Appendix

Locical Equivalences

Name	Rule
Identity Laws	$p \wedge T \equiv p$ $p \vee F \equiv p$
Domination Laws	$p \wedge F \equiv F$ $p \vee T \equiv T$
Idempotent Laws	$p \wedge p \equiv p$ $p \vee p \equiv p$
Commutative Laws	$p \wedge q \equiv q \wedge p$ $p \vee q \equiv q \vee p$
Associative Laws	$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$ $p \vee (q \vee r) \equiv (p \vee q) \vee r$
Distributive Laws	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
Absorption Laws	$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$
Negation Laws	$p \wedge \neg p \equiv F$ $p \vee \neg p \equiv T$
Double Negation Law	$\neg(\neg p) \equiv p$
De Morgan's Laws	$\neg(p \wedge q) \equiv (\neg p) \vee (\neg q)$ $\neg(p \vee q) \equiv (\neg p) \wedge (\neg q)$

Rules of Inference

Name	Rule
Specialization	$\begin{array}{c} p \wedge q \\ \hline p \end{array}$
Generalization	$\begin{array}{c} p \\ \hline p \vee q \end{array}$
Modus ponens	$\begin{array}{c} p \\ p \to q \\ \hline q \end{array}$
Modus tollens	$\begin{array}{c} \neg q \\ p \to q \\ \hline \neg p \end{array}$
Transitivity	$\begin{array}{c} p \to q \\ q \to r \\ \hline p \to r \end{array}$
Elimination	$\begin{array}{c} p \vee q \\ \neg p \\ \hline q \end{array}$
Proof by cases.	$\begin{array}{c} p \to r \\ q \to r \\ \hline p \vee q \to r \end{array}$
Universal Instantiation	For any $a \in D$: $\begin{array}{c} \forall x \in D,\ P(x) \\ \hline P(a) \end{array}$
Universal Generalization	For an **arbitrary** $x \in D$: $\begin{array}{c} P(x) \\ \hline \forall x \in D,\ P(x) \end{array}$
Existential Instantiation	For a **new** $w \in D$: $\begin{array}{c} \exists x \in D,\ P(x) \\ \hline P(w) \end{array}$
Existential Generalization	For any $a \in D$: $\begin{array}{c} P(a) \\ \hline \exists x \in D,\ P(x) \end{array}$

Regular Expressions

Notation	Meaning
	most characters match themselves
.	matches an arbitrary character
\b	matches a word boundary
\w	matches alphanumeric, including "_"
\s	matches a whitespace character
\d	matches a digit, same as [0-9]
[…]	denotes a class of characters to match
[^ …]	matches any character in that class which is not shown after the ^
\	treats the next special character as itself
()	treat a group of elements as a single element
|	matches the element on left or on right
?	matches the preceding element zero or one time
*	matches the preceding element zero or more times
+	matches the preceding element one or more times
{m,n}	matches the preceding element from m to n times

Powers of 2

0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192