vee \top)$$

Proof Rules

Instance Rule If a binary expression is classified,

then all its instances have that same classification.

Proof Rules

Instance Rule If a binary expression is classified,

then all its instances have that same classification.

axiom: $x = x$

Proof Rules

Instance Rule If a binary expression is classified,

then all its instances have that same classification.

axiom: $x = x$

theorem: $x = x$

Proof Rules

Instance Rule If a binary expression is classified,

then all its instances have that same classification.

axiom: $x = x$

theorem: $x = x$

theorem: $(\top{=}\bot \lor \bot) = (\top{=}\bot \lor \bot)$

Proof Rules

Instance Rule If a binary expression is classified,

then all its instances have that same classification.

axiom: $x = x$

theorem: $x = x$

theorem: $(\top=\bot \lor \bot) = (\top=\bot \lor \bot)$

theorem: (intelligent messages are coming from space)
= (intelligent messages are coming from space)

Proof Rules

Instance Rule If a binary expression is classified,

then all its instances have that same classification.

axiom: $x = x$

theorem: $x = x$

theorem: $(\top = \bot \lor \bot) = (\top = \bot \lor \bot)$

theorem: (intelligent messages are coming from space)
= (intelligent messages are coming from space)

Classical Logic: all six rules

Constructive Logic: not Completion Rule

Evaluation Logic: neither Consistency Rule nor Completion Rule

Expression and Proof Format

$a \wedge b \;\vee\; c$

Expression and Proof Format

$a{\wedge}b \;\vee\; c$ **NOT** $a \;\wedge\; b{\vee}c$

Expression and Proof Format

$a{\wedge}b \ \vee \ c$ **NOT** $a \ \wedge \ b{\vee}c$

$($ *first part*

$\wedge$ *second part* $)$

Expression and Proof Format

$a{\wedge}b \;\vee\; c$ **NOT** $a \;\wedge\; b{\vee}c$

(*first part*

∧ *second part*)

C and Java convention

```
while (something) {
        various lines
        in the body
        of the loop
}
```

Expression and Proof Format

$a{\wedge}b \;\vee\; c$ **NOT** $a \;\wedge\; b{\vee}c$

$($ $\quad$ *first part*

$\wedge$ $\quad$ *second part* $\quad)$

Expression and Proof Format

$a{\wedge}b \;\vee\; c$ **NOT** $a \;\wedge\; b{\vee}c$

$($ *first part*

$\wedge$ *second part* $)$

first part

$=$ *second part*

Expression and Proof Format

$a \wedge b \;\; \vee \;\; c$ **NOT** $a \;\; \wedge \;\; b \vee c$

$($ *first part*

$\wedge$ *second part* $)$

first part

$=$ *second part*

expression0

$=$ *expression1*

$=$ *expression2*

$=$ *expression3*

Expression and Proof Format

$a \wedge b \ \vee \ c$ **NOT** $a \ \wedge \ b \vee c$

$$\begin{array}{ll} (& \textit{first part} \\ \wedge & \textit{second part} \quad) \end{array}$$

$$\begin{array}{ll} & \textit{first part} \\ = & \textit{second part} \end{array}$$

$$\begin{array}{ll} & \textit{expression0} \\ = & \textit{expression1} \\ = & \textit{expression2} \\ = & \textit{expression3} \end{array} \qquad \text{means} \qquad \begin{array}{ll} & \textit{expression0}=\textit{expression1} \\ \wedge & \textit{expression1}=\textit{expression2} \\ \wedge & \textit{expression2}=\textit{expression3} \end{array}$$

Expression and Proof Format

$a{\wedge}b \;\vee\; c$ **NOT** $a \;\wedge\; b{\vee}c$

$($ *first part*
$\wedge$ *second part* $)$

first part
$=$ *second part*

expression0
$=$ *expression1*
$=$ *expression2*
$=$ *expression3*

Expression and Proof Format

$a \wedge b \ \vee \ c$ **NOT** $a \ \wedge \ b \vee c$

$($ *first part*

$\wedge$ *second part* $)$

first part

$=$ *second part*

	expression0	hint0
=	*expression1*	hint1
=	*expression2*	hint2
=	*expression3*	

Expression and Proof Format

Prove $a \wedge b \Rightarrow c \;\; = \;\; a \Rightarrow (b \Rightarrow c)$

Expression and Proof Format

Prove $a \wedge b \Rightarrow c \;=\; a \Rightarrow (b \Rightarrow c)$

	$a \wedge b \Rightarrow c$	Material Implication
$=$	$\neg(a \wedge b) \vee c$	Duality
$=$	$\neg a \vee \neg b \vee c$	Material Implication
$=$	$a \Rightarrow \neg b \vee c$	Material Implication
$=$	$a \Rightarrow (b \Rightarrow c)$	

Expression and Proof Format

Prove $a \wedge b \Rightarrow c \;=\; a \Rightarrow (b \Rightarrow c)$

	$a \wedge b \Rightarrow c$	Material Implication
$=$	$\neg(a \wedge b) \vee c$	Duality
$=$	$\neg a \vee \neg b \vee c$	Material Implication
$=$	$a \Rightarrow \neg b \vee c$	Material Implication
$=$	$a \Rightarrow (b \Rightarrow c)$	

Material Implication: $a \Rightarrow b \;=\; \neg a \vee b$

Expression and Proof Format

Prove $a \wedge b \Rightarrow c \;=\; a \Rightarrow (b \Rightarrow c)$

	$a \wedge b \Rightarrow c$	Material Implication
$=$	$\neg(a \wedge b) \vee c$	Duality
$=$	$\neg a \vee \neg b \vee c$	Material Implication
$=$	$a \Rightarrow \neg b \vee c$	Material Implication
$=$	$a \Rightarrow (b \Rightarrow c)$	

Material Implication: $a \Rightarrow b \;=\; \neg a \vee b$

Instance of Material Implication: $a \wedge b \;\Rightarrow\; c \;=\; \neg(a \wedge b) \vee\; c$

Expression and Proof Format

Prove $a \wedge b \Rightarrow c \;=\; a \Rightarrow (b \Rightarrow c)$

	$a \wedge b \Rightarrow c$	Material Implication
$=$	$\neg(a \wedge b) \vee c$	Duality
$=$	$\neg a \vee \neg b \vee c$	Material Implication
$=$	$a \Rightarrow \neg b \vee c$	Material Implication
$=$	$a \Rightarrow (b \Rightarrow c)$	

Expression and Proof Format

Prove $a \wedge b \Rightarrow c \;=\; a \Rightarrow (b \Rightarrow c)$

	$a \wedge b \Rightarrow c$	Material Implication
$=$	$\neg(a \wedge b) \vee c$	Duality
$=$	$\neg a \vee \neg b \vee c$	Material Implication
$=$	$a \Rightarrow \neg b \vee c$	Material Implication
$=$	$a \Rightarrow (b \Rightarrow c)$	

	$(a \wedge b \Rightarrow c \;=\; a \Rightarrow (b \Rightarrow c))$	Material Implication 3 times
$=$	$(\neg(a \wedge b) \vee c \;=\; \neg a \vee (\neg b \vee c))$	Duality
$=$	$(\neg a \vee \neg b \vee c \;=\; \neg a \vee \neg b \vee c)$	Reflexivity of $=$
$=$	$\top$	