Why

Can we generalize the idea of flash codes.[1]

Definition

Let X be a set and A be an alphabet set. We denote the set of all finite sequences (strings) in A by $\mathsf{str}(A)$. We read $\mathsf{str}(A)$ aloud as "the strings in A." The length zero string is $\varnothing$.

A *code* for X *in* A is a function from X to $\mathsf{str}(A)$. In this context, we refer to the finite set A as an *alphabet* and we call $c(x)$ the *codeword* of x. The *length* of $x \in X$, with respect to a code $c : X \to \mathsf{str}(A)$, is the length of the sequence $c(x)$ (its codeword). We call a code *nonsingular* if it is injective.

Examples

Define $c : \{\alpha, \beta\} \to \{0, 1\}$ by $c(\alpha) = (0,)$ and $c(\beta) = (1,).$[2]

Code extensions

Let $s, t \in \mathsf{str}(A)$ of length m and n respectively. The *concatenation* of s with t is the length $m+n$ string $u \in \mathsf{str}(A)$ defined by $u_1 = s_1, \ldots, u_m = s_m$ and $u_{m+1} = t_1, \ldots, u_{m+n} = t_n$. We denote the concatenation of s and t by st. Note, however, that $st \neq ts$, although $s(tr) = (st)r$.

---

[1] The reliance of this sheet on Flash Codes and Dot-Dash Codes is for this justification, and not for any of the terms presented.

[2] Future editions will include additional examples.

Given a code $c : X \to \mathsf{str}(A)$, we can produce a code for $\mathsf{str}(X)$ in a natural way. The *extension* of c is the function $C : \mathsf{str}(X) \to \mathsf{str}(A)$ defined, for $\xi = (\xi_1, \ldots, \xi_n) \in \mathsf{str}(X)$, by

$$C(\xi) = c(\xi_1) \cdots c(\xi_n).$$

We call an code *uniquely decodable* if its extension is injective. In other words, given the code $C(\xi)$ for a sequence $\xi \in \mathsf{str}(X)$, we can recover ξ. We call $C(\xi)$ the *encoding* of ξ. We call ξ the *decoding* of $C(\xi)$.

Codes

Dot-Dash Code

Flash Code

Alphabets

Lists

Direct Products — Set Numbers

Finite Sets

Equivalent Sets

Family Unions and Intersections — Natural Order — Function Inverses

Families — Function Composites — Function Images

Peano Axioms — Functions — Equivalence Relations

Natural Induction — Relations

Set Products

Natural Numbers — Set Powers

Generalized Set Dualities — Intersection of Empty Set — Successor Sets — Unordered Triples — Partitions

Set Dualities — Set Unions and Intersections — Pair Unions — Set Intersections — Ordered Pairs

Set Complements

Set Differences — Pair Intersections — Set Unions

Empty Set — Unordered Pairs

Set Specification

Set Equality

Set Inclusion

Standardized Accounts

Accounts

Deductions

Quantified Statements

Logical Statements

Statements

Identities — Sets

Names

Letters — Objects