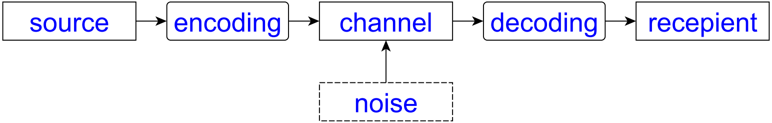
Teaching Mathematics and Statistics in Sciences: Modeling and Computer-Aided Approach
IPA HUSRB/0901/221/088

Boolean Functions and Coding Theory

Eszter K. Horváth, Branimir Šešelja, Andreja Tepavčević



University of Szeged 2011 University of Novi Sad

Introduction

Boolean algebra (named by English mathematician George Boole) appeared in the mid-19th century as the "mathematics of logic."

As we shall see, Boolean logic is turned into logic gates on the chip, and logic circuits actually perform functions such as addition and multiplication.

Introduction

Boolean algebra (named by English mathematician George Boole) appeared in the mid-19th century as the "mathematics of logic."

As we shall see, Boolean logic is turned into logic gates on the chip, and logic circuits actually perform functions such as addition and multiplication.

Boolean algebra

Boolean algebra is an ordered 6-tuple $\mathcal{B} = (B, \wedge, \vee, {}', 0, 1)$, where B is a non-empty set, $\wedge$ (meet) and $\vee$ (join) are binary, $'$ (complement) is unary operation, and 0 and 1 are constants, so that the following axioms are satisfied:

b1: $x \wedge y = y \wedge x$

b2: $x \vee y = y \vee x$; (*commutation laws*)

b3: $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

b4: $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ (*distributivity laws*)

b5: $x \wedge 1 = x$

b6: $x \vee 0 = x$ (*properties of 0 and 1*)

b7: $x \wedge x' = 0$

b8: $x \vee x' = 1$ (*complement properties*)

b9: $0 \neq 1$.

Examples

a) For $A \neq \emptyset$ and $\mathcal{P}(A) = \{X \mid X \subseteq A\}$, we have the power set Boolean algebra

$$(\mathcal{P}(A), \cap, \cup, ', \emptyset, A).$$

b) If n is a square free positive integer (i.e., having the factorization $p_1 \cdot \ldots \cdot p_n$, all primes being different), then the collection of all its divisors is a Boolean algebra:

$$(D(n), \ \mathrm{nzd}, \mathrm{nzs}, n/x, 1, n).$$

Examples

a) For $A \neq \emptyset$ and $\mathcal{P}(A) = \{X \mid X \subseteq A\}$, we have the power set Boolean algebra

$$(\mathcal{P}(A), \cap, \cup, ', \emptyset, A).$$

b) If n is a square free positive integer (i.e., having the factorization $p_1 \cdot \ldots \cdot p_n$, all primes being different), then the collection of all its divisors is a Boolean algebra:

$$(D(n),\ \mathrm{nzd}, \mathrm{nzs}, n/x, 1, n).$$

Examples

c) By Axiom b9, the smallest Boolean algebra is a two-element one:

$$\mathcal{B}_2 = (\{0,1\}, \wedge, \vee, {}', 0, 1),$$

where the operations are given by the tables:

$\wedge$	1	0
1	1	0
0	0	0

$\vee$	1	0
1	1	1
0	1	0

	$'$
1	0
0	1

Properties of Boolean Algebras

If F is a statement about Boolean algebras, than its **dual** is obtained by replacing each appearance of $\wedge$ by $\vee$ and vice versa, as well as each appearance of 0 by 1 and vice versa.

Axioms are given in dual pairs (except the last one, which is self-dual). Therefore, it is obvious that the following meta-theorem holds in the class of Boolean algebras.

Duality principle for Boolean algebras: *If some statement follows from the axioms* $\mathrm{b1}$ - $\mathrm{b9}$*, then also its dual can be deduced from these axioms.*

Properties of Boolean Algebras

If F is a statement about Boolean algebras, than its **dual** is obtained by replacing each appearance of $\wedge$ by $\vee$ and vice versa, as well as each appearance of 0 by 1 and vice versa.

Axioms are given in dual pairs (except the last one, which is self-dual). Therefore, it is obvious that the following meta-theorem holds in the class of Boolean algebras.

Duality principle for Boolean algebras: *If some statement follows from the axioms* $\mathrm{b1}$ - $\mathrm{b9}$, *then also its dual can be deduced from these axioms.*

Properties of Boolean Algebras

If F is a statement about Boolean algebras, than its **dual** is obtained by replacing each appearance of $\wedge$ by $\vee$ and vice versa, as well as each appearance of 0 by 1 and vice versa.

Axioms are given in dual pairs (except the last one, which is self-dual). Therefore, it is obvious that the following meta-theorem holds in the class of Boolean algebras.

Duality principle for Boolean algebras: *If some statement follows from the axioms* $\mathrm{b1}$ - $\mathrm{b9}$*, then also its dual can be deduced from these axioms.*

Properties of Boolean Algebras

Theorem

In every Boolean algebra, the following identities are satisfied:
a) $x \wedge 0 = 0$;
b) $x \vee 1 = 1$;
c) $x \wedge (x \vee y) = x$;
d) $x \vee (x \wedge y) = x$
e) $x \wedge x = x$;
f) $x \vee x = x$.

Properties of Boolean Algebras

Proof a) By the axioms we have

$$\begin{aligned} x \wedge 0 &= (x \wedge 0) \vee 0 = (x \wedge 0) \vee (x \wedge x') \\ &= x \wedge (0 \vee x') = x \wedge (x' \vee 0) = x \wedge x' = 0. \end{aligned}$$

b) Dually to a).

Properties of Boolean Algebras

c) By the axioms and by a) we have

$$x \wedge (x \vee y) = (x \vee 0) \wedge (x \vee y) = x \vee (0 \wedge y) = x \vee 0 = x.$$

d) Dually to c).
e) By the absorption law,

$$x = x \wedge (x \vee (x \wedge x)) = x \wedge x.$$

f) is dual to e).

Properties of Boolean Algebras

Lemma 2 If for some t in a Boolean algebra we have

$$y \vee t = z \vee t \;\text{ i }\; y \vee t' = z \vee t',$$

then $y = z$.

Proof Axioms imply

$$\begin{aligned} y &= y \vee 0 = y \vee (t \wedge t') = (y \vee t) \wedge (y \vee t') \\ &= (z \vee t) \wedge (z \vee t') = z \vee (t \wedge t') = z \vee 0 = z. \end{aligned}$$

Properties of Boolean Algebras

Lemma 2 If for some t in a Boolean algebra we have

$$y \vee t = z \vee t \text{ i } y \vee t' = z \vee t',$$

then $y = z$.

Proof Axioms imply

$$\begin{aligned} y &= y \vee 0 = y \vee (t \wedge t') = (y \vee t) \wedge (y \vee t') \\ &= (z \vee t) \wedge (z \vee t') = z \vee (t \wedge t') = z \vee 0 = z. \end{aligned}$$

Properties of Boolean Algebras

Theorem 3

In every Boolean algebra the following identities are satisfied:
a) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$;
b) $x \vee (y \vee z) = (x \vee y) \vee z$.

Proof

The previous lemma is applied by replacing y with $x \wedge (y \wedge z)$, z with $(x \wedge y) \wedge z$, and t with x; then we have

$$\begin{aligned} (x \wedge (y \wedge z)) \vee x &= x \quad \text{and} \\ ((x \wedge y) \wedge z) \vee x &= ((x \wedge y) \vee x) \wedge (z \vee x) \\ &= x \wedge (z \vee x) = x. \end{aligned}$$

Properties of Boolean Algebras

Theorem 3

In every Boolean algebra the following identities are satisfied:
a) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$;
b) $x \vee (y \vee z) = (x \vee y) \vee z$.

Proof

The previous lemma is applied by replacing y with $x \wedge (y \wedge z)$, z with $(x \wedge y) \wedge z$, and t with x; then we have

$$\begin{aligned}(x \wedge (y \wedge z)) \vee x &= x \quad \text{and} \\ ((x \wedge y) \wedge z) \vee x &= ((x \wedge y) \vee x) \wedge (z \vee x) \\ &= x \wedge (z \vee x) = x.\end{aligned}$$

Properties of Boolean Algebras

Similarly,

$$\begin{aligned}
(x \wedge (y \wedge z)) \vee x' &= (x \vee x') \wedge ((y \wedge z) \vee x') \\
&= 1 \wedge ((y \wedge z) \vee x') \\
&= (y \wedge z) \vee x', \quad \text{and} \\
((x \wedge y) \wedge z) \vee x' &= ((x \wedge y) \vee x') \wedge (z \vee x') \\
&= ((x \vee x') \wedge (y \vee x')) \wedge (z \vee x') \\
&= (1 \wedge (y \vee x')) \wedge (z \vee x') \\
&= (y \vee x') \wedge (z \vee x') = (y \wedge z) \vee x'.
\end{aligned}$$

Properties of Boolean Algebras

Hence,

$$(x \wedge (y \wedge z)) \vee x = ((x \wedge y) \wedge z) \vee x \quad \text{and} \quad (x \wedge (y \wedge z)) \vee x' = ((x \wedge y) \wedge z) \vee x',$$

and by Lemma 2,

$$x \wedge (y \wedge z) = (x \wedge y) \wedge z.$$

b) Dually.

Properties of Boolean Algebras

The following can be easily proved.

Theorem 4

In every Boolean algebra, the following identities are satisfied:
a) $(x')' = x$; (involution)
b) $(x \wedge y)' = x' \vee y'$;
c) $(x \vee y)' = x' \wedge y'$. (De Morgan laws)

Order

An ordering relation (reflexive, antisymmetric and transitive) is introduced on Boolean algebras as follows:

$$x \leq y \text{ if and only if } x \wedge y = x$$

The proof that this is indeed an ordering relation is straightforward.

Order

Theorem 5 The following statements for Boolean algebras are equivalent:
$a)$ $x \leqslant y$;
$b)$ $x \vee y = y$;
$c)$ $x \wedge y' = 0$;
$d)$ $x' \vee y = 1$.
Proof If $a)$ holds, then $x \vee y = (x \wedge y) \vee y = y$, by absorption law, proving b), and similarly other way round.
$a) \Rightarrow c)$:

$$x \wedge y' = x \wedge y \wedge y' = x \wedge 0 = 0.$$

Order

$c) \Rightarrow a)$:

$$x \wedge y = (x \wedge y) \vee 0 = (x \wedge y) \vee (x \wedge y') = x \wedge (y \vee y') = x \wedge 1 = x.$$

Equivalence of $b)$ and $d)$ is proved analogously.
The proof is ready.

As usual, we define the relation $<$:

$$x < y \text{ if and only if } x \neq y \text{ and } x \leq y.$$

Order

Due to the ordering relation, finite Boolean algebras, like all other finite ordered structures, can be represented by a diagram, so called **Hasse-diagram**. Elements of the underlying set are represented by points (circles) in a plain; points x and y are connected by an upwards oriented line from x to y if and only if $x < y$, and there is no z such that $x < z$ and $z < y$.

Examples

Boolean algebras with 2, 4, 8 and 16 elements can be drawn by diagrams. Actually these can be considered as power set algebras $\mathcal{P}(A)$, A having 1, 2, 3 and 4 elements, respectively.

Direct power algebras

The two-element Boolean algebra is denoted by $\mathcal{B}_2$. If the power $\{0,1\}^n$, $n \in \mathbb{N}$ is taken as the underlying set, and the operations are defined componentwise, then Boolean algebra $\mathcal{B}_2^n$ is obtained (the proof is straightforward:

$$\mathcal{B}_2^n = (\{0,1\}^n, \wedge, \vee, {}', \mathbf{0}, \mathbf{1}),$$

where for two ordered n-tuples $(a_1, \dots, a_n)$ and $(b_1, \dots, b_n)$ we have

$$(a_1, \dots, a_n) \wedge (b_1, \dots, b_n) := (a_1 \wedge b_1, \dots, a_n \wedge b_n),$$

and the operation $\wedge$ on the right is the one in $\mathcal{B}_2$.

Direct power algebras

Analogously we define the operation $\vee$, and further

$$(a_1, \dots, a_n)' := (a_1', \dots, a_n'), \;\; \mathbf{0} := (0, \dots, 0), \;\; \mathbf{1} := (1, \dots, 1).$$

The order on $\mathcal{B}_2^n$ is given by

$$(a_1, \dots, a_n) \leq (b_1, \dots, b_n) \text{ if and only if } a_1 \leq b_1, \dots, a_n \leq b_n,$$

and this follows directly from the definition of the order on Boolean algebras.

Isomorphism. Representation of finite Boolean algebras

A function f from Boolean algebra $\mathcal{B}$ into a Boolean algebra $\mathcal{C}$ is a **homomorphism** if the following hold:

$$\begin{aligned} f(x \wedge y) &= f(x) \wedge f(y); \\ f(x \vee y) &= f(x) \vee f(y); \\ f(x') &= (f(x))'; \\ f(0) &= 0; \\ f(1) &= 1. \end{aligned}$$

Isomorphism. Representation of finite Boolean algebras

Homomorphism from $\mathcal{B}$ to $\mathcal{C}$ which is a bijection is said to be an **isomorphism**.

We need the following notion. An element a of a Boolean algebra $\mathcal{B}$ is an **atom** in $\mathcal{B}$, if $a \neq 0$, and $0 \leq x \leq a$ implies $x = 0$ or $x = a$.

Boolean algebra $\mathcal{B}$ is said to be **atomic**, if for every $x \neq 0$, there is an atom a, such that $a \leq x$ (under each non-zero element there is an atom).

Isomorphism. Representation of finite Boolean algebras

Lemma 6 Every finite Boolean algebra is atomic.

The following is the **Representation theorem** for finite Boolean algebras.

Theorem 7 Every finite Boolean algebra $\mathcal{B}$ is isomorphic to the power set Boolean algebra $\mathcal{P}(A)$, where A is the set of atoms in $\mathcal{B}$.

Isomorphism. Representation of finite Boolean algebras

Corollary 8 Every finite Boolean algebra has 2^n elements, where n is the number of its atoms.

Corollary 9 Any two Boolean algebras with the same number of elements are isomorphic.

Isomorphism. Representation of finite Boolean algebras

Finite Boolean algebras can be represented also as follows.

Theorem 10 For a Boolean algebra $\mathcal{P}(A)$, where $A = \{a_1, \dots, a_n\}$, the following is satisfied:

$$\mathcal{P}(A) \cong \mathcal{B}_2^n.$$

Since every finite Boolean algebra is isomorphic to a power set algebra we have the following consequence.

Corollary 11 Every finite Boolean algebra is isomorphic to the direct power of a two-element algebra.

Boolean terms and term-functions

Boolean terms are defined recursively:

1. Variables $x, y, z, \dots$ and constants $0, 1$ are Boolean terms;

2. If A and B are Boolean terms, then also $(A \wedge B)$, $(A \vee B)$ and (A') are Boolean terms;

3. Boolean terms are only those expressions which can be obtained by applying the above two rules finitely many times.

By an additional agreement, external parentheses are omitted.

Boolean terms and term-functions

All previously listed expressions, including those by which axiom identities are formed, are Boolean terms in the sense of the above definition. In addition, according to associativity laws, the following expressions are also considered to be Boolean terms:

$$x_1^{\alpha_1} \wedge x_2^{\alpha_2} \wedge \cdots \wedge x_n^{\alpha_n}, \qquad x_1^{\alpha_1} \vee x_2^{\alpha_2} \vee \cdots \vee x_n^{\alpha_n},$$

where $x_1, \ldots, x_n$ are distinct variables and $\alpha_i \in \{0, 1\}$, with

$$x^{\alpha} := \begin{cases} x & \text{for} \quad \alpha = 1 \\ x' & \text{for} \quad \alpha = 0. \end{cases}$$

Boolean terms and term-functions

Due to the analogy with statement forms in logic, the above terms are also called **elementary conjunction** and **elementary disjunction**, respectively. An elementary conjunction (disjunction) is said to be **canonic** with respect to variables $x_1, \ldots, x_n$ if all these variables appear in the term.

Following mentioned analogy with statement forms, Boolean term of the form $k_1 \vee \cdots \vee k_m$, k_i being elementary conjunctions, is said to be a **disjunctive form**, abbreviated as DF. If all elementary conjunctions in a DF are canonic with respect to variables $x_1, \ldots, x_n$, then this disjunctive form is **canonic**, briefly CDF.

Terms which are defined dually to DF and CDF are respectively **conjunctive form** and **canonic conjunctive form**, denoted by CF and CCF.

Examples

Boolean terms are e.g.:

$$x' \wedge (y \vee z')', \;\; (x \vee (y \wedge x'))' \vee z, \;\; 1 \wedge x', \;\; ((u' \vee v)' \wedge (u' \wedge v')) \vee v.$$

The first and the second are *ternary*, the third is *unary*, and the last one is a *binary* term.

Examples

Terms x, $x \wedge z'$, $x \wedge y \wedge u' \wedge v'$ are elementary conjunctions, and x, $x \vee y'$, $x' \vee y \vee z'$ elementary disjunctions. With respect to x, y, z elementary conjunctions $x \wedge y \wedge z$, $x \wedge y' \wedge z'$ are canonic; with respect to the same variables, the term $x' \vee y' \vee z$ is a canonic disjunctive form (CDF). Examples of DF are $(x \wedge y') \vee z, \quad (x' \wedge y) \vee (x \wedge z) \vee (x' \wedge y \wedge z')$. With respect to x, y, a CDF is e.g., $(x \wedge y') \vee (x' \wedge y)$, and with regard to x, y, z, a CDF is

$$(x \wedge y \wedge z) \vee (x' \wedge y \wedge z') \vee (x \wedge y' \wedge z') \vee (x' \wedge y' \wedge z').$$

Equivalent terms

Boolean terms u and v are **equivalent** if the identity $u = v$ can be deduced from axioms for Boolean algebras.
Using induction, one can prove the following.

Theorem 12 For every Boolean term $t(x_1, \ldots, x_n)$ there is an equivalent term f which is CDF with respect to variables $x_1, \ldots, x_n$.

Example

If

$$f(x, y, z) = (x \vee y')' \vee z',$$

then the corresponding CDF is constructed as follows:

$$\begin{aligned}
&(x \vee y')' \vee z' = \\
&(x' \wedge y) \vee z' = \\
&(x' \wedge y \wedge (z \vee z')) \vee (z' \wedge (x \vee x') \wedge (y \vee y')) = \\
&(x' \wedge y \wedge z) \vee (x' \wedge y \wedge z') \vee (z' \wedge x \wedge y) \vee (z' \wedge x \wedge y') \vee \\
&(z' \wedge x' \wedge y) \vee (z' \wedge x' \wedge y') = \\
&(x' \wedge y \wedge z) \vee (x' \wedge y \wedge z') \vee (x \wedge y \wedge z') \vee (x \wedge y' \wedge z') \vee (x' \wedge y' \wedge z').
\end{aligned}$$

Term functions

Each Boolean term $f(x_1, \ldots, x_n)$ determines on an arbitrary Boolean algebra $\mathcal{B}$ a **term function** $B^n \to B$, also called a **Boolean function**, $f_{\mathcal{B}}(x_1, \ldots, x_n)$: variables are replaced by elements from B, and then operations corresponding in $\mathcal{B}$ to $\wedge, \vee$ i $'$ are applied. For a two-element Boolean algebra $\mathcal{B}_2$ the converse also holds.

Theorem 13 Let $\varphi : \{0,1\}^n \to \{0,1\}$ be a function (n-ary operation) on $\mathcal{B}_2$. The there is a Boolean term $g(x_1, \ldots, x_n)$, such that the corresponding term function $f_{\mathcal{B}_2}$ coincides with φ.

Term functions

The sketch of the proof. Consider the Boolean term
$g(x_1, \ldots, x_n) = \bigvee_{(\alpha_1,\ldots,\alpha_n)\in\{0,1\}^n}(\varphi(\alpha_1, \ldots, \alpha_n) \wedge x_1^{\alpha_1} \wedge \cdots \wedge x_n^{\alpha_n})$,
since $\alpha \in \{0, 1\}$ $x^0 = x'$, and $x^1 = x$.
For e.g., $n = 2$, the formula is developed as follows:

$$\begin{aligned} g(x_1, x_2) \quad = \quad & (\varphi(0,0) \wedge x_1' \wedge x_2') \vee (\varphi(0,1) \wedge x_1' \wedge x_2) \vee \\ & (\varphi(1,0) \wedge x_1 \wedge x_2') \vee (\varphi(1,1) \wedge x_1 \wedge x_2). \end{aligned}$$

Values $\varphi(\alpha_1, \ldots, \alpha_n)$ are taken from the set $\{0, 1\}$. It is not difficult to check that the term function $g_{\mathcal{B}_2}$ (determined by the term above) coincides with the function φ.

Term functions

The term defined above can be transformed to an equivalent CDF, due to the following theorem.

Theorem 14 Let $\varphi : \{0,1\}^n \to \{0,1\}$ be an operation on $\mathcal{B}_2$, which is not constantly equal 0 and g the corresponding term defined above. Then g is equivalent with CDF
$f(x_1, \ldots, x_n) = \bigvee_{\varphi(\alpha_1,\ldots,\alpha_n)=1}(x_1^{\alpha_1} \wedge \cdots \wedge x_n^{\alpha_n})$,
If φ is a zero function then the corresponding term is e.g.,

$$f(x_1, \ldots, x_n) = x_1 \wedge x_1'.$$

Term functions

Theorem 15

If u and v are Boolean terms, then the identity $u = v$ holds on every Boolean algebra if and only if this identity is satisfied on a two-element Boolean algebra $\mathcal{B}_2$.

Remark. The above theorem provides an answer to so called **Word problem** for Boolean algebras. There is an effective algorithm to check whether an identity $u = v$ is satisfied on every Boolean algebra: one should check it on $\mathcal{B}_2$, using e.g., tables of values.

Example

By Theorem, elementary conjunctions in the corresponding Boolean term $f(x, y, z)$ apply precisely to the rows in which the function φ has value 1. It is straightforward to check that the function $f_{\mathcal{B}_2}$ coincides with φ.

x	y	z	$\varphi(x, y, z)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$$f(x, y, z) = (x' \wedge y' \wedge z') \vee (x' \wedge y \wedge z) \vee (x \wedge y' \wedge z).$$

Operations on the set $\{0, 1\}$

Theorem 16 The number of different n-ary operations on the set $\{0, 1\}$ is 2^{2^n}.

By Theorem, every operation on $\{0, 1\}$ (not only the above given unary and binary ones, but every n-ary for each $n \in \mathbb{N}$) IS Boolean, i.e., for each of these there is a corresponding Boolean term.

By Theorem, all n-ary operations on $\{0, 1\}$, can be expressed (using superposition) only by g_3, f_2 and f_8, since these symbols correspond respectively to $'$, $\wedge$ i $\vee$ in CDF. Due to De Morgan laws, each of two binary operations (f_2 and f_8) can be omitted, and the remaining two can be used to express all others.

Minimization problem

Boolean functions appearing in real life problems usually have many variables (often more than 10). Therefore the corresponding Boolean terms are complicate and long (as expressions). It is necessary to prevent or minimize appearance of errors and to make the usage of these terms as fast as possible. In addition, we have to enable construction of very small objects (like chips) preforming the corresponding operations, and to lower their cost. Therefore, we have to simplify these terms. In some sense, we need to minimize the number of occurrences of variables and sub-terms in the Boolean term we are handling. Since (canonic) disjunctive forms are representatives of all Boolean forms, our task is to solve this minimization problem within the class of disjunctive forms.

Minimization problem

In order to simplify our notation, in the following we use the sign $\cdot$ instead of $\wedge$ in all Boolean terms.
Now let F be a Boolean term which is a disjunctive form. We denote by
v_F - the number of all appearances of variables in F and
c_F - the number of all appearances of elementary conjunctions in F.
Let F_1 i F_2 be Boolean terms being DF. We say that F_1 is **more simple** than F_2, if $v_{F_1} \leq v_{F_2}$ and $k_{F_1} \leq k_{F_2}$, and at least one inequality is strict ($<$). A disjunctive form ϕ is **minimal** disjunctive form of a term F, if $\phi = F$ and there is no DF which is more simple than ϕ and equal to F.
Boolean term may have several minimal DFs.

Examples

(a) If

$$F(x, y, z) = xy \vee x'z \vee xz'$$

then $v_F = 6$, and $k_F = 3$.

(b)

$$\begin{array}{rcl} F_1(x, y, z) & = & xyz' \vee xyz \qquad v_{F_1} = 6 \quad k_{F_1} = 2 \\ F_2(x, y, z) & = & xy \vee z' \vee x'y \quad v_{F_2} = 5 \quad k_{F_2} = 3 \\ F_3(x, y, z) & = & x'y' \vee xy' \qquad v_{F_3} = 4 \quad k_{F_3} = 2. \end{array}$$

The term F_3 more simple than F_1 as well as then F_2, and F_1, F_2 are not comparable in this sense.

Examples

(c) Boolean term $\phi(x) = x$ is a minimal disjunctive form for the term $F(x, y) = x \vee xy$, since $x = x \vee xy$, and there is no DF which is more simple than x, being equal to F.

Field **GF(2)**

Operations f_7 and f_2 are given by the tables, and denoted respectively by "$\oplus$" and " $\cdot$ ". These can be used in a particular way to express all other operations on the set $\{0, 1\}$.

$\oplus$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

Field GF(2)

Theorem 17 The structure $(\{0, 1\}, \oplus, \cdot)$ is a field, denoted by $\mathsf{GF}(2)$ (Galois Field, according to the name of the French mathematician from 18th century, Everisto Galois).

To construct polynomials over the field $\mathsf{GF}(2)$, we use its properties

$$x \oplus x = 0 \quad \text{and} \quad x^2 = x \cdot x = x,$$

as well as distributivity laws. An arbitrary polynomial with one variable

$$a_n x^n \oplus a_{n-1} x^{n-1} \oplus \cdots \oplus a_1 x \oplus a_0$$

reduces to

$$g(x) = (a \cdot x) \oplus b, \quad a, b \in \{0, 1\}.$$

Field **GF(2)**

Similarly, general form of a polynomial with two variables is

$$f(x,y) = (a \cdot x \cdot y) \oplus (b \cdot x) \oplus (c \cdot y) \oplus d, \quad a,b,c,d \in \{0,1\}.$$

Varying coefficients we get:

$$\begin{aligned}
g_1(x) &= (0 \cdot x) \oplus 0 = 0 \\
g_2(x) &= (1 \cdot x) \oplus 0 = x \\
g_3(x) &= (1 \cdot x) \oplus 1 = x \oplus 1 \\
g_4(x) &= (0 \cdot x) \oplus 1 = 1.
\end{aligned}$$

Functions corresponding to these polynomials are those 4 presented as functions on the set $\{0,1\}$.

Field **GF(2)**

Similarly there are 16 polynomials of two variables. As an example, we have

$$f(x,y) = (x \cdot y) \oplus x \oplus y$$

which is a polynomial corresponding to disjunction. Since we have also $g_3 = x \oplus 1$, it is possible to express all operations on the set f $\{0,1\}$ as polynomials over GF(2).
Functions corresponding to these polynomials are the ones presented as two variable functions on the set $\{0,1\}$.

Example

Here we determine the polynomial over GF(2), corresponding to the operation f_{10} (logical equivalence).

$$\begin{aligned}
f(0,0) &= (a \cdot 0 \cdot 0) \oplus (b \cdot 0) \oplus (c \cdot 0) \oplus d = 1 \\
f(0,1) &= (a \cdot 0 \cdot 1) \oplus (b \cdot 0) \oplus (c \cdot 1) \oplus d = 0 \\
f(1,0) &= (a \cdot 1 \cdot 0) \oplus (b \cdot 1) \oplus (c \cdot 0) \oplus d = 0 \\
f(1,1) &= (a \cdot 1 \cdot 1) \oplus (b \cdot 1) \oplus (c \cdot 1) \oplus d = 1.
\end{aligned}$$

Example

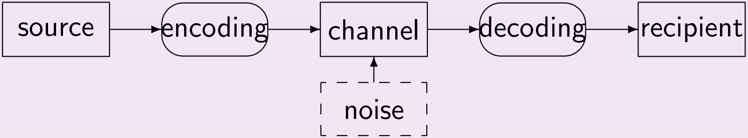
Hence

$$\begin{array}{rcl} & d & = & 1 \\ c \oplus d = 0 \ \text{ i.e., } & c & = & 1 \\ b \oplus d = 0 \ \text{ i.e., } & b & = & 1 \\ a \oplus b \oplus c \oplus d = 1, \ \text{ i.e., } & a & = & 0, \end{array}$$

and we get the polynomial

$$f(x, y) = x \oplus y \oplus 1.$$

Communication system



Probability distribution

Let $X = \{x_1, \ldots, x_n\}$ be a finite nonempty set and $p : X \to \mathbb{R}$ a function satisfying:
a) $p_i = p(x_i) \geq 0$ for $i = 1, \ldots, n$ and
b) $\sum_{i=1}^{n} p(x_i) = 1$.
The function p is a **probability distribution** over X.

Examples

A set $X = \{x_1, \dots, x_n\}$, together with a probability distribution $p(x)$ over X, is said to be a **finite probabilistic system** (shortly **system**), and it is denoted by $\{X, p(x)\}$. Elements of the set X are called **states**, and $p(x)$ should be understood as a probability for a system to be in the state x.

a) The system $\{X, p(x)\}$, where $X = \{1, 2, 3, 4, 5, 6\}$ and $p(i) = \frac{1}{6}$, $i = 1, 2, \dots, 6$, represents the outputs of a fair die rolling.

b) If X is the alphabet of some language, then relative frequencies $\frac{n_x}{n}$ of appearance of each letter x in some printed (written) text could be considered as probabilities $p(x)$. The obtained system is $\{X, p(x)\}$. □

Two-dimensional system

Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ be finite sets, and $X \times Y = \{(x, y) \mid x \in X,\ y \in X\}$ the corresponding direct product. $\{X \times Y,\ p(x, y)\}$ is a system in which $p(x, y)$ represents a probability distribution over $(x, y),\ x \in X,\ y \in Y$.

Two-dimensional system

Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ be finite sets, and $X \times Y = \{(x, y) \mid x \in X,\ y \in Y\}$ the corresponding direct product. $\{X \times Y,\ p(x, y)\}$ is a system in which $p(x, y)$ represents a probability distribution over $(x, y),\ x \in X,\ y \in Y$.

Marginal distributions are defined on X and Y by:

$$p(x_i) := \sum_{y_j \in Y} p(x_i, y_j); i = 1, \ldots, n$$

$$p(y_j) := \sum_{x_i \in X} p(x_i, y_j); j = 1, \ldots, m.$$

Two-dimensional system

If $x_i \in X$ and $p(x_i) > 0$, define for each $j \in \{1, \ldots, m\}$ $p(y_j|x_i) := \frac{p(x_i, y_j)}{p(x_i)}$. $p(y|x_i)$ is a probability distribution over Y, provided that $p(x_i)$ is given and $p(x_i) > 0$, for some fixed $x_i \in X$. It is called **conditional distribution** over Y with respect to $x_i \in X$.

Two-dimensional system

If for the system $\{X \times Y, p(x, y)\}$ we have

$$p(x_i, y_j) = p(x_i) \cdot p(y_j) \text{ for all } \; x_i \in X, \; y_j \in Y,$$

then X and Y are said to be **independent** subsystems. Otherwise, they are **dependent**.

Examples

(i)
Let $\{X, p(x)\}$ be the system connected with rolling a fair die, and $\{Y, p(y)\}$, where $Y = \{y_1, y_2\}$ and $p(y_1) = p(y_2) = \dfrac{1}{2}$, the system arising from tossing a coin.
If both (a die and a coin) are rolled, then the system $\{Y \times X,\ p(y, x)\}$ is obtained, where $p(y_i, x_j) = \dfrac{1}{12}$, for all $i \in \{1, 2\}$, $j \in \{1, \ldots, 6\}$, since $\mathrm{card}\{Y \times X\} = 12$ and all outputs are equally probable. Hence $p(y, x) = p(y)p(x)$, implying that Y and X are independent.

Examples

(ii)

Suppose now that first a fair die is rolled. If an even number appears, then an irregular coin is tossed, with the probabilities of outputs $\frac{1}{4}$ (head) and $\frac{3}{4}$ (tail). If the output is an odd number, then another irregular coin is tossed, the probabilities (in the same order) being $\frac{3}{4}$ i $\frac{1}{4}$. Both probabilities are conditional:

$$p(y_1|x_p) = \frac{1}{4},\ p(y_2|x_p) = \frac{3}{4} \quad \text{and} \quad p(y_1|x_n) = \frac{3}{4},\ p(y_2|x_n) = \frac{1}{4}\,,$$

where $x_p \in \{2, 4, 6\}$ a $x_n \in \{1, 3, 5\}$; y_1 and y_2 correspond respectively to the outputs "head" and "tail".

Examples

Since we have $p(y,x) = p(x)p(y|x)$, the distribution $p(y,x)$ of the system $Y \times X$ is given by the following table.

$Y \backslash X$	1	2	3	4	5	6	$p(y)$
y_1	$1/8$	$1/24$	$1/8$	$1/24$	$1/8$	$1/24$	$1/2$
y_2	$1/24$	$1/8$	$1/24$	$1/8$	$1/24$	$1/8$	$1/2$
$p(x)$	$1/6$	$1/6$	$1/6$	$1/6$	$1/6$	$1/6$	

Marginal distributions, $p(x)$ i $p(y)$, are also given, and it can be seen that the systems $\{X, p(x)\}$ and $\{Y, p(y)\}$ are not independent.

Examples

(iii)
Let $X = \{x_1, x_2, x_3\}$, and $Y = \{y_1, y_2\}$. The system $\{X \times Y, p(x, y)\}$ is given by its distribution

$Y \backslash X$	x_1	x_2	x_3	$p(y)$
y_1	$0,2$	$0,1$	0	$0,3$
y_2	$0,3$	$0,3$	$0,1$	$0,7$
$p(x)$	$0,5$	$0,4$	$0,1$	

Marginal distributions $p(x)$ i $p(y)$ of subsystems $\{X, p(x)\}$ and $\{Y, p(y)\}$ are also filled in.

Examples

Conditional distributions are counted as follows:

X	x_1	x_2	x_3
$p(x\|y_1)$	$\frac{2}{3}$	$\frac{1}{3}$	0

X	x_1	x_2	x_3
$p(x\|y_2)$	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{1}{7}$

Y	y_1	y_2
$p(y\|x_1)$	$\frac{2}{5}$	$\frac{3}{5}$

Y	y_1	y_2
$p(y\|x_2)$	$\frac{1}{4}$	$\frac{3}{4}$

Y	y_1	y_2
$p(y\|x_3)$	0	1

Entropy

The degree of uncertainty for the system $\{X, p(x)\}$ of being in some of its states is measured by its entropy.

The **entropy** is a function

$$H : \mathcal{D}_n \to \mathbb{R}, \;\; n \in \mathbb{N}, \quad \text{where}$$

$$\mathcal{D}_n = \{(p_1, \ldots, p_n) \in \mathbb{R}^n \mid p_i \geq 0, \; \sum_{i=1}^{n} p_i = 1\}$$

is the set of all probability distributions over an n-element set.

Entropy

The formula is given by

$$H(p_1, \ldots, p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i,$$

For a given system $\{X, p(x)\}$, where $X = \{x_1, \ldots, x_n\}$, the value

$$H(p_1, \ldots, p_n)$$

is denoted by $H(X)$.
We introduce the following notation.

$$h(n) := H\Big(\frac{1}{n}, \ldots, \frac{1}{n}\Big).$$

Entropy

Theorem 18 The function $H(p_1, \ldots, p_n)$ has the following properties:

I If for $m, n \in \mathbb{N} \quad m < n$, then $h(m) < h(n)$;

II For $m, n \in \mathbb{N}$

$$h(m \cdot n) = h(m) + h(n);$$

Entropy

Theorem 18 The function $H(p_1, \ldots, p_n)$ has the following properties:

I If for $m, n \in \mathbb{N} \quad m < n$, then $h(m) < h(n)$;

II For $m, n \in \mathbb{N}$
$$h(m \cdot n) = h(m) + h(n);$$

Entropy

III Let $1 \leq r \leq n, \quad r \in \mathbb{N}$. Divide the distribution into two parts:

$$p_1, \ldots, p_r \quad \text{i} \quad p_{r+1}, \ldots, p_n \quad \text{and denote}$$

$$q_1 = p_1 + \cdots + p_r, \quad q_2 = p_{r+1} + \cdots + p_n\,.$$

Then

$$H(p_1, \ldots, p_r, p_{r+1}, \ldots, p_n) =$$

$$H(q_1, q_2) + q_1 H\left(\frac{p_1}{q_1}, \ldots, \frac{p_r}{q_1}\right) + q_2\, H\left(\frac{p_{r+1}}{q_2}, \ldots, \frac{p_n}{q_2}\right).$$

IV $H(p, 1-p)$ is a continuous function over p, for $p \in (0,1)$.

Entropy

III Let $1 \leq r \leq n, \quad r \in \mathbb{N}$. Divide the distribution into two parts:

$$p_1, \ldots, p_r \quad \text{i} \quad p_{r+1}, \ldots, p_n \quad \text{and denote}$$

$$q_1 = p_1 + \cdots + p_r, \quad q_2 = p_{r+1} + \cdots + p_n\,.$$

Then

$$H(p_1, \ldots, p_r, p_{r+1}, \ldots, p_n) =$$

$$H(q_1, q_2) + q_1 H\left(\frac{p_1}{q_1}, \ldots, \frac{p_r}{q_1}\right) + q_2\, H\Big(\frac{p_{r+1}}{q_2}, \ldots, \frac{p_n}{q_2}\Big)\,.$$

IV $H(p, 1-p)$ is a continuous function over p, for $p \in (0,1)$.

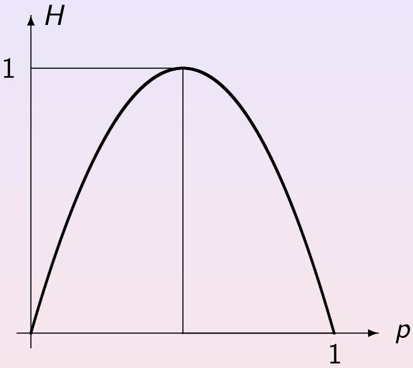
Units and properties of entropy

$$H\left(\frac{1}{2},\frac{1}{2}\right) = h(2) := 1 \ \ \textbf{bit}.$$

(bit – binary digit)

$$H(p, 1-p) = -p\log p - (1-p)\log(1-p)$$

Units and properties of entropy



Units and properties of entropy

We define $0 \cdot \log 0 := 0$, then the function $H(p_1, \dots, p_n)$ is defined for all $p \in [0, 1]$, and it is non-negative, since in this interval $p \log p \geq 0$).

Theorem 19

$$H(p_1, \dots, p_n) \leq \log n,$$

and the equality holds if and only if for all $i \in \{1, \dots, n\},\ p_i = \dfrac{1}{n}$.

Examples

(a) If a chess piece is placed randomly on the board, then the system X with 64 states is obtained: the distribution is uniform, i.e., for each state x, $p(x) = 1/64$. Therefore,

$$H(X) = h(64) = \log_2 64 = 6 \text{ bits.}$$

This is the greatest entropy value for systems with 64 states.
(b) The output of the roulette game is a system X with 36 state and the uniform distribution: for each output x (the number between 1 and 36) we have $p(x) = 1/36$. Therefore,

$$H(X) = h(36) = \log_2 36 = 5.1699 \text{ bit.}$$

Examples

(c) Let $\{X, p(x)\}$ be given by:

$$X = \begin{pmatrix} x_1 & x_2 & x_3 \\ 0,3 & 0,5 & 0,2 \end{pmatrix}.$$

Then

$$\begin{aligned} H(X) &= -0,3 \log 0,3 - 0,5 \log 0,5 - 0,2 \log 0,2 = \\ &= 0,5211 + 0,5000 + 0,4639 = 1,4850 \text{ bit.} \end{aligned}$$

Entropy of two-dimensional system

For a two-dimensional system $\{X \times Y, p(x,y)\}$, we have by the previous definition

$$H(X \times Y) = -\sum_{x \in X, y \in Y} p(x,y) \log p(x,y).$$

$$H(X) = -\sum_{x \in X} p(x) \log p(x), \quad \text{where} \quad p(x) = \sum_{y \in Y} p(x,y);$$

$$H(Y) = -\sum_{y \in Y} p(y) \log p(y), \quad \text{where} \quad p(y) = \sum_{x \in X} p(x,y);$$

Entropy of two-dimensional system

Theorem 20

$$H(X \times Y) \leq H(X) + H(Y),$$

and the equality holds if and only if X and Y are independent. We also define

$$H(X|Y) = - \sum_{x \in X, y \in Y} p(x,y) \log p(x|y)$$

$$H(Y|X) = - \sum_{x \in X, y \in Y} p(x,y) \log p(y|x) \,.$$

Theorem 21

$$H(X \times Y) = H(X) + H(Y|X) = H(Y) + H(X|Y).$$

Examples

Entropy of the system arising from rolling a die and tossing irregular coins (Example (*ii*)):

$Y \backslash X$	1	2	3	4	5	6	$p(y)$
y_1	1/8	1/24	1/8	1/24	1/8	1/24	1/2
y_2	1/24	1/8	1/24	1/8	1/24	1/8	1/2
$p(x)$	1/6	1/6	1/6	1/6	1/6	1/6	

Examples

$$H(Y \times X) = -6\left(\frac{1}{8}\log\frac{1}{8} + \frac{1}{24}\log\frac{1}{24}\right) = 3,396 \ \text{ bits.}$$

This is an average uncertainty when the die and a coin (irregular) are thrown.

$$H(X) = \log 6 = 2,585 \ \text{ bits}$$

$$H(Y) = \log 2 = 1 \ \text{ bit.}$$

The systems Y i X are not independent. Therefore the sum $H(Y) + H(X) = 3,585$ bits is greater than $H(Y \times X)$.

Information

For a system $\{X, p(x)\}$, the entropy $H(X)$ is considered to be (the measure of) its **information**, denoted by $I[X]$:

$$I[X] := -\sum_{x \in X} p(x) \log p(x).$$

For a two-dimensional system $\{X \times Y, p(x, y)\}$, we define the mutual information of its subsystems X and Y by

$$I[X, Y] = H(X) - H(X|Y).$$

By this definition we have

$$I[X, Y] := \sum_{x \in X, y \in Y} p(x, y) I[x, y] = \sum_{x \in X, y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)},$$

and also the following are satisfied.

Information

Theorem 22
(*i*) $I[X, Y] = H(Y) - H(Y|X)$;
(*ii*) $I[X, Y] = H(X) + H(Y) - H(X \times Y)$.
It follows also that

$$I[X, Y] = I[Y, X].$$

Information is measured in bits.

Example

Suppose that for some geographical place chances of rain for June 15. are 0.4, and for September 15. these chances are 0.8. Further, suppose that for June 15. weather forecast ”rain” is true in 60% of cases, and the forecast ”no rain” matches in 80% of cases; for September 15. the forecast ”rain” is true in 90%, and ”no rain” in 50% of cases.
The question is for which of two days the forecast gives more information about the weather.

Example

Denote by X_1 the system whose states are x_1- *it is raining*, $\overline{x}_1-$ *no rain*, all for June 15. Similarly, let X_2 be the corresponding system with states $x_2, \overline{x}_2$ for Septembar 15.
Since $p(x_1) = 0,4,\ p(\overline{x}_1) = 0,6$, we have

$$H(X_1) = -0,4 \log 0,4 - 0,6 \log 0,6 = 0,971 \text{ bit} .$$

Similarly, from $p(x_2) = 0,8$ and $p(\overline{x}_2) = 0,2$, we get

$$H(X_2) = 0,722 \quad \text{bit.}$$

Example

Let Y_1 the system whose states y_1 and $\overline{y}_1$ are considered respectively as *the forecast is "rain"*, and *the forcast is "no rain"* for June 15. and let Y_2 be the corresponding system with states y_2 and $\overline{y}_2$ for September 15.
Since

$$\begin{array}{ll} p(x_1|y_1) = 0.6, & p(\overline{x}_1|y_1) = 0,4, \\ p(x_1|\overline{y}_1) = 0,2, & p(\overline{x}_1|\overline{y}_1) = 0,8, \end{array}$$

and by

$$p(x_1) = p(y_1)p(x_1|y_1) + p(\overline{y}_1)p(x_1|\overline{y}_1)$$

(total probability formula), with $p(\overline{y}_1) = 1 - p(y_1)$, we obtain

$$p(y_1) = 0,5, \quad p(\overline{y}_1) = 0,5.$$

Example

Similarly, by

$$\begin{array}{ll} p(x_2|y_2) = 0,9 & p(\overline{x}_2|y_2) = 0,1, \\ p(x_2|\overline{y}_2) = 0,5 & (\overline{x}_2|\overline{y}_2) = 0,5, \end{array}$$

and

$$p(x_2) = p(y_2)p(x_2|y_2) + p(\overline{y}_2)p(x_2|\overline{y}_2)$$

we get

$$p(y_2) = 0,75, \quad p(\overline{y}_2) = 0,25.$$

Hence

$$\begin{aligned} H(X_1|Y_1) &= -0,5(0,6\log 0,6 - 0,4\log 0,4 - 0,2\log 0,2 \\ &\quad -0,8\log 0,8) = 0,846 \quad \text{bit}, \end{aligned}$$

and

$$H(X_2|Y_2) = 0,602 \quad \text{bit}.$$

Example

By

$$I[X_1, Y_1] = H(X_1) - H(X_1|Y_1),$$
$$I[X_2, Y_2] = H(X_2) - H(X_2|Y_2],$$

we finally get

$$\begin{aligned} I[X_1, Y_1] &= 0,125 \text{ bit },\\ I[X_2, Y_2] &= 0,120 \text{ bit }. \end{aligned}$$

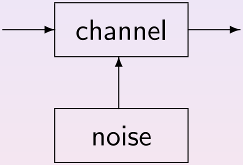
Conclusion: the forecast for June 15. provides more information, in spite of the fact that the probability of precise forecast for this day is less.

Source

Let $X = \{\alpha_1, \ldots, \alpha_a\}$ be an **alphabet**. Its elements are **letters**. Let $P = \{p_1, \ldots, p_a\}$ be a probability distribution over X. Then the system $\{X, p(x)\}$ is called **discrete memoryless source** over X (in the following we use the term *source*), and is usually denoted by (X, P).
Entropy of the above source is given by

$$H(X) = -\sum_{x \in X} p(x) \log\, p(x)\,.$$

Communication channel



Communication channel

Input alphabet: $U = \{u_1, \ldots, u_a\}$;
output alphabet: $V = \{v_1, \ldots, v_b\}$.
Channel matrix:

$$\Pi = \begin{bmatrix} p(v_1|u_1) & p(v_2|u_1) & \ldots & p(v_b|u_1) \\ p(v_1|u_2) & p(v_2|u_2) & \ldots & p(v_b|u_2) \\ \vdots & \vdots & \ldots & \vdots \\ p(v_1|u_a) & p(v_2|u_a) & \ldots & p(v_b|u_a) \end{bmatrix}$$

$$K = (U, \mathcal{P}, V),$$

K is a **discrete stationary memoryless channel** over alphabets U and V (in the sequel: *channel*).
Properties of the matrix:
1) $p(v_j|u_i) \geq 0, \quad i = 1, \ldots, a, \;\; j = 1, \ldots, b$;
2) $\sum_{j=1}^{b} p(v_j|u_i) = 1$, for every $i = 1, \ldots, a$.

Examples

(i) **Noiseless channel**

$$\Pi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$U = \{u_1, u_2, u_3\}$, a $V = \{v_1, v_2, v_3\}$,
$p(v_2|u_1) = 1,\ p(v_3|u_2) = 1$ i $p(v_1|u_3) = 1$

$$\begin{array}{ccc} u_1 & \longrightarrow & v_2 \\ u_2 & \longrightarrow & v_3 \\ u_3 & \longrightarrow & v_1\ . \end{array}$$

Examples

(ii) **Useless channel**

$$\Pi = \left[\begin{array}{ccc} 0,2 & 0,5 & 0,3 \\ 0,2 & 0,5 & 0,3 \end{array} \right]$$

$U = \{u_1, u_2\},\ V = \{v_1, v_2, v_3\},$

$$\begin{array}{ll} p(v_1|u_1) & = p(v_1|u_2), \\ p(v_2|u_1) & = p(v_2|u_2), \\ p(v_3|u_1) & = p(v_3|u_2), \end{array}$$

Examples

(*iii*)

$$\Pi = \begin{bmatrix} 0,2 & 0,1 & 0,7 \\ 0 & 0 & 1 \\ 0,5 & 0,5 & 0 \end{bmatrix}$$

Output system

Let $K = (U, \mathcal{P}, V)$ be a channel connected to the source (U, P), i.e., to the system $\{U, p(u)\}$. Then for every $v_j \in V$ we define

$$p(v_j) = \sum_{i=1}^{a} p(u_i) p(v_j | u_i). \quad \text{Hence}$$

$$\sum_{j=1}^{b} p(v_j) = \sum_{i=1}^{a} p(u_i) \sum_{j=1}^{b} p(v_j | u_i) = \sum_{i=1}^{a} p(u_i) \cdot 1 = 1 \,.$$

IN this way, a system $\{V, p(v)\}$ is obtained at the output of the channel. If $P_U = (p(u_1), \ldots, p(u_a))$ and $P_V = (p(v_1), \ldots, p(v_b))$ then

$$P_V = P_U \cdot \Pi \,.$$

Examples

(i) Let $\{U, p(u)\}$, gde je $U = \{u_1, u_2\}$, $p(u_1) = 0,2$, $p(u_2) = 0,8$, and

$$\Pi = \begin{bmatrix} 0,4 & 0,6 \\ 0,2 & 0,8 \end{bmatrix}$$

The two-element system at the output is obtained using

$$P_V = P_U \cdot \Pi \ :$$

$$(p(v_1), p(v_2)) = [0,2\ 0,8] \cdot \begin{bmatrix} 0,4 & 0,6 \\ 0,2 & 0,8 \end{bmatrix}$$

and thus $p(v_1) = 0,24, \ \ p(v_2) = 0,76.$

Examples

(*ii*) Let the channel be useless. Then we have at the output

$$p(v_j) = \sum_{i=1}^{a} p(u_i)p(v_j|u_i) = p(v_j|u_i)\sum_{i=1}^{a} p(u_i) = p(v_j|u_i),$$

since each column of the matrix contains equal elements.

Channel capacity

If the channel $K = (U, \mathcal{P}, V)$ is connected to the source (U, P), then it is possible to find
$p(u|v)$, for each $v \in V$ satisfying $p(v) > 0$:

$$p(u|v) = \frac{p(u)p(v|u)}{p(v)}$$

Indeed, a distribution is obtained:

$$\sum_{i=1}^{a} p(u_i|v) = \sum_{i=1}^{a} \frac{p(u_i)p(v|u_i)}{p(v)} = \frac{\sum_{i=1}^{a} p(u_i)p(v|u_i)}{p(v)} = \frac{p(v)}{p(v)} = 1.$$

Channel capacity

Similarly, a distribution can be defined on $U \times V$ as follows: If $(u, v) \in U \times V$, then

$$p(u, v) = p(u)p(v|u) = p(v)p(u|v),$$

and $\sum_{(u,v)\in U\times V} p(u, v) = 1$.
Therefore, starting with the channel with matrix Π and the source (U, P) we can calculate the entropies

$$H(U),\ H(V),\ H(U|V),\ H(V|U) \quad \text{and} \quad H(U \times V).$$

Also the mutual information $I[U, V]$ of the input and the output of the channel (related to the given source) can be derived.

Channel capacity

The capacity of the channel K is given by

$$C := \max_{P_U \in \mathcal{D}_a} I[U, V],$$

where $\mathcal{D}_a$ is the set of all distributions over U.
By

$$I[U, V] = H(V) - H(V|U),$$

it follows that

$$C \geq 0,$$

and the equality holds if and only if $H(V) = H(V|U)$ i.e., if U are V independent. This means that for a fixed $v \in V$ and for each $u \in U$ satisfying (at the input) $p(u) > 0$, we have

$$p(v|u) = \frac{p(u, v)}{p(u)} = \frac{p(u)p(v)}{p(u)} = p(v).$$

Channel capacity

The corresponding channel is useless and for this channel we have $C = 0$. On the other hand there are **channels without lost of information**. These are defined as ones satisfying $H(U|V) = 0$. Such a channel is (but not only) e.g., any noiseless channel.

Symmetric channels. BSC

A channel is **symmetric** if each row of its matrix is a permutation of the first row, and each column is a permutation of the first column. It has a square matrix $a \times a$ and is also said to be an a-**ary** symmetric channel. In particular, if both the input and the output alphabets are binary, the corresponding symmetric channel is called a **binary symmetric channel**, or **BSC**. Its matrix is of the form

$$\Pi = \left[\begin{array}{cc} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{array} \right] \; 0 \leq \epsilon \leq 1.$$

If $\epsilon = 0$ or $\epsilon = 1$, this is a noiseless channel, while for $\epsilon = \dfrac{1}{2}$ a useless channel is obtained.

Examples

(i) A ternary symmetric channel:

$$\Pi = \begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.3 & 0.2 & 0.5 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}$$

(ii) An example of a BSC:

$$\Pi = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$$

Capacity of symmetric channel

Theorem 23 For a symmetric a-ary channel,

$$C = \log\, a + \sum_{i=1}^{a} r_i \log r_i \ ,$$

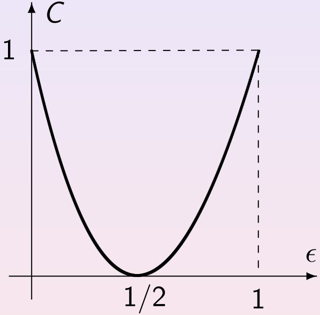
where $r_1, \ldots, r_a$ are elements of the first row of its matrix. In particular, for BSC with the matrix

$$\Pi = \left[\begin{array}{cc} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{array} \right] \ 0 \leq \epsilon \leq 1,$$

$$C = 1 + (1-\epsilon)\log(1-\epsilon) + \epsilon \log \epsilon.$$

Capacity of symmetric channel

Graphic representation of this function is as follows.



Word semigroup

A finite set X is called an **alphabet**, and its elements are **letters**.

$$X^* = X \cup X^2 \cup \cdots \cup X^n \cup \cdots = \bigcup_{i \in \mathbb{N}} X^i$$

X^* is the **set of words** over X.
Ordered $n-$tuples $(x_1, \ldots, x_n)$ from X^* are denote as words: $x_1 \cdots x_n$.
The length $|x|$ of the word $x = x_1 \cdots x_n$ is the number of its letters, i.e.,:
$|x| = n$ if and only if $x \in X^n$.
A binary operation **concatenation** is introduced on X^* as follows.
If x, y are from X^*, $x = x_1 \cdots x_n$, $y = y_1 \cdots y_m$, then

$$z = x\,y,$$

where $z = x_1 \cdots x_n\, y_1 \cdots y_m$.

Word semigroup

x is called a **prefix**, and y a **suffix** in the word z. In other words, $x \in X^*$ is a **prefix** in $z \in X^*$ if and only if there is $y \in X^*$, such that $z = xy$.
Similarly, y is a **suffix** in z if and only if there is x such that $z = xy$.
The empty set can be added to the set of words. Then it is called the **empty word**(frequently denoted by Λ). We then have

$$X^{\circledast} = X^* \cup \{\emptyset\}.$$

By definition, the empty word is a prefix and a suffix of every word x from $X^{\circledast}$:

$$\emptyset x =: x; \quad x\emptyset =: x.$$

The length of the empty word is 0, also by definition.

Word semigroup

It is straightforward to see that the operation of concatenation is not in general commutative (with the exception of the set of words over a one-element alphabet), and that it is associative. Therefore, we have the following.

Theorem 24 *a*) The ordered pair $(X^*, \cdot)$, where X^* is a set of words over X, and " $\cdot$" is a concatenation, is a semigroup (so called the semigroup of words over X).

b) Under the same conditions $(X^{\circledast}, \cdot)$ is a semigroup with unity (a monoid).

Examples

If $X = \{0, 1\}$, then some words over X are $0, 1, 00, 01, 10, 11, 000$, etc., and the corresponding lengths are respectively $1, 1, 2, 2, 2, 2, 3$, etc. The concatenation applied on $x = 100$ i $y = 02$ as words over $X = \{0, 1, 2\}$ is the word $z = 10002$. The sets of prefixes and suffixes of this word are respectively $\{1, 10, 100, 1000\}$ and $\{0002, 002, 02, 2\}$.

Coding - basic definitions

Two finite sets are given.

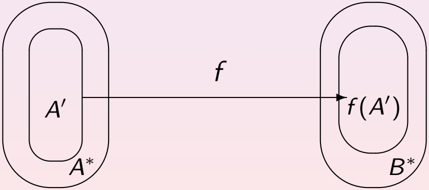
$$A = \{\alpha_1, \ldots, \alpha_a\}, \ \ a > 1, \text{ i } B = \{\beta_1, \ldots, \beta_b\} \ \ b > 1.$$

A is called the **source alphabet** and B the **code alphabet**. The number b is the **code base**.

Let $A' \subseteq A^*$. Any $1-1$ mapping (injection)

$$f : A' \to B^*$$

is a **coding** of words over the alphabet A).



Coding - basic definitions

The set $f(A') \subseteq B^*$ is a **code**, and its elements are **code-words**. A coding is said to be **alphabetic** if $A' = A$ that is, if precisely the letters from A are coded. From now on, we investigate only alphabetic codings. By the above, these are injective mappings $f : A \to B^*$, and the cod $V = f(A)$ is a set of words, the subset of B^*. Each message, the word over A is coded letter by letter. In this way the coded word over B also belongs to B^*.

A coding $f : A \to B^*$ is **with fixed code-word length** if $f(A) \subseteq B^n$, for some $n > 1$. The corresponding code is called a **block-code**.

In general, i.e., if code-words have different lengths, we have a coding **with varying code-word length**.

In general, identification of the original message from the received, coded one, is called **decoding**. Precise definition is given in the sequel.

Examples

Let $A = \{1, 2, \dots, 9, 0\}$ and $B = \{0, 1\}$. The code alphabet B is binary.

$$
a) \qquad \begin{array}{ccl}
1 & \mapsto & 10 \\
2 & \mapsto & 110 \\
3 & \mapsto & 1110 \\
 & \dots & \\
9 & \mapsto & 1111111110 \\
0 & \mapsto & 11111111110
\end{array}
$$

$$
b) \quad \begin{array}{cclccccl}
1 & \mapsto & 0001 & & 6 & \mapsto & 0110 \\
2 & \mapsto & 0010 & & 7 & \mapsto & 0111 \\
3 & \mapsto & 0011 & & 8 & \mapsto & 1000 \\
4 & \mapsto & 0100 & & 9 & \mapsto & 1001 \\
5 & \mapsto & 0101 & & 0 & \mapsto & 0000
\end{array}
$$

The code in a) have been used in telephone communication, and the cod in b) is the Binary Coded Decimal (BCD - code).

Examples

In the above mentioned telephone communication, presently the following BCD code is used[1]:

$$
c) \quad \begin{array}{lcllcl}
1 & \mapsto & 0001 & 9 & \mapsto & 1001 \\
2 & \mapsto & 0010 & 0 & \mapsto & 1010 \\
3 & \mapsto & 0011 & * & \mapsto & 1011 \\
4 & \mapsto & 0100 & \# & \mapsto & 1100 \\
5 & \mapsto & 0101 & A & \mapsto & 1101 \\
6 & \mapsto & 0110 & B & \mapsto & 1110 \\
7 & \mapsto & 0111 & C & \mapsto & 1111 \\
8 & \mapsto & 1000 & D & \mapsto & 0000
\end{array}
$$

[1] California Micro Devices, CM8870/70C, www.calmiro.com.

Unique decipherability. Prefix code

A code $V = f(A) \subseteq B^*$ is said to be **uniquely decipherable** if every word $x \in B^*$ can be in at most one way decomposed into code-words $v_{i_1}, \ldots, v_{i_k}$ from V, so that

$$x = v_{i_1} \cdots v_{i_k}.$$

A cod V is said to be a **prefix code** if neither of code-words from V is a prefix of some other code-word in V.

Example

The code $V = \{01, 11, 000, 0011, 0010\}$ is a prefix binary code. If represented by a code-tree (see the diagram), then code-words are final (end) nodes.

(In a code tree there is a branch from a node x to a node y (downwards in the diagram) if and only if x and y are prefixes of code words, and $y = \alpha y$, where α is a letter.)

Theorem 25 A prefix code is uniquely decipherable.

Kraft's inequality

Theorem 26 Let $V = \{v_1, \ldots, v_a\}$ be a prefix code over an alphabet with the base b, such that $|v_1| = n_1, \ldots, |v_a| = n_a$. Then, the inequality

$$\sum_{i=1}^{a} b^{-n_i} \leq 1,$$

is satisfied ($|v|$ denotes the length of v).
The converse:
Let $n_1, \ldots, n_a, b$ **be a sequence of** $a + 1$ **positive integer, with** $a > 1,\ b > 1$. **If these integers satisfy the inequality, then there is a prefix code** $V = \{v_1, \ldots, v_a\}$ **over an alphabet with the base** b, **such that** $|v_1| = n_1, \ldots, |v_a| = n_a$.
Formula above is called Kraft's inequality.

Example

Let $V = \{00, 01, 100, 1010, 1011\}$.
V is a prefix binary code and its code-word lengths are $2, 2, 3, 4, 4$. Kraft inequality is satisfied:

$$\frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^4} = \frac{3}{4} < 1.$$

Example

Contrary, starting with numbers
$n_1 = n_2 = 2,\ n_3 = 3,\ \ n_4 = n_5 = 4,\ b = 2,$
satisfying Kraft inequality, it is possible to construct a prefix binary code whose code-word lengths are $n_1, \dots, n_5$.
$q_1 = 0,$
$q_2 = 2^{-2},$
$q_3 = 2^{-2} + 2^{-2},$
$q_4 = 2^{-2} + 2^{-2} + 2^{-3}$ i
$q_5 = 2^{-2} + 2^{-2} + 2^{-3} + 2^{-4},$

$(q_1)_2 = 0,00; (q_2)_2 = 0,01; (q_3)_2 = 0,100; (q_4)_2 = 0,1010; (q_5)_2 = 0,1011.$

Decimal digits represent the above code V.

Conditions for unique decipherability

Not only prefix codes have a property of unique decipherability.
Let $V = \{v_1, \ldots, v_a\}$ be a code constructed over the alphabet B.
We construct a particular sequence of words from B^* :

$S_1 = \{x \in B^* \mid vx \in V, \ \text{ for some } \ v \in V\}$.

Conditions for unique decipherability

These are suffixes of code-words, with prefixes in V.

$S_2 = S_2' \cup S_2''$, where

$S_2' = \{x \in B^* \mid vx \in S_1,\ \text{for some } v \in V\}$ i

$S_2'' = \{x \in B^* \mid s_1 x \in V,\ \text{for some } s_1 \in S_1\}$.

Conditions for unique decipherability

S_2' contains suffixes of words in S_1 having code-words as prefixes. S_2'' contains suffixes of code-words having prefixes in S_1.

Similarly, for each $n \in \mathbb{N}$ we have v

$S_n = S_n' \cup S_n''$, where

$S_n' = \{x \in B^* \mid vx \in S_{n-1}, \text{ for some } v \in V\}$ and

$S_n'' = \{x \in B^* \mid s_{n-1}x \in V, \text{ for some } s_{n-1} \in S_{n-1}\}$.

Conditions for unique decipherability

Let $S := S_1 \cup S_2 \cup \dots$.

Theorem 27 A code $V = \{v_1, \dots, v_a\}$ over B ($V \subseteq B^*$) is uniquely decipherable if and only if

$$S \cap V = \emptyset.$$

Theorem 28 [McMillan's Theorem] Any uniquely decipherable code satisfies Kraft's inequality.

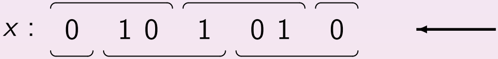
Examples

		S_1	S_2	S_3	S_4	S_5	S_6	S_7	$\dots$
U	$\{0, 10, 110, 1111\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\dots$
V	$\{10, 101, 001\}$	$\{1\}$	$\{0, 01\}$	$\{01\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\dots$
W	$\{0, 010, 101\}$	$\{10\}$	$\{1\}$	$\{01\}$	$\{1, 0\}$	$\{01, 10\}$	$\{1, 0\}$	$\{01, 10\}$	$\dots$

U is a prefix code and $S_1 = \emptyset$. Consequently all other above defined sets of suffixes are empty, and therefore $S = \emptyset$.

Examples

For the code V, $S = \{1, 0, 01\}$, therefore $S \cap V = \emptyset$, and V is uniquely decipherable.
For the code W, we have $S = \{1, 0, 10, 01\}$, $S \cap W = \{0\}$ and this is not a uniquely decipherable code. We construct a word which can be decoded in two ways:



Here we have $s_4 = 0,\ s_3 = 01,\ s_2 = 1,\ s_1 = 10\ v = 0$. The construction goes from right to left, starting with a word from $S \cap V$. Here it is 0.

Examples

Theorem 29 If $V = \{v_1, \ldots, v_a\}$ is a uniquely decipherable code over the alphabet B, then there is a prefix code $W = \{w_1, \ldots, w_a\}$ over the same alphabet B, such that $|v_1| = |w_1|, \ldots, |v_a| = |w_a|$.

Optimality: average code-word length

In the following, we are considering only uniquely decipherable codes. In addition, by Theorem before, we consider only prefix codes.
Let A be an alphabet connected with a source (A, P), where (as it is defined) $A = \{\alpha_1, \ldots, \alpha_a\}$, and $P = \{p_1, \ldots, p_a\}$ so that:

$$p(\alpha_i) = p_i \geq 0,\ i = 1, \ldots, a; \quad \sum_{i=1}^{a} p_i = 1.$$

Let (A, P) be a source and $V = f(A)$ a code over the alphabet B. The value

$$\overline{n}_V := \sum_{i=1}^{a} p_i n_i,$$

where $n_1, \ldots, n_a$ are code-word lengths, is said to be the **average code-word length** for the code V.

Optimality: average code-word length

Denote

$$n_* := \inf_V \ \overline{n}_V \, ,$$

where the infimum runs over all uniquely decipherable codes $V = f(A) \subseteq B^*$.
Every code V satisfying $\overline{n}_V = n_*$ is said to be **optimal**.

Example

Let $A = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ i $p_1 = 0,45,\ p_2 = 0,20,\ p_3 = 0,20,\ p_4 = 0,15.$ Two binary codes for this source, V and W, are given in the table.

α_i	p_i	v	$n_i(v)$	w	$n_i(w)$
α_1	0,45	00	2	0	1
α_2	0,20	01	2	10	2
α_3	0,20	10	2	110	3
α_4	0,15	11	2	111	3

Here we have

$$\begin{aligned} \overline{n}_V &= (0,45 + 0,20 + 0,20 + 0,15) \cdot 2 = 2; \\ \overline{n}_W &= 0,45 + 2 \cdot 0,20 + 3 \cdot 0,20 + 3 \cdot 0,15 = 1,90\,. \end{aligned}$$

Optimality: average code-word length

Theorem 30 Let $(A, P), |A| = a$ be a give source and B $(|B| = b)$ a code alphabet. Then for every uniquely decipherable code $V = f(A) \subseteq B^*$, the following holds:

$$\overline{n}_V \geq \frac{H}{\log b},$$

where $H = -\sum_{i=1}^{a} p_i \log p_i$ is the source entropy.
Theorem 31 Let (A, P) be a source and B $(|B| = b)$ a code alphabet. Then there is a prefix code $V = f(A) \subseteq B^*$ satisfying

$$\overline{n}_V < \frac{H}{\log b} + 1.$$

Example

$$A = \{\alpha_1, \dots, \alpha_7\},\ p_1 = p_2 = \frac{1}{2^2},\ p_3 = p_4 = p_5 = \frac{1}{2^3},\ p_6 = p_7 = \frac{1}{2^4}.$$

If $B = \{0, 1\}$, and

$$V = \{00, 01, 100, 101, 110, 1110, 1111\},$$

then $\log b = \log_2 2 = 1$ and

$$\frac{H}{\log b} = H = \overline{n}_V = 2,625.$$

For a binary coded alphabet of 7 letters this is the smallest possible average code-word length, and it is actually reached, since all probabilities are suitable powers of 2. Obviously, V is an optimal code.

Necessary conditions for optimal coding

Theorem 32 Let $(A, P),\ A = \{\alpha_1, \ldots, \alpha_p\}$ be a source and $V = f(A) \subseteq B^*,\ (|B| = b)$ an optimal code. Then

$$p_i > p_j \text{ implies } n_i \leq n_j,\ i \neq j,\ i, j \in \{1, \ldots, a\}.$$

Theorem 33 Let $V = f(A)$ be an optimal *binary* code for the source $(A, P),\ A = \{\alpha_1, \ldots, \alpha_a\}$. Suppose (without loss of generality) that

$$p_1 \geq p_2 \geq \cdots \geq p_{a-1} \geq p_a\,.$$

Then the code-words $f(\alpha_{a-1})$ and $f(\alpha_a)$ have the same length.

Theorem 34 For a given source (A, P), $A = \{\alpha_1, \ldots, \alpha_a\}$, $p_1 \geq \cdots \geq p_a$, there is an optimal binary prefix code $f(A)$, such that the code-words $f(\alpha_{a-1})$ i $f(\alpha_a)$ differ precisely in the last digit.

Shannon-Fano code

We start with a source (A, p), $A = \{\alpha_1, \dots, \alpha_a\}$, $p_1 \geq \cdots \geq p_a$. (Letters are ordered according to non-increasing probabilities).
Determine i, $1 \leq i < a$, so that $p_1 + \cdots + p_i$ and $p_{i+1} + \cdots + p_a$ are approximately equal numbers.
To each letter of the set $\{\alpha_1, \dots, \alpha_i\}$ we associate digit 0, and to each letter from $\{\alpha_{i+1}, \dots, \alpha_a\}$ digit 1.
We repeat the procedure separately for each of the sets $\{\alpha_1, \dots, \alpha_i\}$, $\{\alpha_{i+1}, \dots\ , \alpha_a\}$ (dividing each of them into two, with respect to probability approximately equal subsets, associating to each, respectively, digits 0 and 1).
This procedure is repeated up to one-element subsets.
The code word associated to each α from A, consists of digits associated to α in every step.

Example

Shannon-Fano code is determined for the source (A, P), where

$$A = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\}$$

and $p(\alpha_1) = 0,27$; $p(\alpha_2) = 0,21$; $p(\alpha_3) = 0,15$; $p(\alpha_4) = 0,15$; $p(\alpha_5) = 0,12$; $p(\alpha_6) = 0,06$; $p(\alpha_7) = 0,04$.

α_i	p_i					$v_i = f(\alpha_i)$	n_i
α_1	0, 27	0	0			0 0	2
α_2	0, 21		1			0 1	2
α_3	0, 15	1	0	0		1 0 0	3
α_4	0, 15			1		1 0 1	3
α_5	0, 12		1	0		1 1 0	3
α_6	0, 06			1	0	1 1 1 0	4
α_7	0, 04				1	1 1 1 1	4

Example

Observe that

$$\frac{H}{\log b} = \frac{H}{\log 2} = H.$$

Therefore we have

$$\begin{aligned} H &= -0,27 \log 0,27 - 0,21 \log 0,21 - 2 \cdot 0,15 \log 0,15 - 0,12 \log 0,12 \\ &\quad -0,06 \log 0,06 - 0,04 \log 0,04 = 2,6002\,. \end{aligned}$$

$$\overline{n} = 2 \cdot (0,27 + 0,21) + 3(0,15 + 0,15 + 0,12) + 4(0,06 + 0,04) = 2,62$$

Optimal code (Huffman's algorithm)

Theorem 35 Let $V = f(A) = \{v_1, \ldots, v_a\}$ be optimal binary code for the source (A, P), $|A| = a$, $P = \{p_1, \ldots, p_a\}$. If $p_j = q_0 + q_1$, where

$$p_1 \geq p_2 \geq \cdots \geq p_{j-1} \geq p_j \geq \cdots \geq p_a \geq q_0 \geq q_1.$$

Then, the code

$$V' = f(A') = \{v_1, \ldots, v_{j-1},\ v_{j+1}, \ldots, v_a, v_j0, v_j1\}$$

is optimal for the source (A', P'), where $|A'| = a + 1$ and

$$P' = \{p_1, \ldots, p_{j-1}, p_{j+1}, \ldots, p_a, q_0, q_1\}.$$

(by v_j0 and v_j1 we denote concatenation of digits $\{0, 1\}$ to the word v_j).

Examples

a) Construction of the optimal (binary) code, according to Huffman's algorithm:
Let (A, P) be a source with $A = \{\alpha_1, \ldots, \alpha_5\}$ and

$$P = \{0,35; 0,25; 0,15; 0,15; 0,10\}.$$

A	P
α_1	0,35
α_2	0,25
α_3	0,15
α_4	0,15
α_5	0,10

↱

A'	P'
α_1	0,35
α_2	0,25
α_{45}	0,25
α_3	0,15

↱

A''	P''
α_{345}	0,40
α_1	0,35
α_2	0,25

↱

A'''	P'''
α_{12}	0,60
α_{345}	0,40

Examples

A'''	$f_3(A''')$
α_{12}	$0 \searrow$
α_{345}	1

A''	$f_2(A'')$
α_{345}	$1 \searrow$
α_1	00
α_2	01

A'	$f_1(A')$
α_1	00
α_2	01
α_{45}	$10 \searrow$
α_3	11

A	$V = f(A)$
α_1	00
α_2	01
α_3	11
α_4	100
α_5	101

Examples

b) Simplified method for constructing the optimal code (also by Huffman's algorithm): :

A	P	
α_1	0,20	
α_2	0,20	
α_3	0,19	
α_4	0,12	
α_5	0,11	
α_6	0,09	0
α_7	0,09	1

↱

A'	P'	
α_1	0,20	
α_2	0,20	
α_3	0,19	
α_{67}	0,18	
α_4	0,12	0
α_5	0,11	1

↱

A''	P''	
α_{45}	0,23	
α_1	0,20	
α_2	0.20	
α_3	0,19	0
α_{67}	0,18	1

↱

A'''	P'''	
α_{367}	0,37	
α_{45}	0,23	
α_1	0.20	0
α_2	0,20	1

↱

A^{iv}	P^{iv}	
α_{12}	0,40	
α_{367}	0,37	0
α_{45}	0,23	1

↱

⟵⟵

Examples

The code V:

A	V
α_1	1 0
α_2	1 1
α_3	0 0 0
α_4	0 1 0
α_5	0 1 1
α_6	0 0 1 0
α_7	0 0 1 1

According to Theorem, V is optimal binary code for the source (A, P).

Block-codes: general

$A = \{\alpha_1, \dots, \alpha_a\}$ *source alphabet*
$B = \{\beta_1, \dots, \beta_b\}$ *code alphabet* (b is a *code base*) .
Each $1-1$ mapping $f : A \to B^n$, for some $n \in \mathbb{N}$, is a **coding of alphabet** A **with a fixed** (n) **code-word length**.
The set $V = f(A) \subseteq B^n$ is a block-code, n is its length, and if $|B| = b = 2$, the code is said to be **binary**. Equivalently, a subset V from B^n is a block-code of cardinality a, over the alphabet B.

Theorem 36 A block-code $V \subseteq B^n$, $|B| = b$, $b, n \in \mathbb{N}$, of cardinality $a \in \mathbb{N}$ exists if and only if

$$n \geq \frac{\log_2 a}{\log_2 b} .$$

In particular, if $b = 2$, the above inequality has a form

$$n \geq \log_2 a.$$

Example

For encoding digits $0, 1, 2, \ldots, 9$ by a binary code, by the above inequality it is necessary for the code-length to be at least 4, since

$$n \geq \log_2 10 \approx 3,2.$$

digit	BCD code	Gray's	code plus 3	Gray-Stibitz	Aiken's
		code		code	code
0	0000	0000	0011	0010	0000
1	0001	0001	0100	0110	0001
2	0010	0011	0101	0111	0010
3	0011	0010	0110	0101	0011
4	0100	0110	0111	0100	0100
5	0101	0111	1000	1100	1011
6	0110	0101	1001	1101	1100
7	0111	0100	1010	1111	1101
8	1000	1100	1011	1110	1110
9	1001	1101	1100	1010	1111

Algebraic structures over the set $\{0,1\}^n$

The set $\{0,1\}$ is ordered by relation $\leq$ in the usual way: $0 \leq 1$. The set $\{0,1\}^n$ of all ordered n-tuples of elements 0 i 1 is ordered componentwise:

$$(\alpha_1, \alpha_2, \ldots, \alpha_n) \leq (\beta_1, \beta_2, \ldots, \beta_n) \text{ ako } \alpha_i \leq \beta_i \text{ for all } i = 1, \ldots, n.$$

On the set $\{0,1\}$ we consider binary operations "$\oplus$" and "$\cdot$", given by the tables:

$\oplus$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

Algebraic structures over the set $\{0,1\}^n$

Theorem 37 The structure $(\{0,1\},\oplus,\cdot)$ is a field.
The $(\{0,1\},\oplus,\cdot)$ is denoted by $GF(2)$ (Galois-field)
Define a binary operation "$\oplus$" on $\{0,1\}^n$, for $n\in\mathbb{N}$. If $x,y\in\{0,1\}^n$ $(x=(x_1,\ldots,x_n),\ y=(y_1,\ldots,y_n))$, we define

$$x\oplus y := (x_1\oplus y_1,\ldots,x_n\oplus y_n),$$

where on the right side $\oplus$ denotes the first operation in $GF(2)$.
Theorem 38 $(\{0,1\}^n,\oplus)$ is an Abelian group.
Define now the mapping $\{0,1\}\times\{0,1\}^n\to\{0,1\}^n$, denoted by "$\cdot$":
if $a\in\{0,1\}$ and $x=(x_1,\ldots,x_n)\in\{0,1\}^n$, then

$$a\cdot x := (a\cdot x_1,\ldots,a\cdot x_n),$$

where on the right "$\cdot$" denotes the second operation in $GF(2)$.

Algebraic structures over the set $\{0, 1\}^n$

By this definition, we have:
for $a \in \{0, 1\}$ and $x \in \{0, 1\}^n$,

$$a \cdot x = \begin{cases} (0, \ldots, 0), & \text{for } \ a = 0 \\ x, & \text{for } \ a = 1\,. \end{cases}$$

Theorem 39 Abelian group $(\{0, 1\}^n, \oplus)$ is a vector space over the field $GF(2)$ (denoted here by $\mathcal{S}_2^n$).
Clearly, $\mathcal{S}_2^n$) is an $n-$dimensional vector space, its base being e.g., $\{(1, , 0, \ldots, 0), (0, 1, 0, \ldots, 0), \ldots, (0, 0, \ldots, 0, 1)\}$.
The **weight** of a vector $x \in \{0, 1\}^n$, $x = (x_1, \ldots, x_n)$, denoted by $\|x\|$, is defined by:

$$\|x\| := \sum_{i=1}^{n} x_i$$

where on the right we have the sum of the numbers $0, 1$ as integers.

Algebraic structures over the set $\{0,1\}^n$

The **Hamming distance**, $d(x,y)$ of vectors x and y from $\{0,1\}^n$ is defined by

$$d(x,y) := \|x \oplus y\| .$$

Obviously, if $x = (x_1, \ldots, x_n),\ y = (y_1, \ldots, y_n)$ then

$$d(x,y) = \sum_{i=1}^{n} x_i \oplus y_i .$$

Observe the following: *The weight of a vector is the number of its non-zero coordinates*, and *the Hamming distance between x and y is equal to the number of coordinates on which these vectors differ*.

Algebraic structures over the set $\{0,1\}^n$

The following is straightforward.

If x i y are from $\{0,1\}^n$, then

1) $\|x\| = 0$ if and only if $x = (0,\ldots,0)$;

2) $\|x \oplus y\| \leq \|x\| + \|y\|$.

As a consequence, we have: for x, y, z from $\{0,1\}^n$,

a) $d(x,y) = 0$ if and only if $x = y$;

b) $d(x,y) = d(y,x)$;

c) $d(x,y) \leq d(x,z) + d(z,y)$.

Algebraic structures over the set $\{0,1\}^n$

Theorem 40 The ordered pair $(\{0,1\}^n, d)$ is a metric space, where

$$d : \{0,1\}^n \times \{0,1\}^n \to \mathbb{N}_0$$

is a mapping defined by the Hamming distance.
Metric space $(\{0,1\}^n, d)$ has a simple geometric interpretation:
Elements of the set $\{0,1\}^n$ are considered to be vertices of n-dimensional unit square in $\mathbb{R}^n$. Then $d(x,y)$ is the minimal number of edges connecting vertices x and y.

Examples

(a) The vector space $\int_2^4$ consists of vectors $(0,0,0,0)$, $(0,0,0,1), \dots,$ $(1,1,1,1)$, the cardinality of the space being 16. Here we have, e.g.,

$$\begin{array}{rcrcl}
(0,1,0,1) & \oplus & (1,1,1,0) & = & (1,0,1,1), \\
(1,0,0,1) & \oplus & (1,0,0,1) & = & (0,0,0,0), \\
0 & \cdot & (1,0,1,1) & = & (0,0,0,0), \\
1 & \cdot & (0,1,1,0) & = & (0,1,1,0), \quad \text{etc.}
\end{array}$$

(b) Hamming distance: if $x = 11101011$, $y = 10101010$, then $d(x,y) = 2$, since $x \oplus y = 01000001$, and $\|x \oplus y\| = 1 + 1 = 2$.

Decoding; errors

For a code $V \subseteq \{0,1\}^n$ we define the **code distance**, denoted by $d(V)$, with

$$d(V) := \min_{u \neq v,\, u,v \in V} d(u,v),$$

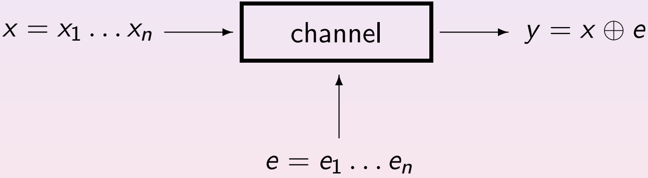
where $d(u,v)$ is the Hamming distance between u i v.
Suppose we are using a code $V \subseteq \{0,1\}^n$ to prepare messages which are sent through (BSC), whose matrix is

$$\Pi = \left[\begin{array}{cc} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{array} \right], \ \ 0 \leq \epsilon < \frac{1}{2}.$$

(we assume that the probability of an error, (ϵ), is less than $\dfrac{1}{2}$).

Decoding; errors

Passing through the channel, a word $x \in \{0,1\}^n$ is transformed into $y \in \{0,1\}^n$, where $y = x \oplus e$. The vector $e = e_1 \dots e_n \in \{0,1\}^n$ is an **error vector**.



Decoding; errors

We have $e_i = 0$ with probability $1-\epsilon$ (in which case the i-th coordinate is not changed), and $e_i = 1$ with probability ϵ (there was an error on $i-$th coordinate: $0 \to 1$ or $1 \to 0$).
If y differs from x in s $(0 \leq s \leq n)$ coordinates, we say that y is obtained from x due to s errors (it happens precisely when $\|e\| = s$).
Observe that the Hamming distance of the word x and another word obtained due to s errors is precisely s.
Suppose we are given a code $V \subseteq \{0,1\}^n$ which is used to prepare and send messages through BSC. Each mapping f of the set $\{0,1\}^n$ onto V is a **decoding** of words from $\{0,1\}^n$.
The kernel of the function f induces a partition of the set $\{0,1\}^n$ into (disjoint) classes $f^{-1}(v)$, $v \in V$.

Decoding; errors

The following proposition enable a **general decoding procedure**.

Theorem 41 Let BSC be given by the matrix

$$\Pi = \begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix}, \quad 0 < \epsilon < \frac{1}{2},$$

and let $V \subseteq \{0,1\}^n$ be a code. Then for all $u, v \in V$ and $x \in \{0,1\}^n$, we have

$$p(x|u) > p(x|v) \text{ if and only } d(u,x) < d(v,x).$$

Examples

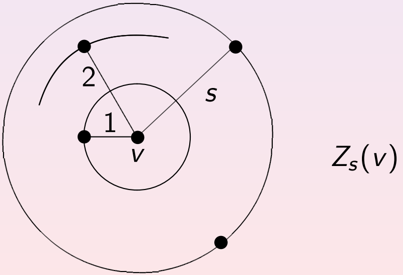
a) Code distance of the whole set $\{0,1\}^n$ equals 1 (e.g., $d(x,y) = 1$, where $x = (0,\dots,0)$, $y = (1,0,\dots,0)$).
b) For the code $V = \{0101, 1010, 1100, 0011, 1111\}$, we have $d(V) = 2$, since e.g., $d(x,y) = 2$, where $x = 1111$, and $y = 1100$. This is the minimal distance between two distinct code-words in V.
Errors: If $x = 101101$, then the word $y = 111101$ is obtained from x due to one error, and $z = 000011$ due to four errors.

Correcting and detecting errors

Let $V \subseteq \{0,1\}^n$ be a block-code. For $v \in V$ and $0 \leq s \leq n$, let

$$Z_s(v) = \{x \mid x \in \{0,1\}^n \quad \text{and} \quad d(v,x) \leq s\}.$$

$Z_s(v)$ is obviously the set of all words from $\{0,1\}^n$ which can be obtained from v due to at most s errors. In the metric space $(\{0,1\}^n, d)$ $Z_s(v)$ is a sphere whose center is v and radius s.



Correcting and detecting errors

Detecting errors is not so restrictive as correcting them. Namely, detecting errors means that we know that the received message differs from the sent one, but we do not know which digits are changed during transmission. The precise definition follows.

A code $V \subseteq \{0,1\}^n$ **detects s errors** $(0 \leq s \leq n)$ if for each $v \in V$ the following holds:

$$\text{If } \; x \in Z_s(v), \;\; \text{then } \; x \notin V \setminus \{v\}.$$

Theorem 42 a) A code $V \subseteq \{0,1\}^n$ corrects s errors if and only if

$$d(V) > 2s;$$

b) A code $V \subseteq \{0,1\}^n$ detects s errors if and only if

$$d(V) > s.$$

Correcting and detecting errors

Channels satisfying the following are frequently in use:

On the word whose length is n at most s $(0 \leq s \leq n)$ errors are possible.

It means that the set of error vectors is of the form:

$$\{e \mid e \in \{0,1\}^n \text{ and } \|e\| \leq s\}.$$

Example

The block-code $V = \{00000000, 00011111, 11111000, 11100111\}$ corrects at most two errors, since $d(V) = 5$. For example, the word 01011011 should be decoded as its nearest (in the sense of Hamming distance) code-word 00011111, from which it was created due to two errors.
The same code detects at most four errors. For example, the word 00111100 (in a channel with at most four errors on a code-word) could be decoded as any of the following two $11111000, 00011111$, since it differs from each of these in three digits.

Linear codes

A code $V \subseteq \{0,1\}^n$ is said to be **linear** (n,k)-**code** $(0 \leq k \leq n)$ if the set of its vectors form a k-dimensional subspace of the vector space $\mathcal{S}_2^n = (\{0,1\}^n, \oplus)$ over the field $GF(2)$.

Theorem 43 A code $V \subseteq \{0,1\}^n$ is linear if and only if the set of its vectors is closed under the operation $\oplus$ of vector addition.

Theorem 44 The code distance $d(V)$ of a linear code V is equal to the minimal weight of its non-zero vectors.

Linear codes

The number of elements of an (n, k)-code V is equal to the number of linear combinations of k arbitrary linearly independent vectors from V, with coefficients from the set $\{0, 1\}$. Therefore, we have the following.
Theorem 45 Linear (n, k)-code has 2^k elements.
A $k \times n$ matrix G, whose rows are vectors forming a basis of a linear (n, k)-code V, is said to be a **generating matrix** of V.
Scalar product of vectors $x = x_1 \cdots x_n,\ y = y_1 \cdots y_n,\ x, y \in \{0, 1\}^n$, is defined by

$$x \circ y := x_1 y_1 \oplus \cdots \oplus x_n y_n \,,$$

where "$\oplus$" and "$\cdot$" are operations in the field $GF(2)$.
Obviously, "$\circ$" is a mapping $(\{0, 1\}^n)^2 \to \{0, 1\}$, associating to a pair of vectors from $\mathcal{S}_2^n$ an element of the field $GF(2)$).

Linear codes

The following is fulfilled: for all $x, y, z \in \{0,1\}^n$
$x \circ y = y \circ x$ and
$x \circ (y \oplus z) = (x \circ y) \oplus (x \circ z)$.
Vectors x and y are said to be **orthogonal** if $x \circ y = 0$.
The set of vectors from $\{0,1\}^n$ orthogonal to all vectors in a linear (n,k)-code V, is called the **orthogonal complement** of a code V and usually is denoted by $\overline{V}$.
Theorem 46 If V is a linear (n,k)-code, then $\overline{V}$ is a linear $(n, n-k)$-code.
If $\overline{V}$ is the orthogonal complement of a code V, then clearly V is the orthogonal complement of $\overline{V}$.
A generating matrix F (of the type $(n-k) \times n$) of $\overline{V}$ is a **control matrix** of a code V.

Linear codes

The name of the control matrix is connected with its basic property:

$$F \cdot x = 0 \text{ if and only if } x \in V.$$

Therefore, using the matrix F it is possible to check whether a vector does or does not belong to V (more details in the following).

Examples

(a) The rows of the matrix

$$G = \begin{bmatrix} 0\,0\,1\,1\,1\,0 \\ 0\,1\,0\,0\,0\,1 \\ 1\,1\,0\,0\,1\,0 \end{bmatrix}$$

are linearly independent, hence this is a generating matrix of a linear $(6, 3)$-code. We determine its elements:

Examples

If $x_1 = 001110$, $x_2 = 010001$ and $x_3 = 110010$ (vector from the basis), then V consists of the following vectors:

$$\begin{array}{ccccccccc}
0 \cdot x_1 & \oplus & 0 \cdot x_2 & \oplus & 0 \cdot x_3 & = & 000000 & & \\
1 \cdot x_1 & \oplus & 0 \cdot x_2 & \oplus & 0 \cdot x_3 & = & 001110 & = & x_1 \\
0 \cdot x_1 & \oplus & 1 \cdot x_2 & \oplus & 0 \cdot x_3 & = & 010001 & = & x_2 \\
0 \cdot x_1 & \oplus & 0 \cdot x_2 & \oplus & 1 \cdot x_3 & = & 110010 & = & x_3 \\
0 \cdot x_1 & \oplus & 1 \cdot x_2 & \oplus & 1 \cdot x_3 & = & 100011 & & \\
1 \cdot x_1 & \oplus & 0 \cdot x_2 & \oplus & 1 \cdot x_3 & = & 111100 & & \\
1 \cdot x_1 & \oplus & 1 \cdot x_2 & \oplus & 0 \cdot x_3 & = & 011111 & & \\
1 \cdot x_1 & \oplus & 1 \cdot x_2 & \oplus & 1 \cdot x_3 & = & 101101 & &
\end{array}$$

Examples

Hence

$$V = \{000000, 001110, 010001, 110010, 100011, 111100, 011111, 101101\}.$$

Code distance $d(V)$ is 2, since, e.g., $\|x_2\| = 2$.

Examples

(b) If $x = 110101$, $y = 101000$ and $z = 101011$, then $x \circ y = 1$, $x \circ z = y \circ z = 0$, which means that the vectors x and z, as well as y and z are orthogonal. Each of these is also orthogonal to itself, since $x \circ x = y \circ y = z \circ z = 0$. This is a property of each vector having even weight.

Examples

(c) Let

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

be a control matrix of a linear (6,3) - code V. We are determining vectors of V using the equation

$$F \cdot x = 0, \qquad \text{i.e.,}$$

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} .$$

Examples

Thereby we have

$$\begin{array}{rcl} x_1 \oplus x_3 \oplus x_5 & = & 0 \\ x_2 \oplus x_3 \oplus x_6 & = & 0 \\ x_4 \oplus x_5 \oplus x_6 & = & 0 \end{array}$$

i.e.,

$$\begin{array}{rcl} x_1 & = & x_3 \oplus x_5 \\ x_2 & = & x_3 \oplus x_6 \\ x_4 & = & x_5 \oplus x_6 \end{array} .$$

Examples

Free variables are x_3, x_5 i x_6. The code V is given in the sequel.

	x_1	x_2	x_3	x_4	x_5	x_6
v_1	0	0	0	0	0	0
v_2	0	1	0	1	0	1
v_3	1	0	0	1	1	0
v_4	1	1	1	0	0	0
v_5	1	1	0	0	1	1
v_6	1	0	1	1	0	1
v_7	0	1	1	1	1	0
v_8	0	0	1	0	1	1

We associate arbitrary digits to free variables $(0, 0, 0$ to $1, 1, 1)$, and the values of the remaining coordinates are determined from the condition (1).

Linear codes: decoding

Let V be a linear (n, k) - code. For an arbitrary vector $x \in \{0,1\}^n$, suppose

$$x \oplus V := \{x \oplus v \mid v \in V\}$$

By the definition of a linear code V as a subspace of the vector space $\mathcal{S}_2^n$, it follows that $(V, \oplus)$ is an Abelian subgroup of the group $(\{0,1\}^n, \oplus)$. Therefore, $x \oplus V$ is a **coset** of (the subgroup) V in $\{0,1\}^n$.

Theorem 47 a) Every vector y from $\{0,1\}^n$ belongs to a coset of V;
b) x and y belong to V if and only if $x \oplus y \in V$;
c) $x \oplus V = y \oplus V$, for every $y \in x \oplus V$;
d) each coset of V has exactly 2^k elements.

Theorem 48 The family $\{x \oplus V \mid x \in \{0,1\}^n\}$ is a partition of the set $\{0,1\}^n$.

Linear codes: decoding

Suppose that a vector $y \in \{0,1\}^n$ appeared at the output of BSC. By the above comments, it belongs to the coset $y \oplus V$. If the word $v \in V$ was sent into the channel, then for an error vector e we have:

$$e = y \oplus v \in y \oplus V.$$

Hence, the coset $y \oplus V$ consists of all possible error vectors for the word y. *Therefore, the vector y is decoded by $v = y \oplus e$, where e is the vector with the minimal weight in the coset $y \oplus V$.*
If this vector e with a minimal weight is unique, then it is called the **leader** of the corresponding coset. If in the coset $y \oplus V$ there are several vectors with a minimal weight, then the decoding is not unique, each of this vectors plays the role of e.
Starting with a code V usually the table of cosets $x \oplus V,\ x \in \{0,1\}^n$ is constructed.

Linear codes: decoding

The following algorithm is used to determine the coset $y \oplus V$ in such a table.
If F is the control matrix of an (n, k)-code V, and $y \in \{0, 1\}^n$ (vector appearing at the channel output), then the vector

$$c = F \cdot y$$

is called the **corrector** of y. Obviously, $c \in \{0, 1\}^{n-k}$ and the corrector is a zero vector if and only if $y \in V$. Else, if $y = v \oplus e, \; v \in V$, we have

$$c = F \cdot y = F(v \oplus e) = F \cdot v \oplus F \cdot e = F \cdot e\,.$$

Linear codes: decoding

By the above, the following holds:
Two vectors belong to the same coset of V if and only if they have the same corrector c.
Therefore, each $c \in \{0,1\}^{n-k}$ determines precisely one coset of V and vice versa. Therefore, correctors are placed in headings of the class tables.
The algorithm follows. If $y \in \{0,1\}^n$ (the vector at the channel output), then:

I *the corrector c is determined from the equation $F \cdot y = c$;*

II *in the coset determined by c (i.e., in $y \oplus V$) the vector e with the minimal weight (the leader) is identified;*

III *y is decoded by $v = y \oplus e$.*

Linear codes: decoding

By the above, the following holds:
Two vectors belong to the same coset of V if and only if they have the same corrector c.
Therefore, each $c \in \{0,1\}^{n-k}$ determines precisely one coset of V and vice versa. Therefore, correctors are placed in headings of the class tables.
The algorithm follows. If $y \in \{0,1\}^n$ (the vector at the channel output), then:

I *the corrector c is determined from the equation $F \cdot y = c$;*

II *in the coset determined by c (i.e., in $y \oplus V$) the vector e with the minimal weight (the leader) is identified;*

III *y is decoded by $v = y \oplus e$.*

Linear codes: decoding

By the above, the following holds:
Two vectors belong to the same coset of V if and only if they have the same corrector c.
Therefore, each $c \in \{0,1\}^{n-k}$ determines precisely one coset of V and vice versa. Therefore, correctors are placed in headings of the class tables.
The algorithm follows. If $y \in \{0,1\}^n$ (the vector at the channel output), then:

I *the corrector c is determined from the equation $F \cdot y = c$;*

II *in the coset determined by c (i.e., in $y \oplus V$) the vector e with the minimal weight (the leader) is identified;*

III *y is decoded by $v = y \oplus e$.*

Examples

(*a*) Let $F = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$ be a control matrix of the $(4, 2)$ - code $V = \{0000, 1101, 1010, 0111\}$.

	leader				corrector
cosets of V	0000	1101	1010	0111	00
	1000	0101	0010	1111	01
	0100	1001	1110	0011	10
	0001	1100	1011	0110	11

Partition of the set $\{0, 1\}^4$ into cosets $x \oplus V, \ x \in \{0, 1\}^4$ is given in the table, correctors are indicated.

Examples

Code distance of V is 2, hence detecting one error over a code-word is possible, but not (precise) correction of each. Every class (cosset) in the table (except the second) contains the leader, i.e., a single vector with the minimal weight. Therefore, some single errors can be corrected. As an example, suppose $y = 1110$ appeared at the channel output. Then

$$c = F \cdot y = \begin{bmatrix} 0\,1\,0\,1 \\ 1\,0\,1\,1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

and the leader of this class is $e = 0100$.

Examples

The vector y is decoded by

$$y \oplus e = 1110 \oplus 0100 = 1010.$$

If the vector 1111 is received, then the corresponding corrector is $c = 01$, and in his class there are two vectors with the minimal weight: 1000 and 0010. Therefore, the vector 1111 can be decoded as 0111, but also as 1101.

Examples

(b) Let F (given in the sequel) be a control matrix of a linear code V:

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Suppose that we have to decode the message 001111100101101010. From the equation $F \cdot x = 0$, the code V can be determined:

$$V = \{000000, 101101, 111010, 010111\}$$

Examples

Partition of $\{0,1\}^6$ into cosets $x \oplus V$ is given in the table. These classes can be determined by the definition: if e.g., $x = 100000$, then $x \oplus 000000 = 100000$, $x \oplus 101101 = 001101$, $x \oplus 111010 = 011010$ i $x \oplus 010111 = 110111$. This is the ninth row of the table, corresponding to the corrector 1000. Another way to construct the table is to calculate the corrector c for each x, using the equation $c = F \cdot x$, and then by the same equation, one should determine other vectors from the coset.

Examples

The code distance of this code is 4, this code can correct one and detect three errors. If we are decoding words 001111, 100101 and 101010 (since the above message consists of these words), then we have correctors respectively, 0110, 0010 and 0100. In the corresponding cosets, vectors with minimal weight are: 011000 (100010), 001000, 010000. Therefore, the given message is decoded by 010111101101111010 or by 101101101101111010.

Examples

coset $x \oplus V$				corrector
000000	101101	111010	010111	0000
000100	101001	111110	010011	0001
001000	100101	110010	011111	0010
001100	100001	110110	011011	0011
010000	111101	101010	000111	0100
010100	111001	101110	000011	0101
011000	110101	100010	001111	0110
011100	110001	100110	001011	0111
100000	001101	011010	110111	1000
100100	001001	011110	110011	1001
101000	000101	010010	111111	1010
000001	101100	111011	010110	1011
001010	100111	110000	011101	1100
110100	011001	001110	100011	1101
000010	101111	111000	010101	1110
000110	101011	111100	010001	1111

Hamming codes: Detecting an error

For $n \in \mathbb{N}$, let

$$V_H(n) = \{x \in \{0,1\}^n \mid \|x\| \equiv 0 \pmod 2\}.$$

These are obviously vectors of the length n, with an even weight. $V_H(n)$ is a linear code. Indeed, if x and y are from $V_H(n)$, then $\|x\| \equiv 0 \pmod 2$, $\|y\| \equiv 0 \pmod 2$, and $\|x \oplus y\| \equiv 0 \pmod 2$, i.e., $x \oplus y \in V_H(n)$. Further on, $d(V_H(n)) = 2$, by the definition. This code detects an error (by checking the parity of words), and its cardinality is 2^{n-1} . Therefore a generating matrix is

$$G = \begin{bmatrix} 1\,0\,0 & \dots & 0\,1 \\ 0\,1\,0 & \dots & 0\,1 \\ \vdots & \vdots & \vdots \\ 0\,0\,0 & \dots & 1\,1 \end{bmatrix}$$

of the type $(n-1) \times n$, and a control matrix is

$$F = [1\,1 \cdots 1].$$

Example

The code $V_H(4)$ is given in the table. A single error on a code-word is detected as a word with odd weight.

0000	0110
0011	1010
0101	1100
1001	1111

Hamming codes: Correcting an error

Let $n = 2^s - 1,\ s = 2, 3, \ldots$ and let F be a control matrix of the type $s \times (2^s - 1)$, whose columns are binary coded numbers $1, 2, \ldots, 2^s - 1$, with s digits.

$$F = \left[\begin{array}{ccc} 0\,0\,0 & \cdots & 1\,1 \\ 0\,0\,0 & \cdots & 1\,1 \\ \multicolumn{3}{c}{\dots\dots\dots\dots\dots} \\ 0\,1\,1 & \cdots & 1\,1 \\ 1\,0\,1 & \cdots & 0\,1 \end{array} \right]$$

For $s = 2$ and $s = 3$, we have:

$$\left[\begin{array}{ccc} 0 & 1 & 1 \\ 1 & 0 & 1 \end{array} \right] \quad \text{and} \quad \left[\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array} \right]$$

Hamming codes: Correcting an error

The code $W_H(n)$, having F as a control matrix, has dimension $n - s = 2^s - 1 - s$. We show that $d(W_H(n)) \geq 3$. Let $x \in W_H(n),\ x = (x_1, \ldots, x_n)$ and let $x \neq 0$. Clearly, $F \cdot x = 0$, which is, if $F_1, \ldots, F_n$ are columns in F, equivalent with

$$x_1 F_1 \oplus\ x_2 F_2 \oplus \cdots \oplus\ x_n F_n = 0$$

(0 is the corresponding zero-vector).

Hamming codes: Correcting an error

Since $F_i \neq 0,\ i = 1, \ldots, n$, it follows that $\|x\| \neq 1$. In addition, $\|x\| \neq 2$, since in case $x_i = x_j = 1$ (for $i \neq j$), and all other coordinates being zeroes, we would have $F_i \oplus F_j = 0$, i.e., $F_i = F_j$, which is not true. Therefore, the smallest non-zero weight of a vector in $W_H(n)$ is greater than 2, hence the code corrects one error.

The algorithm for correcting errors follows from the definition of its control matrix F. Let x from $W_H(n)$ be changed on the i-th coordinate, so that y appeared at the output:

$$y = x \oplus e = (x_1, \ldots, x_n) \oplus (0, \ldots, 1, \ldots, 0) = (x_1, \ldots, x_i \oplus 1, \ldots, x_n).$$

Hamming codes: Correcting an error

Then

$$F \cdot y = F(x \oplus e) = F \cdot x \oplus F \cdot e = 0 \oplus F_i = F_i \, .$$

In other words, *the vector $F \cdot y$ is a binary denoted coordinate on which the error appeared*. If it is a zero vector, then obviously there was no error.

Example

The code $W_H(7)$ s given by the table

0000000	1110000
1101001	0011001
0101010	1011010
1000011	0110011
1001100	0111100
0100101	1010101
1100110	0010110
0001111	1111111

which could be easily obtained by solving the equation

$$F \cdot x = 0,$$

where

$$F = \left[\begin{array}{ccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array} \right].$$

Example

For example, if $y = 0110010$ is a received vector, then

$$F \cdot y = 0 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \oplus 1 \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \oplus 1 \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \oplus 0 \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \oplus 0 \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \oplus 1 \cdot \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \oplus 0 \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},$$

and since 111 is a binary coded number 7, the error is at the seventh digit. It follows that $x = 0110011$ and this is the vector by which y should be decoded.