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 $\top \quad \bot$

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: $\Rightarrow \ \Leftarrow \ = \ \neq$

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

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

big operators: $\boldsymbol{=} \quad \boldsymbol{\Longrightarrow} \quad \boldsymbol{\Longleftarrow}$

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

$x \boldsymbol{=} y \boldsymbol{\Longrightarrow} z$ means $(x \boldsymbol{=} y) \ \wedge \ (y \boldsymbol{\Longrightarrow} z)$

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

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: $\bot \wedge \bot$

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

new binary expressions

(the grass is green)

(the sky is green)

(there is life elsewhere in the universe)

(intelligent messages are coming from space)

$1 + 1 = 2$

$0 / 0 = 5$

---

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.

$x+y = y+x$ 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

$x+y = y+x$ 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.

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 truth 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.

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)

antitheorem: (there is life elsewhere in the universe)

$\land$ $\neg$(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.

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

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 five rules

Constructive Logic: not Completion Rule

Evaluation Logic: neither Consistency Rule nor Completion Rule

Expression and Proof Format

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

(*first part*

$\wedge$ *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 $a \ \wedge \ b \vee c$

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

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

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

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

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

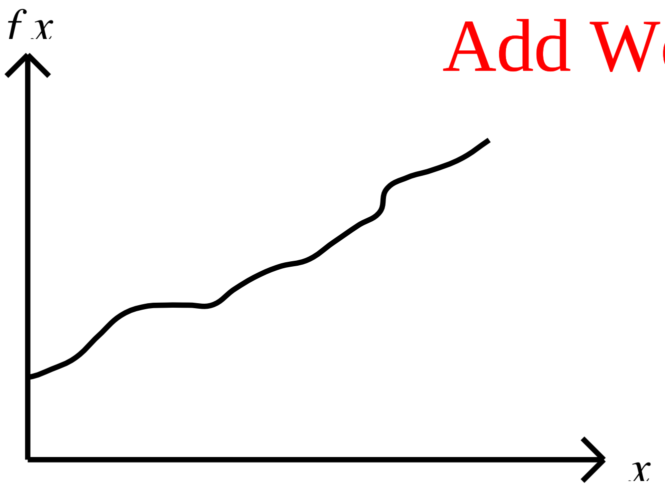
	$a \wedge b \Rightarrow c$	Material Implication
$=$	$\neg(a \wedge b) \vee c$	Duality
$=$	$\neg a \vee \neg b \vee c$	Material Implication
$=$	$a \Rightarrow \neg b \vee c$	Material Implication
$=$	$a \Rightarrow (b \Rightarrow c)$	

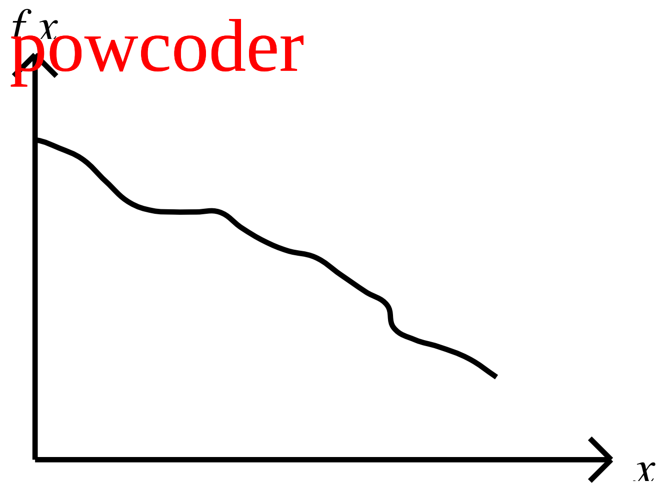
	$(a \wedge b \Rightarrow c \;=\; a \Rightarrow (b \Rightarrow c))$	Material Implication 3 times
$=$	$(\neg(a \wedge b) \vee c \;=\; \neg a \vee (\neg b \vee c))$	Duality
$=$	$(\neg a \vee \neg b \vee c \;=\; \neg a \vee \neg b \vee c)$	Reflexivity of $=$
$=$	$\top$	

Monotonicity and Antimonotonicity

covariance	**and**	**contravariance**
varies directly as	**and**	**varies inversely as**
nondecreasing	**and**	**nonincreasing**
sorted	**and**	**sorted backwards**

$x \le y \;\Rightarrow\; f\,x \le f\,y$ $\qquad\qquad$ $x \le y \;\Rightarrow\; f\,x \ge f\,y$





Monotonicity and Antimonotonicity

number:	$x \le y$	x is less than or equal to y
	$-\infty \le +\infty \quad 0 \le 1$	smaller $\le$ larger
	$x \le y \Rightarrow f\,x \le f\,y$	f is monotonic
		as x gets larger, $f\,x$ gets larger (or equal)
	$x \le y \Rightarrow f\,x \ge f\,y$	f is antimonotonic
		as x gets larger, $f\,x$ gets smaller (or equal)
binary:	$x \Rightarrow y$	x implies y $\quad$ x is stronger than or equal to y
	$\bot \Rightarrow \top$	stronger $\Rightarrow$ weaker
	$x \Rightarrow y \implies f\,x \Rightarrow f\,y$	f is monotonic
		as x gets weaker, $f\,x$ gets weaker (or equal)
	$x \Rightarrow y \implies f\,x \Leftarrow f\,y$	f is antimonotonic
		as x gets weaker, $f\,x$ gets stronger (or equal)

Monotonicity and Antimonotonicity

$\neg a$	antimonotonic in a	
$a \wedge b$	monotonic in a	monotonic in b
$a \vee b$	monotonic in a	monotonic in b
$a \Rightarrow b$	antimonotonic in a	monotonic in b
$a \Leftarrow b$	monotonic in a	antimonotonic in b
if a **then** b **else** c **fi**	monotonic in b	monotonic in c

	$\neg(a \wedge \neg(a \vee b))$	use the Law of Generalization $a \Rightarrow a \vee b$
$\Leftarrow$	$\neg(a \wedge \neg a)$	now use the Law of Noncontradiction
$=$	$\top$	

Context

In $a \wedge b$, when changing a, we can assume b.

$$a \wedge b$$

$$= \quad \underset{\downarrow}{} c \wedge b$$

If b is $\top$, we have assumed correctly.

If b is $\bot$, then $a \wedge b$ and $c \wedge b$ are both $\bot$, so the equation is $\top$ anyway.

Context

In $a \wedge b$, when changing a, we can assume b.

In $a \wedge b$, when changing b, we can assume a.

	$\neg(a \wedge \neg(a \vee b))$	assume a to simplify $\neg(a \vee b)$
$=$	$\neg(a \wedge \neg(\top \vee b))$	Symmetry Law and Base Law for $\vee$
$=$	$\neg(a \wedge \neg\top)$	Truth Table for $\neg$
$=$	$\neg(a \wedge \bot)$	Base Law for $\wedge$
$=$	$\top$	

Context

In $a \wedge b$, when changing a , we can assume b .

In $a \wedge b$, when changing b , we can assume a .

In $a \vee b$, when changing a , we can assume $\neg b$.

In $a \vee b$, when changing b , we can assume $\neg a$.

In $a \Rightarrow b$, when changing a , we can assume $\neg b$.

In $a \Rightarrow b$, when changing b , we can assume a .

In $a \Leftarrow b$, when changing a , we can assume b .

In $a \Leftarrow b$, when changing b , we can assume $\neg a$.

In **if** a **then** b **else** c **fi** , when changing a , we can assume $b \neq c$.

In **if** a **then** b **else** c **fi** , when changing b , we can assume a .

In **if** a **then** b **else** c **fi** , when changing c , we can assume $\neg a$.

Number Theory

number expressions represent quantity

number expressions

$0 \quad 1 \quad 2 \quad 597 \quad 1.2 \quad 1\mathrm{e}10 \quad \infty$

$-x \quad x+y \quad x-y \quad x \times y \quad x/y \quad x^y$

if a **then** x **else** y **fi**

binary expressions

$x=y \quad x \neq y \quad x<y \quad x>y \quad x \leq y \quad x \geq y$

Character Theory

“A”　　“a”　　“ ”　　“““”　　“”””

succ　　*pred*　　**if then else fi**

$= \quad \neq \quad < \quad > \quad \leq \quad \geq$