Why

We name sequences in $\{0, 1\}$ to easily discuss codes in this set.[1]

Definition

A *bit string* (or *binary string*) is a finite sequence in the set $\{0, 1\}$.

If a bit string has length one, we refer to it as a *bit*. Using this terminology, it is natural to call the sequence terms *bits*. Other terminology for bit strings includes *binary sring*, *bit sequence* and *digital datum*.

If the bit string has length eight, we refer to it as a *byte*. Using this terminology, a *kilobyte* is a length $8 \cdot 2^{10}$ bit string. In other words, a kilobyte is $2^{10} = 1024$ bytes, or roughly one thousand bytes. Likewise a *megabyte* is a length $8 \cdot 2^{20}$ bit string. A megabyte is $2^{20} = 1048576$ bytes, or roughly one million bytes. Similarly a *gigabyte* is 2^{30} bytes and a *terabyte* is 2^{40} bytes.[2]

Notation

We often denote the set $\{0, 1\}$ by $\mathbf{B}$. Using this notation, we denote the length n bit strings by $\mathbf{B}^n$.

We occasionally use `false` to denote the length-1 bit string $(0,)$ and `true` to denote the length-1 bit string $(1,)$. In this context, `bool` is another name for the set $\{\texttt{true}, \texttt{false}\}$[3] .

---

[1]Why we want to name these requires more explanation, to be included in future editions.

[2]Warning: some authors use these monikers with powers of ten. For example, a kilobyte is exactly one thousand bytes. etc.

[3]Future editions might break this out into its own sheet.

Bit Strings

Codes

Natural Powers

Dot-Dash Code

Natural Products

Flash Code

Natural Sums

Alphabets

Lists

Recursion Theorem

Direct Products

Set Numbers

Finite Sets

Equivalent Sets

Function Inverses

Natural Order

Family Unions and Intersections

Function Images

Function Composites

Families

Peano Axioms

Functions

Equivalence Relations

Relations

Natural Induction

Ordered Pair Projections

Set Products

Set Powers

Generalized Set Dualities

Natural Numbers

Partitions

Unordered Triples

Intersection of Empty Set

Successor Sets

Set Dualities

Set Unions and Intersections

Ordered Pairs

Set Intersections

Pair Unions

Set Complements

Set Unions

Set Differences

Pair Intersections

Unordered Pairs

Empty Set

Set Specification

Set Equality

Set Inclusion

Deductions

Quantified Statements

Logical Statements

Statements

Identities

Sets

Names

Letters

Objects