Theorem 37 *For all statements* P *and* Q*,*

$$(\mathrm{P} \implies \mathrm{Q}) \implies (\neg\mathrm{Q} \implies \neg\mathrm{P}) \quad .$$

PROOF:

Proof by contradiction

The strategy for proof by contradiction:

To prove a goal P by contradiction is to prove the equivalent statement $\neg P \implies \mathbf{false}$

Proof by contradiction

The strategy for proof by contradiction:

To prove a goal P by contradiction is to prove the equivalent statement $\neg P \implies \mathbf{false}$

Proof pattern:

In order to prove

$$P$$

1. Write: We use proof by contradiction. So, suppose P is false.
2. Deduce a logical contradiction.
3. Write: This is a contradiction. Therefore, P must be true.

Scratch work:

Before using the strategy

Assumptions	Goal
	P
$\vdots$	

After using the strategy

Assumptions	Goal
	contradiction
$\vdots$	
$\neg P$	

Theorem 38 *For all statements* P *and* Q,

$$(\neg \mathrm{Q} \implies \neg \mathrm{P}) \implies (\mathrm{P} \implies \mathrm{Q}) \quad .$$

PROOF:

Lemma 40 *A positive real number* x *is rational iff*

$$\begin{array}{l} \exists\, \textit{positive integers}\ m, n : \\ \qquad x = m/n \ \wedge\ \neg(\exists\, \textit{prime}\ p :\ p \mid m \ \wedge\ p \mid n) \end{array} \tag{$\dagger$}$$

PROOF:

Numbers

Objectives

- Get an appreciation for the abstract notion of number system, considering four examples: natural numbers, integers, rationals, and modular integers.
- Prove the correctness of three basic algorithms in the theory of numbers: the division algorithm, Euclid's algorithm, and the Extended Euclid's algorithm.
- Exemplify the use of the mathematical theory surrounding Euclid's Theorem and Fermat's Little Theorem in the context of public-key cryptography.
- To understand and be able to proficiently use the Principle of Mathematical Induction in its various forms.

Natural numbers

In the beginning there were the *natural numbers*

$$\mathbb{N} \; : \; 0 \, , \; 1 \, , \; \ldots \, , \; n \, , \; n+1 \, , \; \ldots$$

generated from *zero* by successive increment; that is, put in ML:

```
datatype
  N = zero | succ of N
```