Algebraic Foundations of Computer Science

Closures

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: `ferucio.tiplea@uaic.ro`

Spring 2021

Outline

Closures

Structural induction

Definitions by induction

Definitions by recursion

Course readings

Outline

Closures

Structural induction

Definitions by induction

Definitions by recursion

Course readings

Closures – example

Example 1

Let A be a set of atomic propositions. The set $PF(A)$ of propositional formulas over A is the least set which fulfills the following properties:

- $a \in PF(A)$, for any $a \in A$ (that is, $A \subseteq PF(A)$);
- if α and β are propositional formulas over A, then

$$\neg\alpha,\ (\alpha \vee \beta),\ (\alpha \wedge \beta),\ (\alpha \Rightarrow \beta),\ \text{and } (\alpha \Leftrightarrow \beta)$$

  are propositional formulas over A.

The three key features of $PF(A)$:

1. "includes A"
2. "closed under" $\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\Leftrightarrow$
3. "least set" with the above properties

Constructors and closures

An n-ary constructor over a set V is a relation r from V^n to V. That is, the elements of r are of the form $((a_1, \ldots, a_n), a)$.

Given an n-ary constructor r and a set A, denote by $r(A)$ the set:

$$r(A) = \{a | (\exists a_1, \ldots, a_n \in A)(((a_1, \ldots, a_n), a) \in r)\}$$

Definition 2

Let A be a set and $\mathcal{R}$ be a set of constructors. The closure of A under $\mathcal{R}$ is the least set $B \subseteq V$ with the properties:

- $A \subseteq B$;
- B is closed under $\mathcal{R}$, i.e., $r(B) \subseteq B$, for any $r \in \mathcal{R}$.

Existence of Closures

Theorem 3 (Existence of closures)

Given a set A and a set $\mathcal{R}$ of constructors, the closure of A under $\mathcal{R}$ exists and it is unique. Moreover, if $\mathcal{R}[A]$ denotes the closure of A under $\mathcal{R}$, then

$$\mathcal{R}[A] = \bigcup_{m \geq 0} B_m,$$

where

- $B_0 = A$;
- $B_{m+1} = B_m \cup \bigcup_{r \in \mathcal{R}} r(B_m)$, *for any* $m \geq 0$;

The closure of A under $\mathcal{R}$ is the union of a chain of sets:

$$B_0 = A, B_1 = B_0 \cup \mathcal{R}(B_0), B_2 = B_1 \cup \mathcal{R}(B_1), \ldots, \bigcup_{m \geq 0} B_m = \mathcal{R}[A],$$

where $\mathcal{R}(B_i) = \bigcup_{r \in \mathcal{R}} r(B_i)$.

The set of natural numbers as a closure

Definition 4

The successor of a set x, denoted $S(x)$, is the set $S(x) = x \cup \{x\}$.

Recall that the natural numbers are defined as follows:

- $0 = \emptyset$;
- $1 = S(0) = \{0\} = \{\emptyset\}$;
- $2 = S(1) = \{0, 1\} = \{\emptyset, \{\emptyset\}\}$ etc.

Therefore, $\mathbb{N}$ is the closure of $\{0\}$ under $\mathcal{R} = \{S\}$.

Closures of a binary relation

Definition 5

The reflexive closure of a binary relation $\rho \subseteq A \times A$ is the least reflexive binary relation $r(\rho)$ which includes ρ.

Claim: $r(\rho)$ can be computed as follows:

$$r(\rho) = \rho \cup \iota_A$$

Definition 6

The symmetric closure of a binary relation $\rho \subseteq A \times A$ is the least symmetric binary relation $s(\rho)$ which includes ρ.

Claim: $s(\rho)$ can be computed as follows:

$$s(\rho) = \rho \cup \rho^{-1}$$

Closures of a binary relation

Definition 7

The transitive closure of a binary relation $\rho \subseteq A \times A$ is the least transitive binary relation $t(\rho)$ which includes ρ.

Claim: $t(\rho)$, also denoted by ρ^+, can be computed as follows:

$$t(\rho) = \rho^+ = \bigcup_{m \geq 1} \rho^m,$$

where

- $\rho^1 = \rho$ and
- $\rho^{m+1} = \rho \circ \rho^m$, for all $m \geq 1$.

Closures of a binary relation

Definition 8

The reflexive and transitive closure of a binary relation $\rho \subseteq A \times A$ is the least reflexive and transitive binary relation ρ^* which includes ρ.

Claim: ρ^* can be computed as follows:

$$\rho^* = t(r(\rho)) = r(t(\rho)) = \bigcup_{m \geq 0} \rho^m,$$

where

- $\rho^0 = \iota_A$ and
- $\rho^{m+1} = \rho \circ \rho^m$, for all $m \geq 0$.

Closures of a binary relation

Definition 9

The closure under equivalence of a binary relation $\rho \subseteq A \times A$ is the least equivalence relation $equiv(\rho)$ which includes ρ.

Claim: $equiv(\rho)$ can be computed as follows:

$$equiv(\rho) = t(s(r(\rho))) = t(r(s(\rho))) = r(t(s(\rho))).$$

Remark 1

In general, $s(t(\rho)) \neq t(s(\rho))$.

Outline

Closures

Structural induction

Definitions by induction

Definitions by recursion

Course readings

Structural induction

Theorem 10 (Structural induction)

Let $B = \mathcal{R}[A]$ be the closure of A under $\mathcal{R}$ and let P be a property such that:

- *$P(a)$, for any $a \in A$;*
- *$(P(a_1) \wedge \cdots \wedge P(a_n) \;\Rightarrow\; P(a))$, for any $r \in \mathcal{R}$ and $a_1, \ldots, a_n, a \in B$ with $((a_1, \ldots, a_n), a) \in r$.*

Then, P is satisfied by any $a \in B$.

Remark 2

1. *Structural induction is equivalent to mathematical induction.*
2. *Structural induction is more appropriate for proving properties of closures than mathematical induction.*

Structural induction – example

Example 11

Let A be a set of atomic propositions. The set $PF(A)$ of propositional formulas as defined in Example 1 is the closure of A under some set of constructors (prove it!).

Let $P(\alpha)$ be the following property:

$$P(\alpha): \quad \alpha \text{ has as many left brackets as right brackets.}$$

By structural induction we can easily prove that P is satisfied by all propositional formulas over A (prove it!).

Outline

Closures

Structural induction

Definitions by induction

Definitions by recursion

Course readings

Definitions by induction

Definition 12

A set B is inductively defined by A and $\mathcal{R}$ if $B = \mathcal{R}[A]$.

If $B = \mathcal{R}[A]$, then B is obtained as follows:

- $B_0 = A$;
- $B_{m+1} = B_m \cup \mathcal{R}(B_m)$, for all $m \geq 0$;
- $B = \bigcup_{m \geq 0} B_m$.

If the chain

$$B_0, B_1, B_2, \ldots, B_m, B_{m+1} = B_m, B_{m+2} = B_m, \ldots$$

stabilizes to some set B_m, then its union is B_m and, therefore, $B = B_m$.

Definitions by induction

A definition by induction corresponds to the following while-loop (that might be non-terminating):

Algorithm 1: Computing closures

input : set A and set $\mathcal{R}$ of constructors;
output: $B = \mathcal{R}[A]$;

begin
  $B := A$;
  while $\mathcal{R}(B) \not\subseteq B$ **do**
    $B := B \cup \mathcal{R}(B)$

Outline

Closures

Structural induction

Definitions by induction

Definitions by recursion

Course readings

Definitions by recursion

Assume that B is inductively defined by A and $\mathcal{R}$. It would be a good idea to define functions f on B in a recursive way as follows:

- define f for any $a \in A$;
- if $((a_1, \ldots, a_n), a) \in r$ and the function has already been defined for $a_1, \ldots, a_n$, then define the function for a as a combinations of the values $f(a_1), \ldots, f(a_n)$ in the form

$$h(r)(f(a_1), \ldots, f(a_n)),$$

where h associates a (partial) function $h(r)$ to r.

Definitions by recursion

The definition above has a main drawback: it could not work for some sets B. Just think that the element a above might be defined in at least two different ways,

$$((a_1, \ldots, a_n), a) \in r$$

and

$$((a'_1, \ldots, a'_m), a) \in r'.$$

In such a case, you must be assured that

$$h(r)(f(a_1), \ldots, f(a_n)) = h(r')(f(a'_1), \ldots, f(a'_m)).$$

The easiest way to have this property fulfilled is to ask for each element $a \in B$ to have exactly one inductive construction of it from A and $\mathcal{R}$. If B has this property then it is called a free inductively defined set.

Definitions by recursion

However, for inductively defined sets we can prove the following result:

Lemma 13

Let $B = \mathcal{R}[A]$, C a set, $g : A \to C$, and h a function which associates a partial function $h(r) : C^n \to C$ to each $r \in \mathcal{R}$, where n is the arity of r. Then, there exists a unique relation $f \subseteq B \times C$ such that:

(1) $(a, g(a)) \in f$, *for any* $a \in A$;

(2) *if* $(a_1, b_1), \ldots, (a_n, b_n) \in f$, $((a_1, \ldots, a_n), a) \in r$ *and* $h(r)(b_1, \ldots, b_n)\downarrow$, *then* $(a, h(r)(b_1, \ldots, b_n)) \in f$;

(3) f *is the least relation from* B *to* C *which satisfies (1) and (2).*

Definitions by recursion

Definition 14

A set B is called free inductively defined by A and $\mathcal{R}$ if, for any $a \in B$,

- either $a \in A$,
- or there exists a unique $r \in \mathcal{R}$ and a unique n-tuple $(a_1, \ldots, a_n)$ such that $((a_1, \ldots, a_n), a) \in r$, where n is the arity of r (for $n = 0$ we understand that $a \in r$).

Now, we can obtain the following important result.

Theorem 15 (Recursion theorem)

Let B, C, g, and h as in Lemma 13. If B is free inductively defined by A and $\mathcal{R}$, then the binary relation f from Lemma 13 is a function.

Definitions by recursion

A slight extension of the recursion theorem is the following:

Theorem 16

Let $B = \mathcal{R}[A]$, C a set, $g : A \to C$, and h a function which associates a partial function $h(r) : B^n \times C^n \to C$ to each $r \in \mathcal{R}$, where n is the arity of r. If B is free inductively defined by A and $\mathcal{R}$, then there exists a unique function $f : B \to C$ such that:

(1) $f(a) = g(a)$, *for any* $a \in A$;

(2) $f(a) = h(r)(a_1, \ldots, a_n, f(a_1), \ldots, f(a_n))$, *for any* $a, a_1, \ldots, a_n$ *with* $((a_1, \ldots, a_n), a) \in r$ *and* $h(r)(a_1, \ldots, a_n, f(a_1), \ldots, f(a_{n_r}))\downarrow$, *where* n *is the arity of* r.

Definitions by recursion – example

Example 17

Let $PF(A)$ be the set of propositional formulas over A. It is easy to see that this set is free inductively defined.
Define a function $f : PF(A) \to \mathbb{N}$ in a recursive way as follows:

- $f(a) = 1$, for any $a \in A$;
- $f(\neg\alpha) = f(\alpha)$, for any $\alpha \in PF(A)$;
- $f((\alpha \vee \beta)) = f(\alpha) + f(\beta)$, for any $\alpha, \beta \in PF(A)$;
- $f((\alpha \wedge \beta)) = f(\alpha) + f(\beta)$, for any $\alpha, \beta \in PF(A)$;
- $f((\alpha \Rightarrow \beta)) = f(\alpha) + f(\beta)$, for any $\alpha, \beta \in PF(A)$;
- $f((\alpha \Leftrightarrow \beta)) = f(\alpha) + f(\beta)$, for any $\alpha, \beta \in PF(A)$.

The function f returns the length of propositional formulas.

Definitions by recursion – more examples

Pick up your favorite programming language and:

- show that its set of arithmetic and logic expressions is inductively defined;
- define recursively the length of an arithmetic expression;
- define inductively the set of variables of an arithmetic expression;
- define recursively the "height" of an arithmetic expression.

Definitions by recursion

An important particular case of the recursion theorem:

Theorem 18 (Recursion theorem for $\mathbb{N}$)

Let A be a set, $a \in A$, and $h : \mathbb{N} \times A \to A$ be a function. Then, there exists a unique function $f : \mathbb{N} \to A$ such that:

(1) $f(0) = a$;

(2) $f(n+1) = h(n, f(n))$, *for any* n.

This result can be strengthen to:

Theorem 19 (Parametric recursion theorem for $\mathbb{N}$)

Let A and P be sets, and $g : P \to A$ and $h : P \times \mathbb{N} \times A \to A$ functions. Then, there exists a unique function $f : P \times \mathbb{N} \to A$ such that:

(1) $f(p, 0) = g(p)$, *for any* $p \in P$;

(2) $f(p, n+1) = h(p, n, f(p, n))$, *for any* $p \in P$ *and* $n \in \mathbb{N}$.

Definitions by recursion

Addition, multiplication, and exponentiation on natural numbers are defined by recursion:

- Addition:
  - $x + 0 = x$
  - $x + (n + 1) = (x + n) + 1$;
- Multiplication:
  - $x \cdot 0 = 0$
  - $x \cdot (n + 1) = (x \cdot n) + x$;
- Exponentiation:
  - $x^0 = 1$
  - $x^{n+1} = (x^n) \cdot x$.

Definitions by recursion

In some cases the value of a function f at a natural number n may depend on the values of f at $0, \ldots, n-1$ (Fibonacci's sequence is such an example).
The recursion in such cases is called hereditary.

Theorem 20 (Hereditary recursion theorem)

Let A be a set, $S = \bigcup_{n \in \mathbb{N}} A^n$, and $h : \mathbb{N} \times S \to A$ be a function. Then, there exists a unique function $f : \mathbb{N} \to A$ such that

$$f(n) = h(n, f|_n),$$

for any $n \in \mathbb{N}$ (recall that $f|_0 = f|_\emptyset = \emptyset \in A^0$).

Exercise: Develop a parametric version of the hereditary recursion theorem.

Outline

Closures

Structural induction

Definitions by induction

Definitions by recursion

Course readings

Course readings

Course readings:

1. F.L. Țiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iași, 2006, pag. 70–79.

2. F.L. Țiplea: *Introducere în Teoria Mulțimilor*, Ed. Univesității "Al.I.Cuza", Iași, 1998, pag. 83–90.

Algebraic Foundations of Computer Science

Computational Introduction to Number Theory: Part I

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Divisibility. Prime numbers

The greatest common divisor

Congruences

Euler's totient function

Course readings

Outline

Divisibility. Prime numbers

The greatest common divisor

Congruences

Euler's totient function

Course readings

The division theorem

The absolute value of an integer a, denoted $|a|$, is defined by:

$$|a| = \begin{cases} a, & \text{if } a \geq 0 \\ -a, & \text{otherwise.} \end{cases}$$

Theorem 1 (The Division Theorem)

For any two integers a and b with $b \neq 0$, there are unique integers q and r such that $a = bq + r$ and $0 \leq r < |b|$.

In the equality $a = bq + r$ in the division theorem, a is called the dividend, b is called the divisor, q is called the quotient, and r is called the remainder. We usually write:

$$q = a \ div \ b \quad \text{and} \quad r = a \ mod \ b$$

Divisibility relation

Definition 2

The binary relation $| \subseteq \mathbb{Z} \times \mathbb{Z}$ given by

$$a|b \quad \Leftrightarrow \quad (\exists c \in \mathbb{Z})(b = ac),$$

for any $a, b \in \mathbb{Z}$, is called the divisibility relation on $\mathbb{Z}$.

If $a|b$ then we will say that a divides b, or a is a divisor/factor of b, or b is divisible by a, or b is a multiple of a.

Remark 1

If $a \neq 0$, then $a|b$ iff $b \bmod a = 0$.

If $a|b$ and $a \notin \{-1, 1, -b, b\}$, then a is called a proper divisor of b.

Basic properties of divisibility

Proposition 1

Let $a, b, c \in \mathbb{Z}$. Then:

1. *0 divides only 0;*
2. *a divides 0 and a;*
3. *1 divides a;*
4. *$a|b$ iff $a|-b$;*
5. *if $a|b$ and $b|c$, then $a|c$;*
6. *if $a|b+c$ and $a|b$, then $a|c$;*
7. *if $a|b$, then $ac|bc$. Conversely, if $c \neq 0$ and $ac|bc$, then $a|b$;*
8. *if $a|b$ and $a|c$, then $a|\beta b + \gamma c$, for any $\beta, \gamma \in \mathbb{Z}$;*
9. *if $a|b$ and $b \neq 0$, then $|a| \leq |b|$. Moreover, if a is a proper divisor of b, then $1 < |a| < |b|$.*

Prime numbers

Definition 3

A natural number $n \geq 2$ is called prime if the only positive factors of n are 1 and n. A natural number $n \geq 2$ that is not a prime is called composite.

Definition 4

Let $a_1, \ldots, a_m \in \mathbb{Z}$, where $m \geq 2$. We say that $a_1, \ldots, a_m$ are co-prime or relatively prime, denoted $(a_1, \ldots, a_m) = 1$, if the only common factors of these numbers are 1 and -1.

Example 5

- 2, 3, 5, 7, and 11 are prime numbers and 4, 6, and 9 are composite numbers.
- $(0, 1) = 1$ (0 and 1 are co-prime) and $(4, 6, 8) \neq 1$ (4, 6, and 8 are not co-prime).

Characterization of co-prime numbers

Theorem 6

Let $a_1, \ldots, a_m \in \mathbb{Z}$, where $m \geq 2$. Then, $(a_1, \ldots, a_m) = 1$ iff there are $\alpha_1, \ldots, \alpha_m \in \mathbb{Z}$ such that $\sum_{i=1}^{m} \alpha_i a_i = 1$.

Corollary 7

Let $a_1, \ldots, a_m, b \in \mathbb{Z}$, where $m \geq 2$. Then:

1. *if $(b, a_i) = 1$, for any i, then $(b, a_1 \cdots a_m) = 1$;*
2. *if $a_1, \ldots, a_m$ are pairwise co-prime and $a_i | b$, for any i, then $a_1 \cdots a_m | b$;*
3. *if $(b, a_1) = 1$ and $b | a_1 \cdots a_m$, then $b | a_2 \cdots a_m$;*
4. *if b is prime and $b | a_1 \cdots a_m$, then there exists i such that $b | a_i$.*

The fundamental theorem of arithmetic

Theorem 8 (The Fundamental Theorem of Arithmetic)

Every natural number $n \geq 2$ can be written uniquely in the form

$$n = p_1^{e_1} \cdots p_k^{e_k},$$

where $k \geq 1$, $p_1, \ldots, p_k$ are prime numbers written in order of increasing size, and $e_1, \ldots, e_k > 0$.

Example 9

- $4 = 2^2$, $9 = 3^2$, $12 = 2^2 \cdot 3$, $36 = 2^2 \cdot 3^2$.
- $105 = 3 \cdot 5 \cdot 7$.

The prime number theorem

Theorem 10

There are infinitely many primes.

Theorem 11 (The Prime Number Theorem)

Let $\pi(n) = |\{p | p \text{ is a prime and } p \leq n\}|$. *Then,*

$$\lim_{n\to\infty} \frac{\pi(n)}{\frac{n}{\ln n}} = 1.$$

We write

$$\pi(n) \sim \frac{n}{\ln n}$$

and say that $\pi(n)$ and $\frac{n}{\ln n}$ are asymptotically equivalent.

Values of $\pi(n)$

A few values of $\pi(n)$:

n	10^1	10^2	10^3	10^4	10^5	10^6	10^7
$\pi(n)$	4	25	168	1229	9592	78496	664579

How many 100-digit primes are there?

$$\begin{aligned}
\pi(10^{100}) - \pi(10^{99}) &\approx \frac{10^{100}}{100 \ln 10} - \frac{10^{99}}{99 \ln 10} \\
&= \frac{10^{99}}{ln\, 10}\left(\frac{1}{10} - \frac{1}{99}\right) \\
&> 0.39 \cdot 10^{98} \\
&\approx 4 \cdot 10^{97}
\end{aligned}$$

Large numbers

How large is 10^{97}? Below are a few interesting estimates and comparisons:

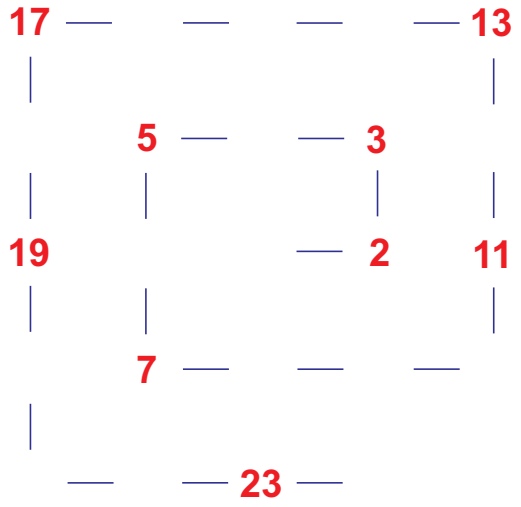
- the number of cells in the human body is estimated at 10^{14};
- the number of neuronal connections in the human brain is estimated at 10^{14};
- the universe is estimated to be $5 \cdot 10^{17}$ seconds old;
- the total number of particles in the universe has been variously estimated at numbers from 10^{72} up to 10^{87}.

Very large numbers often occur in fields such as mathematics, cosmology and cryptography. They are particularly important to cryptography where security of cryptosystems (ciphers) is usually based on solving problems which require, say, 2^{128} operations (which is about what would be required to break the 128-bit SSL commonly used in web browsers).

The prime spiral

There is no known formula for generating prime numbers in a row which is more efficient than the ancient sieve of Eratosthenes or the modern sieve of Atkin.

The Ulam spiral (or prime spiral), discovered by Stanislaw Ulam in 1963, is a simple method of graphing the prime numbers.



The prime numbers tend to line up along diagonal lines !

Outline

Divisibility. Prime numbers

The greatest common divisor

Congruences

Euler's totient function

Course readings

The greatest common divisor

Definition 12

Let $a_1, \ldots, a_m \in \mathbb{Z}$, not all zero, where $m \geq 2$. The greatest common divisor of these numbers, denoted $gcd(a_1, \ldots, a_m)$ or $(a_1, \ldots, a_m)$, is the largest integer d such that $d|a_i$, for all i.

Example 13

- $(2, 5, 7) = 1$.
- $(9, 3, 15) = 3$.

Proposition 2

Let $a_1, \ldots, a_m \in \mathbb{Z}$, not all zero, where $m \geq 2$. Then:

1. $(0, a_1, \ldots, a_m) = (a_1, \ldots, a_m)$;
2. $(0, a_1) = |a_1|$, *provided that* $a_1 \neq 0$;
3. $(a_1, a_2) = (a_2, a_1 \ mod \ a_2)$, *provided that* $a_2 \neq 0$.

Linear combination of the gcd

Theorem 14

Let $a_1, \ldots, a_m \in \mathbb{Z}$, not all zero, where $m \geq 2$. Then,

$$(a_1, \ldots, a_m) = \alpha_1 a_1 + \cdots + \alpha_m a_m$$

for some $\alpha_1, \ldots, \alpha_m \in \mathbb{Z}$.

Corollary 15

Let $a_1, \ldots, a_m \in \mathbb{Z}$, not all zero, where $m \geq 2$. Then, the equation

$$a_1 x_1 + \cdots + a_m x_m = b$$

has solutions in $\mathbb{Z}$ iff $(a_1, \ldots, a_m) | b$.

Example 16

$2x + 3y = 5$ has solutions in $\mathbb{Z}$ because $(2, 3) = 1$ divides 5, but $4x + 2y = 3$ does not have solutions in $\mathbb{Z}$ because $(4, 2) = 2$ does not divide 3.

The least common multiple

Definition 17

Let $a_1, \ldots, a_m \in \mathbb{Z}$, where $m \geq 2$. The least common multiple of these numbers, denoted $lcm(a_1, \ldots, a_m)$ or $[a_1, \ldots, a_m]$, is

- 0, if at least one of these numbers is 0;
- the smallest integer $b > 0$ such that $a_i | b$, for all i, otherwise.

Example 18

- $[0, a] = 0$, for any a.
- $[4, 6, 2] = 12$.

Theorem 19

Let $a, b \in \mathbb{N}$, not both zero. Then, $ab = (a, b)[a, b]$.

The Euclidean algorithm

The Euclidean Algorithm

If $a = 0$ or $b = 0$, but not both zero, then $(a, b) = max\{|a|, |b|\}$.

Let $a > b > 0$ and

$$\begin{array}{lcll} r_{-1} & = & r_0 q_1 + r_1, & 0 < r_1 < r_0 \\ r_0 & = & r_1 q_2 + r_2, & 0 < r_2 < r_1 \\ \cdots & & & \\ r_{n-2} & = & r_{n-1} q_n + r_n, & 0 < r_n < r_{n-1} \\ r_{n-1} & = & r_n q_{n+1} + r_{n+1}, & r_{n+1} = 0, \end{array}$$

where $r_{-1} = a$ și $r_0 = b$. Then,

$$(a, b) = (r_{-1}, r_0) = (r_0, r_1) = \cdots = (r_n, 0) = r_n$$

The Euclidean algorithm

Algorithm 1: Computing gcd

input : $a, b \in \mathbb{Z}$ *not both* 0;
output: $gcd(a, b)$;

begin
  while $b \neq 0$ **do**
    $r := a \ mod \ b$;
    $a := b$;
    $b := r$
  $gcd(a, b) := |a|$;

Theorem 20 (Lamé, 1844)

Let $a \geq b > 0$ be integers. The number of division steps performed by **Euclid**(a, b) *does not exceed 5 times the number of decimal digits in b.*

The extended Euclidean algorithm

The Euclidean algorithm can be easily adapted to compute a linear combination of the gcd as well. The resulting algorithm is called the Extended Euclidean Algorithm.

Given a and b there are α and β such that $(a, b) = \alpha a + \beta b$. The numbers α and β can be computed as follows:

$$
\begin{array}{rlcl}
 & & & \\
 & & & \\
1. & a & = & bq_1 + r_1 \\
2. & b & = & r_1 q_2 + r_2 \\
3. & r_1 & = & r_2 q_3 + r_3 \\
 & \dots & & \\
n. & r_{n-2} & = & r_{n-1} q_n + r_n \\
n+1. & r_{n-1} & = & r_n q_{n+1}
\end{array}
\qquad
\begin{array}{rcl}
V_a & = & (1, 0) \\
V_b & = & (0, 1) \\
V_{r_1} & = & V_a - q_1 V_b \\
V_{r_2} & = & V_b - q_2 V_{r_1} \\
V_{r_3} & = & V_{r_1} - q_3 V_{r_2} \\
 & & \\
V_{r_n} & = & V_{r_{n-2}} - q_n V_{r_{n-1}}
\end{array}
$$

$r_n = (a, b)$ and $V_{r_n} = (\alpha, \beta)$.

The extended Euclidean algorithm

Algorithm 2: Computing gcd and a linear combination of it

input : $a, b \in \mathbb{Z}$ *not both* 0;
output: $gcd(a, b)$ *and* $V = (\alpha, \beta)$ *s.t.* $gcd(a, b) = \alpha a + \beta b$;

begin
- $V_0 := (1, 0)$;
- $V_1 := (0, 1)$;
- **while** $b \neq 0$ **do**
  - $q := a \ div \ b$;
  - $r := a \ mod \ b$;
  - $a := b$;
  - $b := r$;
  - $V := V_0$;
  - $V_0 := V_1$;
  - $V_1 := V - qV_1$
- $gcd(a, b) := |a|$;
- $V := V_0$;

Linear Diophantine equations

The extended Euclidean algorithm can be used to compute integer solutions to linear Diophantine equations:

Algorithm 3: Computing solutions to linear Diophantine equations

input : $a, b, c \in \mathbb{Z}$ *such that not both* a *and* b *are* 0;
output: integer solution to $ax + by = c$, if it has;

begin
  compute $gcd(a, b) := \alpha a + \beta b$;
  if $gcd(a, b)|c$ **then**
    $c' := c/gcd(a, b)$;
    $x := \alpha c'$;
    $y := \beta c'$
  else
    "no integer solutions"

Outline

Divisibility. Prime numbers

The greatest common divisor

Congruences

Euler's totient function

Course readings

Congruences

Definition 21

Let $a, b, m \in \mathbb{Z}$. We say that a is congruent to b modulo m, denoted $a \equiv_m b$ or $a \equiv b \ mod \ m$, if $m|(a-b)$.

Example 22

- $6 \equiv 0 \ mod \ 2$.
- $-7 \equiv 1 \ mod \ 2$.
- $3 \not\equiv 2 \ mod \ 2$.
- $-11 \equiv 1 \ mod \ -4$ and $-11 \equiv 1 \ mod \ 4$.

Remark 2

If $m \neq 0$, then $a \equiv b \ mod \ m$ iff $a \ mod \ m = b \ mod \ m$.

Basic properties of congruences

Proposition 3

Let $a, b, c, d, m, m' \in \mathbb{Z}$ and $f : \mathbb{Z} \to \mathbb{Z}$ be a polynomial function with integer coefficients. Then:

1. $\equiv_m$ *is an equivalence relation on* $\mathbb{Z}$;
2. *if* $a \equiv_m b$, *then* $(a, m) = (b, m)$;
3. *if* $a \equiv_m b$ *și* $c \equiv_m d$, *then* $a + c \equiv_m b + d$, $a - c \equiv_m b - d$, $ac \equiv_m bd$, *and* $f(a) \equiv_m f(b)$;
4. 4.1 *if* $ac \equiv_{mc} bc$ *and* $c \neq 0$, *then* $a \equiv_m b$;
   4.2 *if* $ac \equiv_m bc$ *and* $d = (m, c)$, *then* $a \equiv_{m/d} b$;
   4.3 *if* $ac \equiv_m bc$ *and* $(m, c) = 1$, *then* $a \equiv_m b$;
5. 5.1 *if* $a \equiv_{mm'} b$, *then* $a \equiv_m b$ *and* $a \equiv_{m'} b$;
   5.2 *if* $a \equiv_m b$ *and* $a \equiv_{m'} b$, *then* $a \equiv_{[m,m']} b$;
   5.3 *if* $a \equiv_m b$, $a \equiv_{m'} b$, *and* $(m, m') = 1$, *then* $a \equiv_{mm'} b$.

$\mathbb{Z}_m$

Let $\mathbb{Z}_m$ be the set of all equivalence classes induced by $\equiv_m$. Then:

- $[a]_m = [a]_{-m}$, for any $a \in \mathbb{Z}$. Therefore, we may consider only $m \geq 0$;
- for any $a, b \in \mathbb{Z}$, if $a \neq b$ then $[a]_0 \neq [b]_0$. Therefore, $\mathbb{Z}_0$ has as many elements as $\mathbb{Z}$;
- for $m \geq 1$, $\mathbb{Z}_m = \{[0]_m, \ldots, [m-1]_m\}$ has exactly m elements.

Example 23

- $\mathbb{Z}_1 = \{[0]_1\}$, $\mathbb{Z}_2 = \{[0]_2, [1]_2\}$, $\mathbb{Z}_3 = \{[0]_3, [1]_3, [2]_3\}$.

Remark 3

We usually write $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$ instead of $\mathbb{Z}_m = \{[0]_m, \ldots, [m-1]_m\}$, for any $m \geq 1$.

Addition and multiplication modulo m

Define the following operations on $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$:

- $a + b = (a + b) \ mod \ m$; (binary operation)
- $a \cdot b = (a \cdot b) \ mod \ m$; (binary operation)
- $-a = (m - a) \ mod \ m$, (unary operation)

for any $a, b \in \mathbb{Z}_m$.

These operations fulfill the following properties:

- $+$ and $\cdot$ are associative and commutative;
- $a + 0 = 0 + a = a$, for any a;
- $a \cdot 1 = 1 \cdot a = a$, for any a;
- $a + (-a) = 0$, for any a.

$a + (-b)$ is usually denoted by $a - b$.

Inverses modulo m

- additive inverse modulo m.

  We have seen that $a + (-a) = 0$, for any a. $-a$ is called the additive inverse of a modulo m (it is unique);

- multiplicative inverse modulo m.
  - Given $a \in \mathbb{Z}_m - \{0\}$, is there any $b \in \mathbb{Z}_m$ such that $a \cdot b = 1$? That is, does any $a \in \mathbb{Z}_m$ have a multiplicative inverse modulo m?
  - Let us consider $m = 6$. There is no $b \in \mathbb{Z}_6$ such that $2 \cdot b = 1$.
  - Moreover, $\mathbb{Z}_6$ exhibits the following interesting property:

$$2 \cdot 3 = 0$$

(the product of two non-zero numbers is zero !!!).

Inverses modulo m and the group of units

Proposition 4

$a \in \mathbb{Z}_m$ has a multiplicative inverse modulo m iff $(a, m) = 1$.

The multiplicative inverse of a, when it exists, is unique and it is denoted by a^{-1}.

$\mathbb{Z}_m^* = \{a \in \mathbb{Z}_m | (a, m) = 1\}$ is called the group of units of $\mathbb{Z}_m$ or the group of units modulo m.

Example 24

- $\mathbb{Z}_1^* = \{0\}$.
- $\mathbb{Z}_{26}^*$ has 12 elements:
  - $1^{-1} = 1$, $3^{-1} = 9$, $5^{-1} = 21$,
  - $7^{-1} = 15$, $11^{-1} = 19$, $17^{-1} = 23$,
  - $25^{-1} = 25$.

Computing multiplicative inverses

The extended Euclidean algorithm can be easily used to compute multiplicative inverses modulo m:

Algorithm 4: Computing multiplicative inverses

input : $m \geq 1$ *and* $a \in \mathbb{Z}_m$;
output: a^{-1} *modulo* m, *if* $(a, m) = 1$;

begin
  compute $gcd(a, m) := \alpha a + \beta m$;
  if $gcd(a, m) = 1$ **then**
    $a^{-1} := \alpha \ mod \ m$
  else
    "a^{-1} *does not exist*"

Outline

Divisibility. Prime numbers

The greatest common divisor

Congruences

Euler's totient function

Course readings

Euler's totient function

Euler's totient function ϕ is given by:

$$\phi(m) = |\mathbb{Z}_m^*|,$$

for any $m \geq 1$. That is, $\phi(m)$ is the number of positive integers less than or equal to m and co-prime to m.

Theorem 25

1. $\phi(1) = 1$;
2. $\phi(p) = p - 1$, *for any prime* p;
3. $\phi(ab) = \phi(a)\phi(b)$, *for any co-prime integers* $a, b \geq 1$;
4. $\phi(p^e) = p^e - p^{e-1}$, *for any prime* p *and* $e > 0$;
5. $\phi(n) = (p_1^{e_1} - p_1^{e_1-1}) \cdots (p_k^{e_k} - p_k^{e_k-1})$, *for any* $n \geq 1$, *where* $n = p_1^{e_1} \cdots p_k^{e_k}$ *is the prime decomposition of* n.

Euler's totient function: examples

Example 26

1. $\phi(5) = 4$.
2. $\phi(26) = \phi(2 \cdot 13) = 12$.
3. $\phi(245) = \phi(5 \cdot 7^2) = 168$.

Remark 4

- *it is easy to compute $\phi(n)$ if the prime decomposition of n is known;*
- *it is hard to compute the prime decomposition of large numbers (512-bit numbers (about 155 decimals) or larger);*
- *it is hard to compute $\phi(n)$ if n is large and the prime decomposition of n is not known.*

Euler's theorem

Theorem 27 (Euler's Theorem)

Let $m \geq 1$. Then, $a^{\phi(m)} \equiv 1 \mod m$, for any integer a with $(a, m) = 1$.

Corollary 28 (Fermat's Theorem)

Let p be a prime. Then:

1. *$a^{p-1} \equiv 1 \mod p$, for any integer a with $p \nmid a$;*
2. *$a^p \equiv a \mod p$, for any integer a.*

Example 29

$1359^4 \equiv 1 \mod 5$ and $3^{168} \equiv 1 \mod 245$.

Outline

Divisibility. Prime numbers

The greatest common divisor

Congruences

Euler's totient function

Course readings

Course readings

1. F.L. Ţiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iași, 2006, pag. 143–164.

Algebraic Foundations of Computer Science

Computational Introduction to Number Theory: Part II

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Linear congruential equations

The Chinese remainder theorem

Quadratic residues

The Legendre symbol

The Jacobi symbol

Course readings

Outline

Linear congruential equations

The Chinese remainder theorem

Quadratic residues

The Legendre symbol

The Jacobi symbol

Course readings

Linear congruential equations

Theorem 1

Let $a, b, m \in \mathbb{Z}$ with $m \geq 1$. Then, the equation

$$ax \equiv b \ mod \ m$$

is solvable in $\mathbb{Z}$ iff $(a, m)|b$. Moreover, if it is solvable, then it has exactly (a, m) solutions in $\mathbb{Z}_m$ which are of the form

$$\left(x_0 + i\frac{m}{(a, m)}\right) \ mod \ m,$$

where x_0 is an arbitrary integer solution and $0 \leq i < (a, m)$.

Example 2

The equation

$$5x \equiv 25 \ mod \ 10$$

has $(5, 10) = 5$ *solutions in* $\mathbb{Z}_{10}$*:* $1, \ 3, \ 5, \ 7, \ 9.$

Linear congruential equations

Algorithm 1: Solving linear congruential equations

input : $m \geq 1$ *and* $a, b \in \mathbb{Z}$;
output: *all solutions modulo* m *of* $ax \equiv b \bmod m$;

begin
  compute $gcd(a, m) := \alpha a + \beta m$;
  if $gcd(a, m) | b$ **then**
    $b' := b/gcd(a, m)$;
    $x_0 := \alpha b'$;
    for $i := 0$ **to** $gcd(a, m) - 1$ **do**
      print $((x_0 + im/gcd(a, m)) \bmod m)$
  else
    "*no integer solutions*"

Outline

Linear congruential equations

The Chinese remainder theorem

Quadratic residues

The Legendre symbol

The Jacobi symbol

Course readings

The Chinese remainder theorem

According to D.Wells, the following problem was posed by Sun Tsu Suan-Ching (4th century AD):

> *There are certain things whose number is unknown. Repeatedly divided by 3, the remainder is 2; by 5, the remainder is 3; and by 7, the remainder is 2. What will be the number?*

The mathematical form of this problem is:

$$\begin{cases} x \equiv 2 \ mod \ 3 \\ x \equiv 3 \ mod \ 5 \\ x \equiv 2 \ mod \ 7 \end{cases}$$

This system of equations has a least integer solution which is $x = 23$.

The Chinese remainder theorem

Theorem 3 (Chinese Remainder Theorem)

Let $k \geq 1$ and $m_1, \ldots, m_k$ be pairwise co-prime integers. Then, for any $b_1, \ldots, b_k \in \mathbb{Z}$, the following system (S) of equations has a unique solution modulo $m_1 \cdots m_k$

$$(S) \begin{cases} x \equiv b_1 \ mod \ m_1 \\ \cdots \\ x \equiv b_k \ mod \ m_k \end{cases}$$

The solution can be obtained as follows:

- compute $c_i = \prod_{j=1,\ j \neq i}^{k} m_j$;
- compute an integer solution x_i of the equation $c_i x \equiv b_i \ mod \ m_i$, for any i;
- $x = (c_1 x_1 + \cdots + c_k x_k) \ mod \ (m_1 \cdots m_k)$ is the unique solution modulo $m_1 \cdots m_k$ of the system.

The Chinese remainder theorem: example

Example 4

Let (S) *be the system*

$$(S)\begin{cases} x \equiv 2 \ mod\ 3 \\ x \equiv 3 \ mod\ 5 \\ x \equiv 2 \ mod\ 7 \end{cases}$$

Then:

- $c_1 = 35$, $c_2 = 21$, *and* $c_3 = 15$;
- $x_1 = 1$ *is a solution of* $35x \equiv 2 \ mod\ 3$;
- $x_2 = 3$ *is a solution of* $21x \equiv 3 \ mod\ 5$;
- $x_3 = 2$ *is a solution of* $15x \equiv 2 \ mod\ 7$;
- $x = (35 \cdot 1 + 21 \cdot 3 + 15 \cdot 2) \ mod\ 105 = 128 \ mod\ 105 = 23$ *is the unique solution modulo* 105 *of the system* (S).

The Chinese remainder theorem: application

There is an important application of CRT to the problem of solving equations of the form $f(x) \equiv 0 \ mod \ m$, where $f(x)$ is a polynomial with integer coefficients and variables x.

Theorem 5

Let $f(x)$ be a polynomial with integer coefficients, and $m_1, \ldots, m_k$ be pairwise co-prime integers. Then, $a \in \mathbb{Z}$ is a solution to the equation

$$f(x) \equiv 0 \ mod \ m_1 \cdots m_k \tag{1}$$

if and only if a is a solution to each of the equations

$$f(x) \equiv 0 \ mod \ m_i, \quad 1 \leq i \leq k. \tag{2}$$

Moreover, the number of solutions in $\mathbb{Z}_{m_1 \cdots m_k}$ of the equation (1) is the product of the numbers of solutions in $\mathbb{Z}_{m_i}$ of the equations (2).

The Chinese remainder theorem: application

Example 6

1. The equation
$$x^2 \equiv 1 \bmod p,$$
where $p > 2$ is a prime number, has exactly 2 solutions in $\mathbb{Z}_p$, namely $x = 1$ and $x = p - 1$.
2. The equation
$$x^2 \equiv 1 \bmod p_1 \cdots p_k,$$
where $p_1, \ldots, p_k$ are distinct odd primes ($k \geq 2$), has exactly 2^k solutions in $\mathbb{Z}_{p_1 \cdots p_k}$.

Outline

Linear congruential equations

The Chinese remainder theorem

Quadratic residues

The Legendre symbol

The Jacobi symbol

Course readings

Quadratic residues - motivation

Proposition 1 (Solving quadratic congruences)

Let $p > 2$ be a prime and $a, b, c \in \mathbb{Z}$ such that $(a, p) = 1$. Then, the quadratic congruence

$$ax^2 + bx + c \equiv 0 \ mod \ p$$

has

1. two roots in $\mathbb{Z}_p$, if $\Delta \equiv y^2 \ mod \ p$ for some $y \in \mathbb{Z}$ with $p \not| y$;
2. one root in $\mathbb{Z}_p$, if $\Delta \equiv 0 \ mod \ p$;
3. no roots, otherwise,

where $\Delta = b^2 - 4ac$.

How hard is to decide if a given $a \in \mathbb{Z}_p^*$ satisfies $a \equiv y^2 \ mod \ p$ for some $y \in \mathbb{Z}$?

Quadratic residues and non-residues

Definition 7

Let $p > 2$ be a prime and $a \in \mathbb{Z}$ non-divisible by p. a is called a quadratic residue modulo p if $a \equiv x^2 \bmod n$ for some integer x.

If a is neither divisible by p nor a quadratic residue modulo p then a is called a quadratic non-residue modulo p.

Remark 1

An integer a non-divisible by a prime $p > 2$ is a quadratic (non-)residue modulo p if and only if $a \bmod p$ is a quadratic (non-)residue modulo p.

Denote

- $QR_p = \{a \in \mathbb{Z}_p^* | a$ is a quadratic residue modulo $p\}$
- $QNR_p = \{a \in \mathbb{Z}_p^* | a$ is a quadratic non-residue modulo $p\}$

Quadratic residues. Basic properties

Proposition 2

Let $p > 2$ be a prime. Then, $|QR_p| = |QNR_p| = \frac{p-1}{2}$.

Proposition 3

Let $p > 2$ be a prime. Then:

1. $a, b \in QR_p \quad \Rightarrow \quad (ab \bmod p) \in QR_p$;
2. $a \in QR_p \ \wedge \ b \in QNR_p \quad \Rightarrow \quad (ab \bmod p) \in QNR_p$;
3. $a, b \in QNR_p \quad \Rightarrow \quad (ab \bmod p) \in QR_p$.

Theorem 8 (Euler's Criterion)

Let $p > 2$ be a prime and $a \in \mathbb{Z}_p^*$. Then,

1. $a \in QR_p$ if and only if $a^{\frac{p-1}{2}} \equiv 1 \ mod \ p$;
2. $a \in QNR_p$ if and only if $a^{\frac{p-1}{2}} \equiv -1 \ mod \ p$.

Outline

Linear congruential equations

The Chinese remainder theorem

Quadratic residues

The Legendre symbol

The Jacobi symbol

Course readings

The Legendre symbol

Introduced by Adrien-Marie Legendre in 1798 when trying to prove the law of quadratic reciprocity.

Definition 9

Let $p > 2$ be a prime. The Legendre symbol of $a \in \mathbb{Z}$, denoted $\left(\frac{a}{p}\right)$, is defined by

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{if } p|a \\ 1, & \text{if } p \nmid a \text{ and } a \text{ is a quadratic residue modulo } p \\ -1, & \text{if } p \nmid a \text{ and } a \text{ is a quadratic non-residue modulo } p \end{cases}$$

Remark that the Legendre symbol is only defined for primes $p > 2$. For $p = 2$, all even integers are divisible by p and all odd integers are quadratic residues modulo p.

The Legendre symbol

Proposition 4

Let $p > 2$ be a prime and $a, b \in \mathbb{Z}$. If $a \equiv b \ mod \ p$ then $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$.

Therefore, $\left(\frac{a}{p}\right) = \left(\frac{a \ mod \ p}{p}\right)$.

Proposition 5

Let $p > 2$ be a prime. Then, for any $a \in \mathbb{Z}$, $\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \ mod \ p$.

Proposition 6

Let $p > 2$ be a prime. Then, for any $a, b \in \mathbb{Z}$, $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$.

The Legendre symbol

According to the above properties, computing the Legendre symbol modulo p comes down to computing $\left(\frac{-1}{p}\right)$ and $\left(\frac{q}{p}\right)$, for any prime q with $2 \leq q < p$.

Proposition 7

Let $p > 2$ be a prime. Then,

$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = \begin{cases} 1, & \text{if } p \equiv 1 \ mod\ 4 \\ -1, & \text{if } p \equiv 3 \ mod\ 4 \end{cases}$$

The Legendre symbol

Theorem 10 (Gauss' Criterion)

Let $p > 2$ be a prime and $a \in \mathbb{Z}$ non-divisible by p. Then, $\left(\frac{a}{p}\right) = (-1)^r$, where

$$r = |\{i \in \{1, \ldots, (p-1)/2\} | ia \ mod \ p > p/2\}|.$$

Proposition 8

Let $p > 2$ be a prime. Then,

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = \begin{cases} 1, & \text{if } p \equiv \pm 1 \ mod \ 8 \\ -1, & \text{if } p \equiv \pm 3 \ mod \ 8 \end{cases}$$

The Legendre symbol

Theorem 11 (Quadratic reciprocity law)

Let $p, q > 2$ be distinct primes. Then,

$$\left(\frac{q}{p}\right)\left(\frac{p}{q}\right) = (-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}}.$$

Equivalently,

$$\left(\frac{q}{p}\right) = \begin{cases} -\left(\frac{p}{q}\right), & \text{if } p, q \equiv 3 \ mod\ 4 \\ \left(\frac{p}{q}\right), & \text{otherwise} \end{cases}$$

Example 12

$$\left(\frac{7}{59}\right) = -\left(\frac{59}{7}\right) = -\left(\frac{3}{7}\right) = \left(\frac{7}{3}\right) = \left(\frac{1}{3}\right) = 1$$

The Legendre symbol

Basic rules for computing the Legendre symbol (review):

1. if $a \equiv b \ mod \ p$ then $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$
2. $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$
3. $\left(\frac{1}{p}\right) = 1$
4. $\left(\frac{-1}{p}\right) = \begin{cases} 1, & \text{if } p \equiv 1 \ mod \ 4 \\ -1, & \text{if } p \equiv 3 \ mod \ 4 \end{cases}$
5. $\left(\frac{2}{p}\right) = \begin{cases} 1, & \text{if } p \equiv \pm 1 \ mod \ 8 \\ -1, & \text{if } p \equiv \pm 3 \ mod \ 8 \end{cases}$
6. $\left(\frac{q}{p}\right) = \begin{cases} \left(\frac{p}{q}\right), & \text{if } p \equiv 1 \ mod \ 4 \text{ or } q \equiv 1 \ mod \ 4 \\ -\left(\frac{p}{q}\right), & \text{if } p \equiv q \equiv 3 \ mod \ 4 \end{cases}$

for any distinct primes $p, q > 2$ and $a, b \in \mathbb{Z}$.

Outline

Linear congruential equations

The Chinese remainder theorem

Quadratic residues

The Legendre symbol

The Jacobi symbol

Course readings

The Jacobi symbol

Introduced by Carl Gustav Jacob Jacobi in 1837 as a generalization of the Legendre symbol.

Definition 13

Let $n > 0$ be an odd integer. The Jacobi symbol of $a \in \mathbb{Z}$, denoted $\left(\frac{a}{n}\right)$, is defined by

$$\left(\frac{a}{n}\right) = \begin{cases} 1, & \text{if } n=1 \\ \left(\frac{a}{p_1}\right)^{e_1} \cdots \left(\frac{a}{p_k}\right)^{e_k}, & \text{otherwise} \end{cases}$$

where $n = p_1^{e_1} \cdots p_k^{e_k}$ is the prime factorization of n.

Remark 2

1. The Jacobi symbol is defined only for odd integers $n > 0$.
2. $(a, n) = 1$ if and only if $\left(\frac{a}{n}\right) \neq 0$, for all $a \in \mathbb{Z}$ and $n > 0$ odd.

The Jacobi symbol

Theorem 14

The following properties hold:

1. if $a \equiv b \ mod \ n$ then $\left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)$
2. $\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$
3. $\left(\frac{1}{n}\right) = 1$
4. $\left(\frac{-1}{n}\right) = \begin{cases} 1, & \text{if } n \equiv 1 \ mod \ 4 \\ -1, & \text{if } n \equiv 3 \ mod \ 4 \end{cases}$
5. $\left(\frac{2}{n}\right) = \begin{cases} 1, & \text{if } n \equiv \pm 1 \ mod \ 8 \\ -1, & \text{if } n \equiv \pm 3 \ mod \ 8 \end{cases}$
6. $\left(\frac{m}{n}\right) = \begin{cases} \left(\frac{n}{m}\right), & \text{if } n \equiv 1 \ mod \ 4 \text{ or } m \equiv 1 \ mod \ 4 \\ -\left(\frac{n}{m}\right), & \text{if } n \equiv m \equiv 3 \ mod \ 4 \end{cases}$

for any distinct odd integers $n, m > 0$ and $a, b \in \mathbb{Z}$.

The Jacobi symbol

Algorithm 2: Computing the Jacobi symbol

input : integer a and odd integer $n > 0$;
output: $\left(\frac{a}{n}\right)$

```
begin
    b := a mod n; c := n; s := 1;
    while b ≥ 2 do
        while 4|b do b := b/4;
        if 2|b then
            if c mod 8 ∈ {3, 5} then s := −s;
            b := b/2;
        if b = 1 then quit;
        if b mod 4 = 3 = c mod 4 then
            s := −s;
            (b, c) := (c mod b, b);
    return s · b;
```

Outline

Linear congruential equations

The Chinese remainder theorem

Quadratic residues

The Legendre symbol

The Jacobi symbol

Course readings

Course readings

1. F.L. Ţiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iași, 2006, pag. 164–172.

2. M. B. Nathanson: *Elementary Methods in Number Theory*, Graduate Texts in Mathematics, Springer-Verlag, 2000, pag. 100–120.

Algebraic Foundations of Computer Science

Computational Introduction to Number Theory: Part III

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Asymptotic notation

Complexity of the basic arithmetic operations

Course readings

Outline

Asymptotic notation

Complexity of the basic arithmetic operations

Course readings

Asymptotic notation

Given $g : \mathbb{N} \to \mathbb{R}_+$ a function, define the following sets:

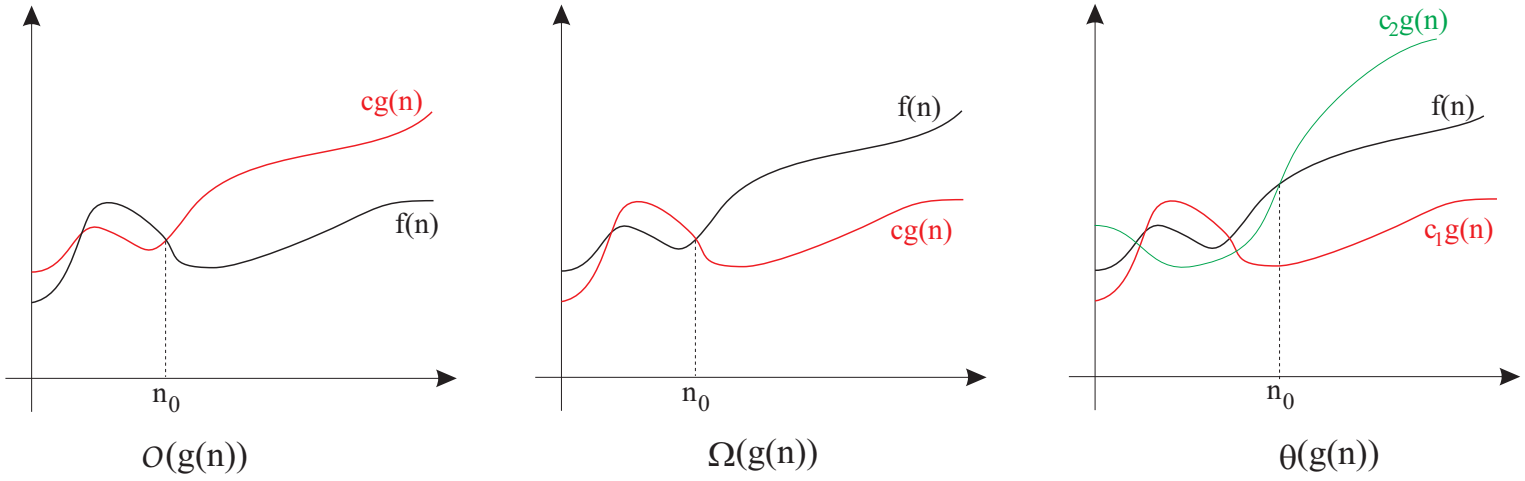
$$\begin{aligned}
\mathcal{O}(g) &= \{f : \mathbb{N} \to \mathbb{R}_+ | (\exists c \in \mathbb{R}_+^*)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)(f(n) \leq cg(n))\} \\
\Omega(g) &= \{f : \mathbb{N} \to \mathbb{R}_+ | (\exists c \in \mathbb{R}_+^*)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)(cg(n) \leq f(n))\} \\
\Theta(g) &= \{f : \mathbb{N} \to \mathbb{R}_+ | (\exists c_1, c_2 \in \mathbb{R}_+^*)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0) \\
&\qquad (c_1 g(n) \leq f(n) \leq c_2 g(n))\} \\
o(g) &= \{f : \mathbb{N} \to \mathbb{R}_+ | (\forall c \in \mathbb{R}_+^*)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)(f(n) \leq cg(n))\}
\end{aligned}$$

Definition 1

Let $f, g : \mathbb{N} \to \mathbb{R}_+$ and $X \in \{\mathcal{O}, \Omega, \Theta, o\}$. f is said to be X of g, denoted $f(n) = X(g(n))$, if $f \in X(g)$.

$\mathcal{O}$ ("big O"), Ω ("big Ω"), Θ ("big Θ"), and o ("little o") are order of magnitude symbols.

Asymptotic notation illustrated



- $f(n) = \mathcal{O}(g(n))$
  - $g(n)$ is an asymptotic upper bound for $f(n)$
  - $f(n)$ is no more than $g(n)$
  - used to state the complexity of a worst case analysis;
- $f(n) = \Omega(g(n))$ – similar interpretation;
- $f(n) = o(g(n))$ – $f(n)$ is less than $g(n)$ (the difference between $\mathcal{O}$ and o is analogous to the difference between $\leq$ and $<$).

Basic properties of asymptotic notation

Proposition 1

Let $f, g, h, k : \mathbb{N} \to \mathbb{R}_+$. Then:

1. $f(n) = \mathcal{O}(f(n))$;
2. if $f(n) = \mathcal{O}(g(n))$ and $g(n) = \mathcal{O}(h(n))$, then $f(n) = \mathcal{O}(h(n))$;
3. $f(n) = \mathcal{O}(g(n))$ iff $g(n) = \Omega(f(n))$;
4. $f(n) = \Theta(g(n))$ iff $f(n) = \mathcal{O}(g(n))$ and $f(n) = \Omega(g(n))$;
5. if $f(n) = \mathcal{O}(h(n))$ and $g(n) = \mathcal{O}(k(n))$, then $(f \cdot g)(n) = \mathcal{O}(h(n)k(n))$ and $(f + g)(n) = \mathcal{O}(max\{h(n), k(n)\})$;
6. if there exists $n_0 \in \mathbb{N}$ such that $g(n) \neq 0$ for any $n \geq n_0$, then $f(n) = o(g(n))$ iff $lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.

Useful inequalities

Some useful inequalities are in order:

- (Stirling's formula)

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n+1}} \leq n! \leq \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}},$$

  for any $n \geq 1$;

- for any real constants ϵ and c such that $0 < \epsilon < 1 < c$,

$$1 < ln\, ln\, n < ln\, n < e^{\sqrt{(ln\, n)(ln\, ln\, n)}} < n^{\epsilon} < n^{c} < n^{ln\, n} < c^{n} < n^{n} < c^{c^{n}}$$

  (each inequality holds for all $n \geq n_0$, where n_0 is suitable chosen).

Asymptotic notation: examples

Example 2

1. *If* $f(x) = a_0 + a_1 x + \cdots + a_k x^k$ *is a polynomial of degree* k *with real coefficients and* $f(x) \geq 0$ *for any* $x \in \mathbb{N}$*, then* $f(n) = \Theta(n^k)$.
2. $\log_c n = \Theta(\log n)$*, for any real constant* $c > 1$.
3. $\log n = \mathcal{O}(n^\epsilon)$*, for any real number* ϵ *such that* $0 < \epsilon < 1$.
4. $\log^k n = \mathcal{O}(n)$*, for any natural number* $k \geq 1$.
5. $n! = \Omega(2^n)$ *and* $n! = o(n^n)$.
6. $\log(n!) = \Theta(n \log n)$.
7. *If* $f : \mathbb{N} \to \mathbb{R}_+$ *satisfies* $f(n) \geq 1$ *for any* $n \geq n_0$ *and some* $n_0 \in \mathbb{N}$*, then*
$$f(n) = \Theta(2^{\lceil \log_2 f(n) \rceil}).$$
8. $4^n \neq \mathcal{O}(2^n)$.

Operations with classes of functions

Let $\mathcal{A}$ and $\mathcal{B}$ be sets of functions as those defined above (e.g., $\mathcal{O}(g)$ etc.), and let $f : \mathbb{N} \to \mathbb{R}_+$. Then, we write

1. $f + \mathcal{A} = \{f + g | g \in \mathcal{A}\}$;
2. $\mathcal{A} + \mathcal{B} = \{f + g | f \in \mathcal{A},\, g \in \mathcal{B}\}$;
3. $f\mathcal{A} = \{f \cdot g | g \in \mathcal{A}\}$. If f is the constant c function, then we will write $c\mathcal{A}$ instead of $f\mathcal{A}$;
4. $\mathcal{A}\mathcal{B} = \{fg | f \in \mathcal{A},\, g \in \mathcal{B}\}$;
5. $\mathcal{O}(\mathcal{A}) = \bigcup_{f \in \mathcal{A}} \mathcal{O}(f)$.

Basic properties

Convention: $\mathcal{A} = \mathcal{B}$ stands for $\mathcal{A} \subseteq \mathcal{B}$.

Proposition 2

Let $f, g : \mathbb{N} \to \mathbb{R}_+$ and $c \in \mathbb{R}_+$. Then:

1. $\mathcal{O}(f(n)) + \mathcal{O}(g(n)) = \mathcal{O}(f(n) + g(n))$;
2. $c\mathcal{O}(f(n)) = \mathcal{O}(f(n))$;
3. $\mathcal{O}(\mathcal{O}(f(n))) = \mathcal{O}(f(n))$;
4. $\mathcal{O}(f(n))\mathcal{O}(g(n)) = \mathcal{O}(f(n)g(n))$;
5. $\mathcal{O}(f(n)g(n)) = f(n)\mathcal{O}(g(n))$.

Outline

Asymptotic notation

Complexity of the basic arithmetic operations

Course readings

Programming with large integers

Programming with large integers (with more than 100 digits) is crucial for cryptography, computer security, computational algebra.

There have been developed several libraries for large integer arithmetic:

1. LIDIA (Darmstadt University of Technology)
   `http://www.cdc.informatik.tu-darmstadt.de/TI/LiDIA/`
2. PARI/GP (Université Bordeaux I, France)
   `http://pari.math.u-bordeaux.fr/`
3. NTL (New York University)
   `http://www.shoup.net/ntl/`
4. GMP (faster than any other multi-precision library)
   `http://gmplib.org/`

MpNT is a large integer library developed at Alexandru Ioan Cuza University of Iasi.

Base representation of integers

Let $\beta \geq 2$ be an integer called base. Each non-negative integer $a < \beta$ is called a digit of the base β or a base β digit.

Any non-negative integer a can be represented in a given base β as:

$$a = a_{k-1}\beta^{k-1} + \cdots + a_1\beta + a_0,$$

where $0 \leq a_i < \beta$ for all $0 \leq i < k$, and $a_{k-1} > 0$. The k-ary vector

$$(a_{k-1}, \ldots, a_0)_\beta$$

is called the representation of a in base β, the numbers a_i are called the digits of a in base β, and k is the length of the representation. k is also denoted by $|a|_\beta$ and it satisfies

$$|a|_\beta = \lfloor \log_\beta a \rfloor + 1$$

(convention: $\log_\beta 0 = 0$). Therefore, $|a|_\beta = \mathcal{O}(\log a)$.

Time complexity

Time complexity of algorithms will always be measured in digit operations, i.e., logical or arithmetic operations on digits. These operations are:

- digit comparison;
- digit addition, subtraction, and multiplication. Any of these operations takes two digits and a carry and produces a digit and a carry;
- division of a 2-digit number by a digit (the result consists of a quotient and a remainder).

Remark 1

The shifting and copying operations are not considered digit operations. In practice, these operations are fast in comparison with digit operations, so they can safely be ignored.

Time complexity: addition and subtraction

Assume that both operands have the same length k of representation in base β:

1. addition with carry: $\mathcal{O}(k)$
2. subtraction with borrow: $\mathcal{O}(k)$

Time complexity: multiplication

Assume that both operands have the same length k of representation in base β:

1. Schoolbook multiplication: $\mathcal{O}(k^2)$
   It is based on computing partial sums after each row multiplication. In this way, the intermediate numbers obtained by addition do not exceed $\beta^2 - 1$ and, therefore, a basic procedure for multiplication of two base β digits can be used;

2. Karatsuba multiplication: $\mathcal{O}(k^{\log 3})$
   It reduces the multiplication of two k-digit numbers to $\mathcal{O}(k^{\log 3})$ single-digit multiplications by recursively splitting the operands into two smaller parts.
   Generalization: Toom-Cook algorithm;

3. FFT based multiplication: $\mathcal{O}(k \log k)$
   Uses the Fast Fourier Transform and it is efficient for very large inputs (e.g., 1000-digit inputs).

Time complexity: division

Assume that the dividend length is $2k$ and the divisor length is k:

1. Schoolbook Division: $\mathcal{O}(k^2)$
   The method uses, as a basic step, the division of a $(k+1)$-digit integer by a k-digit integer. The main problem is to guess efficiently the quotient;

2. Recursive division: $\mathcal{O}(k^{\log 3} + k \log k)$
   It is based on a similar idea to that of Karatsuba's multiplication algorithm.

Time complexity: GCD

1. Euclidean algorithm: $\mathcal{O}((\log a)(\log b))$
2. Binary gcd algorithm: $\mathcal{O}((\log ab)^2)$.
3. Lehmer's or Sorenson's algorithm: $\mathcal{O}(k^2/\log k)$ where operands have length at most k.
4. Schönhage's algorithm (the fastest): $\mathcal{O}(k(\log k)^2 \log\log k)$ where operands have length at most k. This algorithm is based on FFT arithmetic and it is efficient for very large inputs.

Extended version of gcd algorithms: same complexity as for the corresponding gcd algorithm.

Time complexity: modular arithmetic

Arithmetic in $\mathbb{Z}_m$ is usually called modular arithmetic. The basic modular operations (addition, subtraction, multiplication, exponentiation) are obtained from the basic operations on integers plus a modular reduction.

Modular addition and subtraction can be easily implemented taking into account the following remarks:

1. $a + b < 2m$,
2. if $a \geq b$, then $0 \leq a - b < m$, and
3. if $a < b$ then $a + m - b < m$,

for all $a, b \in \mathbb{Z}_m$.

Therefore, the complexity of modular addition/subtraction is $\mathcal{O}(k)$, where k is the length of m.

Time complexity: modular arithmetic

Things are more involved in case of modular multiplication or exponentiation, where efficient modular reduction techniques are required.

Three main techniques for modular reduction are mostly used:

1. reduction by division
2. Barrett reduction
3. Montgomery reduction

	Barrett	Montgomery	Recursive Division
Restrictions on input	$a < \beta^{2k}$	$a < m\beta^k$	None
Pre-computation	$\lfloor \beta^{2k}/m \rfloor$	$-m_0^{-1} \ mod \ \beta$	Normalization
Post-computation	None	None	Unnormalization
Multiplication	$k(k+4)$	$k(k+1)$	$2K(k) + \mathcal{O}(k \log \ k)$

Figure: Complexity of multiplication under the three reduction methods

Time complexity: modular arithmetic

Modular exponentiation problem: Compute $a^e \ mod \ m$, where $m \geq 2$, $a \in \mathbb{Z}_m$, and $e > 0$.

1. The naive method: $\mathcal{O}(e \cdot (\log m)^2)$
   Compute $a^e \ mod \ m$ by performing $e - 1$ multiplications modulo m.

2. Exponentiation by squaring: $\mathcal{O}(\log e \cdot (\log m)^2)$
   Decompose e in base 2, $(e_{k-1}, e_{k-2}, \ldots, e_1, e_0)_2$, and

$$a^e \ mod \ m = (\cdots((a^{e_{k-1}})^2_m \otimes_m a^{e_{k-2}})^2_m \cdots a^{e_1})^2_m \otimes_m a^{e_0}$$

   The number of multiplications is

$$(k-1) + e_{k-2} + \cdots + e_0 < 2|e|_2$$

There is a large variety of exponentiation algorithms.

Time complexity: modular arithmetic

Algorithm 1: Exponentiation by Squaring

input : $m \geq 2$, $a \in \mathbb{Z}_m$, and $e > 0$;
output: $z = a^e \ mod \ m$;

```
begin
    y := a;
    z := 1;
    while e > 1 do
        f := e div 2;
        if e > 2f then
            z := z · y mod m;
        y := y · y mod m;
        e := f
    z := z · y mod m
```

Time complexity: Jacobi symbol

Odd integer $n > 1$ and integer $a > 0$:

1. The naive method: $\mathcal{O}((\log a)(\log n))$
2. Using the asymptotically fastest gcd algorithm: $\mathcal{O}(\log n \cdot (\log n)^2 \cdot \log\log n)$

For practical input sizes, the most efficient algorithms seem to be variants of the binary gcd (adapted to compute the Jacobi symbol)

Time complexity: factorization

Factoring a positive integer n means finding two positive integers $a, b > 1$ such that $n = ab$.

Factoring a composite integer is believed to be a hard problem (due to our failure so far to find a fast and practical factoring algorithm):

1. RSA-200, a 663 bit integer, was factored on May 9, 2005, by a team of the German Federal Agency for Information Security. The work began in late 2003. The sieving effort is estimated to have taken the equivalent of 55 years on a single 2.2 GHz Opteron CPU.

2. RSA-768, a 232 digit number, was factored on Dec 12, 2009, and it took almost two years. On a single core 2.2 GHz AMD Opteron processor with 2 GB RAM, sieving would have taken about 1500 years.

Time complexity: factorization

1. Factorization by trial division: requires $\pi(\sqrt{n})$ trial divisions (without counting primality testing)

2. Number Field Sieve: $\mathcal{O}(e^{(1.923+o(1))\sqrt[3]{(\ln m)(\ln \ln m)^2}})$

3. Quadratic Sieve: $\mathcal{O}(e^{(1+o(1))\sqrt{(\ln m)(\ln \ln m)}})$

Time complexity: primality

Attempts to find efficient algorithm for primality testing:

1. The simplest primality test: check whether there are integers $\leq \sqrt{n}$ which divides n. There are ways to speed up this test but it still remains very slow for large integers that have large prime factors;
2. Pratt, 1975: the primality problem is in $NP \cap co - NP$;
3. Miller, 1976: primality can be solved by deterministic polynomial time algorithms assuming *Extended Riemann Hypothesis* (ERH);
4. Solovay and Strassen, 1977: PPT algorithm $\mathcal{O}(\log^3 n)$;
5. Adleman, Pomerance, and Rumely: $(\log n)^{\mathcal{O}(\log\log\log n)}$ time;
6. Agraval, Kayal, and Saxena, 2002: $\tilde{\mathcal{O}}(\log^{7.5} n)$;
7. Lenstra and Pomerance, 2003: $\tilde{\mathcal{O}}(\log^6 n)$.

soft-$\mathcal{O}$ notation $f(n) = \tilde{\mathcal{O}}(g(n))$: $\exists k$ s.t. $f(n) = \mathcal{O}(g(n)\log^k g(n))$

Probabilistic primality testing – still the most efficient

Time complexity: probabilistic primality testing

Fermat primality test is based on the congruence

$$F(n,a): \quad a^{n-1} \equiv 1 \ mod \ n$$

for any prime n and integer a such that $(a,n)=1$.

Definition 3

Let $n > 2$ be an odd composite integer and $1 \leq a < n$.

1. *a is called a Fermat witness for n if $\neg F(n,a)$.*
2. *a is called a Fermat liar for n if $F(n,a)$.*

There are odd composite integers n such that all $a \in \mathbb{Z}_n^*$ are Fermat liars. These are called Carmichael numbers.

Time complexity: probabilistic primality testing

Algorithm 2: Fermat primality test

input : $n > 2$ odd;
output: "n is composite" or "n is prime";

begin
  choose randomly a, $1 < a < n$;
  if $(a, n) > 1$ **then**
    "n is composite"
  else
    if $a^{n-1} \not\equiv 1 \bmod n$ **then**
      "n is composite"
    else
      "n is prime"

Complexity: $\mathcal{O}((\log n)^3)$

Time complexity: probabilistic primality testing

Given a positive integer $n \geq 2$, denote

$$\mathcal{F}_n = \{a | 1 \leq a < n \ \wedge \ \ F(n, a)\}.$$

Theorem 4

Let $n \geq 2$ be an odd composite integer. Then, $\mathcal{F}_n \subseteq \mathbb{Z}_n^*$ and, if there exists $a \in \mathbb{Z}_n^*$ such that $\neg F(n, a)$, then $\mathcal{F}_n$ is a proper subgroup of $\mathbb{Z}_n^*$.

Fermat primality test gives a wrong answer with probability less than 1/2.

Time complexity: probabilistic primality testing

Solovay-Strassen primality test is based on the congruence

$$E(n,a): \quad \left(\frac{a}{n}\right) \cdot a^{\frac{n-1}{2}} \bmod n = 1$$

for any odd prime n and integer a such that $(a,n)=1$.

Definition 5

Let $n > 2$ be an odd composite integer and $1 \leq a < n$.

1. *a is called an Euler witness for n if $\neg E(n,a)$.*
2. *a is called an Euler liar for n if $E(n,a)$.*

There are no odd composite integers n without Euler witnesses (in other words, there are no Carmichael numbers in this case).

Time complexity: probabilistic primality testing

Algorithm 3: Solovay-Strassen primality test

input : $n > 2$ odd;
output: "n is composite" or "n is prime";

begin
  choose randomly a, $1 < a < n$;
  if $\left(\frac{a}{n}\right) = 0$ **then**
    "n is composite"
  else
    if $\left(\frac{a}{n}\right) \equiv a^{\frac{n-1}{2}} \bmod n$ **then**
      "n is prime"
    else
      "n is composite"

Complexity: $\mathcal{O}((\log n)^3)$

Time complexity: probabilistic primality testing

Given a positive integer $n \geq 2$, denote

$$\mathcal{E}_n = \{a | 1 \leq a < n \ \wedge \ E(n, a)\}.$$

Theorem 6

Let $n \geq 2$ be an odd composite integer. Then,

1. $\mathcal{E}_n \subseteq \mathcal{F}_n$;
2. $\mathcal{E}_n$ is a proper subgroup of $\mathbb{Z}_n^*$.

Solovay-Strassen primality test gives a wrong answer with probability less than $1/2$.

Time complexity: probabilistic primality testing

Miller-Rabin primality test is based on the following theorem:

Theorem 7

Let $n \geq 2$ be an odd integer, $n = 2^s t + 1$, where $s, t \geq 1$ and t is odd. Then, n is prime if and only if

$$M(n, a): \quad a^t \equiv 1 \ mod \ n \quad \vee \quad (\exists i < s)(a^{2^i t} \equiv -1 \ mod \ n),$$

for any integer a such that $(n, a) = 1$.

Definition 8

Let $n > 2$ be an odd composite integer and $1 \leq a < n$.

1. *a is called an Miller-Rabin witness for n if $\neg M(n, a)$.*
2. *a is called an Miller-Rabin liar for n if $M(n, a)$.*

Time complexity: probabilistic primality testing

Algorithm 4: Miller-Rabin primality test

input : $n > 2$ odd;
output: "n is composite" or "n is prime";

begin
  decompose $n := 2^s t + 1$, where t is odd;
  choose randomly a, $1 < a < n$;
  if $(a, n) > 1$ **then**
    "n is composite"
  else
    $r := a^t \bmod n$;
    if $r \in \{1, n-1\}$ **then** "n is prime";
    if $(\exists 1 \leq i \leq s-1)(r^{2^i} \equiv -1 \bmod n)$ **then**
      "n is prime"
    else
      "n is composite"

Complexity: $\mathcal{O}((\log n)^3)$

Time complexity: probabilistic primality testing

Given a positive integer $n \geq 2$, denote

$$\mathcal{M}_n = \{a | 1 \leq a < n \ \wedge \ \ M(n, a)\}.$$

Theorem 9

Let $n \geq 2$ be an odd composite integer. Then, $\mathcal{M}_n \subseteq \mathcal{F}_n$ and

$$|\mathcal{M}_n| \leq \frac{n-1}{4}.$$

Miller-Rabin primality test gives a wrong answer with probability at most $1/4$.

Outline

Asymptotic notation

Complexity of the basic arithmetic operations

Course readings

Course readings

1. F.L. Ţiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iaşi, 2006, pag. 172–178.

2. F.L. Ţiplea, S. Iftene, C. Hriţcu, I. Goriac, R.M. Gordân, E. Erbiceanu: *MpNT: A Multi-precision Number Theory Package. Number-theoretic Algorithms (I)*, Technical Report, "Al.I.Cuza" University of Iaşi, 2003.

Algebraic Foundations of Computer Science

Applications of Number Theory to Cryptography

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Introduction to cryptography

Factorization and Euler's function

Applications of the CRT

Applications of the Jacobi symbol

Course readings

Outline

Introduction to cryptography

Factorization and Euler's function

Applications of the CRT

Applications of the Jacobi symbol

Course readings

Introduction to cryptography

- Cryptography is the field concerned with techniques for securing information, particularly in communications;
- Cryptography focuses on the following main paradigms:
  - Authentication – the process of proving one's identity (the primary forms of host-to-host authentication on the Internet today are name-based or address-based, both of which are notoriously weak);
  - Privacy/confidentiality – ensuring that no one can read the message except the intended receiver;
  - Integrity – assuring the receiver that the received message has not been altered in any way from the original.

Applications of cryptography

- computer and information security: cryptography is necessary when communicating over any untrusted medium, which includes just about any network, particularly the Internet.
- e-commerce, e-payment, e-voting, e-auction, e-lottery, and e-gambling schemes, are all based on cryptographic (security) protocols.

Examples of software tools that havily rely on cryptographic techniques: IPsec, SSL & TLS, DNSsec, S/MIME, SET etc.

Cryptographic primitives

Crypto primitives:

1. Encryption schemes (ciphers)
2. Digital signature schemes
3. Hash Functions
4. Message authentication code (MAC) schemes
5. Secret sharing schemes
6. and more ...

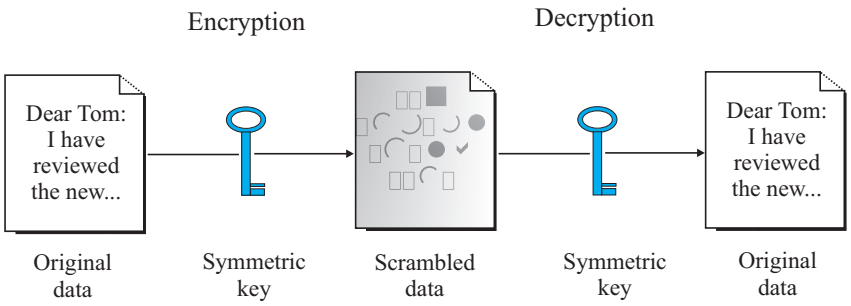
Basic elements of a cryptographic primitive:

1. A message space
2. A key space
3. An output space (of ciphertexts, message digests and so on)
4. Algorithms (for encryption, decryption, hashing, signing and so on)

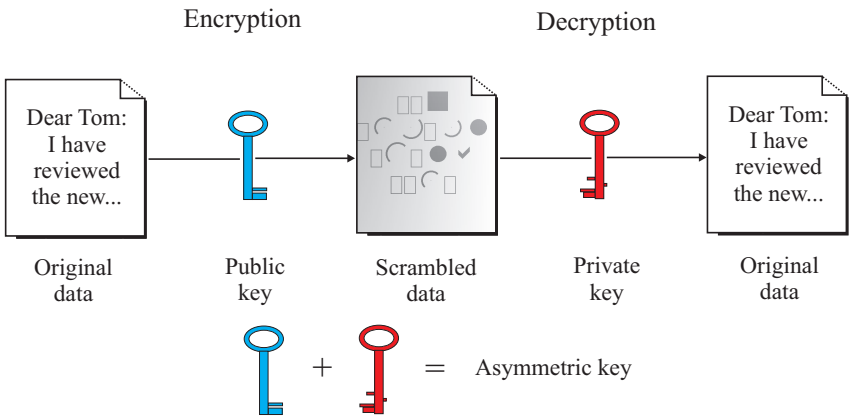
Two classes of cryptographic primitives

Illustration on ciphers:

- Symmetric (private-key, single-key) ciphers – encryptions and decryptions are performed by the same key



- Asymmetric (public-key) ciphers – encryptions are performed by a public-key, while decryptions are performed by a corresponding private key



Outline

Introduction to cryptography

Factorization and Euler's function

Applications of the CRT

Applications of the Jacobi symbol

Course readings

The RSA cipher

The RSA cipher, proposed in 1977 by Ronald Rivest, Adi Shamir, and Leonard Adleman is one of the most prominent public-key ciphers that is still secure and in use.

Key generation: public key: (n, e), where

$n = pq$, p and q distinct primes

$e \in \mathbb{Z}^*_{\phi(n)}$

private key: (p, q, d), where

$ed \equiv 1 \bmod \phi(n)$

Encryption of $x \in \mathbb{Z}_n$: output $c = x^e \bmod n \bmod n$

Decryption of $c \in \mathbb{Z}_n$: output $x = c^d \bmod n$

Correctness: $(x^e)^d \equiv x \bmod n$, for all $x \in \mathbb{Z}_n$ Prove it!

The RSA cipher

Example 1 (With artificially small parameters)

Let $p = 61$ and $q = 53$. Then:

- $n = pq = 3233$ and $\phi(n) = 3120$;
- If we chose $e = 17$, then d can be computed with the extended Euclidean algorithm. We obtain $d = e^{-1} \ mod \ 3120 = 2753$;
- $n = 3233$ and $e = 17$ are public parameters; p, q, and d private;

Let $x = 123$ be a plaintext. The cryptotext is

$$c = 123^{17} \ mod \ 3233 = 855.$$

In order to decrypt y we compute

$$855^{2753} \ mod \ 3233 = 123.$$

Security of the RSA cipher

Security issues:

- If p or q is recovered (e.g., by factoring n in reasonable time), then the system is completely broken;
- If $\phi(n)$ can be computed in reasonable time, then the system is completely broken;
- If d can be easily computed from n and e, then the system is completely broken.

In practice:

- p and q are 512-bit primes (or even larger);
- e is small (fast encryption) but chosen such that $d > \sqrt[4]{n}$ (otherwise, an efficient attack can be mounted).

For more details: http://www.rsasecurity.com/.

Digital signatures

Public key cryptography solves another problem crucial to e-commerce and Internet cyber relationship: it lets you emulate written signatures. This use of the public-key is called digital signature.

A digital signature must provide:

- authenticity and integrity. That is, it must be "impossible" for anyone who does not have access to the secret key to forge (x, s) (x is the original data and s is its associated signature);
- non-repudiation. That is, it must be impossible for the legitimate signer to repudiate his own signature.

Signing (encrypting with a private key) is extremely slow, so you usually add a time-saving (and space-saving) step before you encrypt messages. It is called message digesting or hashing.

Digital signatures from public-key ciphers

Public-key ciphers may be used to produce digital signatures:

- Assume that K_e is A's public key and K_d is A's private key and, moreover, $e_{K_e}(d_{K_d}(x)) = x$;
- Then, the decryption of a message x by K_d is the digital signature associated to x. It can be verified by K_e:

$$x \stackrel{?}{=} e_{K_e}(d_{K_d}(x)).$$

  Therefore, in such a case, K_d is used to sign messages (it will be secret) and K_e is used to verify signatures (it will be public).

The RSA signature is obtained from the RSA public-key cipher.

The RSA digital signature

Example 2 (With artificially small parameters)

Let $p = 61$ and $q = 53$. Then:

- $n = pq = 3233$ and $\phi(n) = 3120$;
- If we chose $e = 17$, then d can be computed with the extended Euclidean algorithm. We obtain $d = e^{-1} \ mod \ 3120 = 2753$;
- $n = 3233$ and $d = 2753$ are public parameters to verify signatures; p, q, and d private parameters to sign messages;

Let $x = 123$ be a message. The signature is

$$s = 123^{2753} \ mod \ 3233 = 2746.$$

So, $(123, 2746)$ is the signed message.

In order to verify the signature for $(123, 2746)$ we compute

$$2746^{17} \ mod \ 3233 = 123.$$

Outline

Introduction to cryptography

Factorization and Euler's function

Applications of the CRT

Applications of the Jacobi symbol

Course readings

Threshold sharing schemes

The Chinese Remainder Theorem (CRT) has numerous applications in various fields of mathematics, computer science, and engineering. We illustrate below an application of it to secret sharing.

Definition 3

A (k, n)-threshold sharing scheme consists of n parties $P_1, \ldots, P_n$ sharing a secret S in such a way that the following properties hold:

- $k \leq n$;
- *each P_i has an information I_i (secret share);*
- *knowledge of any k of $I_1, \ldots, I_k$ enables one to find S easily;*
- *knowledge of less than k of $I_1, \ldots, I_k$ does not enable one to find S easily.*

- Party P_i = people, software/hardware computing device ... Simply denote it by i and refer to it as user or participant;
- Secrets and secret shares are usually viewed as integers.

Goldreich-Ron-Sudan (GRS) scheme

(1) Parameter setup: integers t and n such that $0 < t+1 \leq n$ and sequence of pairwise coprime integers $m_0 < m_1 < \cdots < m_n$. All are public parameters;

(2) Secret and share spaces: the secret space is $\mathbb{Z}_{m_0}$ and the share space of the ith participant is $\mathbb{Z}_{m_i}$, for all $1 \leq i \leq n$;

(3) Secret sharing: given $s \in \mathbb{Z}_{m_0}$, share it by $s_i = s' \ mod \ m_i$, for all $1 \leq i \leq n$, where s' is the unique solution modulo $m_0 \prod_{i=1}^{t} m_i$ of the system

$$x \equiv r_i \ mod \ m_i, \quad 0 \leq i \leq t$$

where $r_0 = s$ and r_i is randomly chosen from $\mathbb{Z}_{m_i}$ for all $1 \leq i \leq t$;

(4) Secret reconstruction: any $t+1$ distinct shares $s_{i_1}, \ldots, s_{i_{t+1}}$ can uniquely reconstruct the secret s by computing first the unique solution modulo $\prod_{j=1}^{t+1} m_{i_j}$ of the system

$$x \equiv s_{i_j} \ mod \ m_{i_j}, \quad 1 \leq j \leq t+1$$

and then reducing it modulo m_0.

Goldreich-Ron-Sudan (GRS) scheme

Example 4 (With artificially small parameters)

(1) Parameter setup: $t = 2$, $n = 5$ and

$$m_0 = 31 < m_1 = 37 < m_2 = 41 < m_3 = 43 < m_4 = 47 < m_5 = 53$$

(3) Secret sharing: We choose $s = 13$ for sharing, generate $r_1 = 3$ and $r_2 = 5$, and solve the system

$$x \equiv_{31} 13,\ x \equiv_{37} 3,\ x \equiv_{41} 5$$

We get $s' = 37151$ and then the shares 3, 5, 42, 21, 51

(4) Secret reconstruction: Assume that the last 3 participants want to recover the secret s. They solve the system

$$x \equiv_{43} 42,\ x \equiv_{47} 21,\ x \equiv_{53} 51$$

and get 37151. By reducing this modulo 31 they obtain $s = 13$.

Security of the GRS scheme

CRT-based secret sharing schemes can only achieve asymptotic security properties:

- X random variable associated with the secret space $\mathbb{Z}_{m_0}$;
- For $I \subseteq \{1, \ldots, n\}$, Y_I random variable associated to the combined share space $\prod_{i \in I} \mathbb{Z}_{m_i}$;
- Loss of entropy: $\Delta(y_I) = H(X) - H(X|Y_I = y_I)$,, where $y_I \in \prod_{i \in I} \mathbb{Z}_{m_i}$ and H is the entropy;
- Asymptotic perfection: $\Delta(y_I) \to 0$ as $m_0 \to \infty$;
- Information rate of the participant i: $|\mathbb{Z}_{m_i}|/|\mathbb{Z}_{m_0}|$
- Asymptotic idealness: asymptotic perfection and the information rate of each participant goes to 1 as $m_0 \to \infty$.

Security of the GRS scheme

Definition 5

A sequence $m_0, m_1, \ldots, m_n$ of pairwise co-prime integers is called compact if $m_1 < \cdots < m_n$ and there exists $\theta \in (0, 1)$ such that

$$m_0 - m_0^\theta < m_i < m_0 + m_0^\theta,$$

for all $1 \leq i \leq n$.

Theorem 6

The GRS secret sharing scheme is asymptotically ideal if it is based on compact sequences of co-primes.

Full details: see (2) and (3) in the course readings section

Outline

Introduction to cryptography

Factorization and Euler's function

Applications of the CRT

Applications of the Jacobi symbol

Course readings

The Cocks cipher

Key generation: public key: (n, a), where

$n = pq$, p and q distinct primes

$a = r^2 \bmod n$ with $r \leftarrow \mathbb{Z}_n^*$

private key: (p, q, r)

Encryption of $m \in \{-1, 1\}$: $t \leftarrow \mathbb{Z}_n^*$ with $\left(\frac{t}{n}\right) = m$

output $c = t + at^{-1} \bmod n$

Decryption of $c \in \mathbb{Z}_n$: output $m = \left(\frac{c+2r}{n}\right)$

Correctness: $\left(\frac{c+2r}{n}\right) = \left(\frac{t^{-1}(t+r)^2}{n}\right) = \left(\frac{t}{n}\right) = m$

More details: see (4) in the course readings section

The Cocks cipher

Example 7 (With artificially small parameters)

Let $p = 61$ and $q = 53$. Then:

- *$n = pq = 3233$ and let $r = 17$. Then, $a = 289$;*
- *$n = 3233$ and $a = 289$ are public parameters; p, q, and r secrete;*

Let $m = -1$ be the message to be encrypted. We choose $t = 951$. Remark that $\left(\frac{951}{3233}\right) = -1 = m$. Then

$$c = t + at^{-1} \bmod n = (951 + 289 \cdot 1625) \bmod 3233 = 1791$$

is the ciphertext.
In order to decrypt c we have to compute

$$\left(\frac{1791 + 2 \cdot 17}{3233}\right) = \left(\frac{1835}{3233}\right) = -1 = m$$

Security of the Cocks cipher

Security issues:

- If a root of a can be efficiently computed, then the system is completely broken;
- If p or q is recovered (e.g., by factoring n in reasonable time), then the system is completely broken;
- $c \equiv t + at^{-1} \bmod n$ is equivalent to $t^2 - ct + a \equiv 0 \bmod n$, for $t \in \mathbb{Z}_n^*$. When a is a quadratic residue modulo n and c is a ciphertext, the congruence may have one, two, or four roots, and all have the same Jacoby symbol. Therefore, if one root can be efficiently computed, the message can be recovered.

Outline

Introduction to cryptography

Factorization and Euler's function

Applications of the CRT

Applications of the Jacobi symbol

Course readings

Course readings

1. F.L. Țiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iași, 2006, pag. 268–283.
2. F.L. Țiplea, C.C. Drăgan: *A Necessary and Sufficient Condition for the Asymptotic Idealness of the GRS Threshold Secret Sharing Scheme*, Information Processing Letters 114 (2014) 299–303.
3. C.C. Drăgan, F.L. Țiplea: *On the asymptotic idealness of the Asmuth-Bloom threshold secret sharing scheme*, Inf. Sci. 463-464: 75-85 (2018).
4. F.L. Țiplea, S. Iftene, G. Teșeleanu, A.-M. Nica: *On the distribution of quadratic residues and non-residues modulo composite integers and applications to cryptography*, Appl. Math. Comput. 372 (2020)
5. S. Iftene: *Secret Sharing Schemes with Applications in Security Protocols*, Ph.D. Thesis, "Al.I.Cuza" University of Iași, 2007, http://thor.info.uaic.ro/~tr/tr.pl.cgi.

Algebraic Foundations of Computer Science

Semigroups and Monoids. Variable-length Codes. Huffman Codes

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Definitions and examples

Definition 1

A semigroup is a pair $(S, \circ)$ which consists of a set S and an associative binary operation $\circ$ on S.

Example 2

1. $(\mathbb{N}, +)$, $(\mathbb{Z}, +)$, $(\mathbb{Q}, +)$, $(\mathbb{R}, +)$, and $(\mathbb{C}, +)$ are (additive) semigroups;
2. $(\mathbb{N}, \cdot)$, $(\mathbb{Z}, \cdot)$, $(\mathbb{Q}, \cdot)$, $(\mathbb{R}, \cdot)$, and $(\mathbb{C}, \cdot)$ are (multiplicative) semigroups;
3. Let $n \in \mathbb{Z}$ and $n\mathbb{Z} = \{n \cdot x | x \in \mathbb{Z}\}$. Then, $(n\mathbb{Z}, +)$ and $(n\mathbb{Z}, \cdot)$ are semigroups;
4. Let $m \in \mathbb{Z}$. Then, $(\mathbb{Z}_m, +)$ and $(\mathbb{Z}_m, \cdot)$, where $+$ and $\cdot$ are the addition and multiplication modulo m, are semigroups.

Definitions and examples

Definition 3

A semigroup $(S, \circ)$ is called commutative if $\circ$ is a commutative operation.

Example 4

All semigroups in Example 2 are commutative.

Remark 1

- *Associativity of a binary operation $\circ$ means that the order of evaluation of an expression $a_1 \circ a_2 \circ a_3$, without changing the order of the terms, is immaterial. In other words, no parenthesis is required for an associative operation;*
- *Commutativity of a binary operation $\circ$ means that the order of the operands in expressions like $a_1 \circ a_2$ is immaterial.*

Definitions and examples

Definition 5

A monoid is a triple $(M, \circ, e)$ which consists of a set M, an associative binary operation $\circ$ on M, and an element $e \in M$ such that

$$x \circ e = e \circ x = x,$$

for any $x \in M$. e is called the identity element or the unity of M.

Remark 2

The identity of any monoid $(M, \circ, e)$ is unique. For, if we assume that e' is an identity too, then:

$$e = e \circ e' = e'.$$

The identity of a monoid $(M, \circ, e)$ is usually denoted by 1_M or even 1.

Definitions and examples

Definition 6

A monoid $(M, \circ, e)$ is called commutative if its binary operation $\circ$ is commutative.

Example 7

1. $(\mathbb{N}, +, 0)$, $(\mathbb{Z}, +, 0)$, $(\mathbb{Q}, +, 0)$, and $(\mathbb{R}, +, 0)$ are commutative monoids;
2. $(\mathbb{N}, \cdot, 1)$, $(\mathbb{Z}, \cdot, 1)$, $(\mathbb{Q}, \cdot, 1)$, and $(\mathbb{R}, \cdot, 1)$ are commutative monoids;
3. $(n\mathbb{Z}, +, 0)$ is a commutative monoid and $(n\mathbb{Z}, \cdot)$ is a commutative semigroup. $(n\mathbb{Z}, \cdot)$ has unity only if $n = 0$ or $n = 1$ and, in such a case it becomes commutative monoid;
4. $(\mathbb{Z}_m, +, 0)$ and $(\mathbb{Z}_m, \cdot, 1)$ are commutative monoids. When $m = 1$, $\mathbb{Z}_1 = \{0\}$ and the multiplicative unity of this monoid is 0.

1. Definitions and examples

A few basic notations:

1. Let $(S, \circ)$ be a semigroup, $A, B \subseteq S$, and $a \in S$. Define:
   - $AB = \{a \circ b | a \in A \ \wedge \ b \in B\}$;
   - $A^1 = A$ and $A^{n+1} = A^n A$, for all $n \geq 1$;
   - $aB = \{a \circ b | b \in B\}$;
   - $a^1 = a$ and $a^{n+1} = a^n \circ a$, for all $n \geq 1$.
2. If $(M, \circ, e)$ is a monoid, $A \subseteq M$, and $a \in M$, we also define:
   - $A^0 = \{e\}$;
   - $a^0 = e$.
3. For any monoid $(M, \circ, e)$ define $S_M = M - \{e\}$.

Definitions and examples

Definition 8

Let $(S, \circ)$ be a semigroup and I a non-empty subset of S.

1. I is called a left ideal of $(S, \circ)$ if $SI \subseteq I$.
2. I is called a right ideal of $(S, \circ)$ if $IS \subseteq I$.
3. I is called an ideal of $(S, \circ)$ if I is a left and a right ideal of $(S, \circ)$.
4. The least (left, right) ideal of $(S, \circ)$ which includes I is called the (left, right) ideal of $(S, \circ)$ generated by I. It is denoted by $\langle I \rangle$.
5. If $I = \{a\}$, then $\langle I \rangle$ is called a (left, right) principal ideal of $(S, \circ)$. It is also denoted by $\langle a \rangle$.

(Left, Right) Ideals of monoids are defined in a similar way.

Example 9

The principal ideal of $(\mathbb{Z}, \cdot, 1)$ generated by $n \in \mathbb{Z}$ is $n\mathbb{Z}$.

Definitions and examples

Definition 10

1. A semigroup $(S', \circ')$ is a sub-semigroup of a semigroup $(S, \circ)$, denoted $(S', \circ') \leq (S, \circ)$, if $S' \subseteq S$ and $\circ' = \circ|_{S'}$.
2. A monoid $(M', \circ', e')$ is a sub-monoid of a monoid $(M, \circ, e)$, denoted $(M', \circ', e') \leq (M, \circ, e)$, if $M' \subseteq M$ and $\circ' = \circ|_{M'}$ and $e' = e$.
3. The least subsemigroup (monoid) of a semigroup (monoid) which includes a given subset A, denoted $\langle A \rangle$, is called the sub-semigroup (sub-monoid) generated by A.
4. A semigroup (monoid) is generated by a subset A of it if it coincides with the sub-semigroup (sub-monoid) generated by A.

The set A in Definition 10(3)(4) is called a set of generators and its elements are called generators.

Definitions and examples

Remark 3

The sub-semigroup (sub-monoid) of a semigroup (monoid), generated by a subset A, is the closure of A under the operation(s) of the host semigroup (monoid):

- *If $(S, \circ)$ is a semigroup and $A \subseteq S$, then the sub-semigroup generated by A is the set of all products*

$$a_1 \circ \cdots \circ a_n,$$

  where $n \geq 1$ and $a_1, \ldots, a_n \in A$;

- *If $(M, \circ, e)$ is a monoid and $A \subseteq S$, then the sub-monoid generated by A is the set of all products*

$$a_1 \circ \cdots \circ a_n,$$

  where $n \geq 1$ and $a_1, \ldots, a_n \in A$, together with the unity e of the host monoid.

Definitions and examples

Example 11

- $(\mathbb{N}, +) \leq (\mathbb{Z}, +) \leq (\mathbb{Q}, +) \leq (\mathbb{R}, +)$;
- $(\mathbb{N}, +, 0) \leq (\mathbb{Z}, +, 0) \leq (\mathbb{Q}, +, 0) \leq (\mathbb{R}, +, 0)$;
- $(\mathbb{N}, \cdot) \leq (\mathbb{Z}, \cdot) \leq (\mathbb{Q}, \cdot) \leq (\mathbb{R}, \cdot)$;
- $(\mathbb{N}, \cdot, 1) \leq (\mathbb{Z}, \cdot, 1) \leq (\mathbb{Q}, \cdot, 1) \leq (\mathbb{R}, \cdot, 1)$;
- The sub-monoid of $(\mathbb{Z}, +, 0)$, generated by $n \in \mathbb{Z}$, is $(n\mathbb{N}, +, 0)$;
- A semigroup (monoid) may have more than one set of generators. For instance, $(\mathbb{Z}, +, 0)$ can be generated by $\{-1, 1\}$ and by $\{-3, 2\}$.

Definitions and examples

Definition 12

1. The order of a semigroup (monoid) is the number of its elements if the semigroup (monoid) is finite, and ∞, otherwise.
2. The order of an element a of a semigroup (monoid) is the order of the sub-semigroup (sub-monoid) generated by a.

Example 13

- $(\mathbb{Z}, +, 0)$ has the order ∞;
- $(\mathbb{Z}_m, +, 0)$ has the order m, if $m \neq 0$. For $m = 0$, $(\mathbb{Z}_m, +, 0)$ has the order ∞.

Definitions and examples

Definition 14

1. A function $f : S \to S'$ is a homomorphism from a semigroup $(S, \circ)$ to a semigroup $(S', \circ')$ if
   - $f(a \circ b) = f(a) \circ' f(b)$, for any $a, b \in S$.
2. A function $f : M \to M'$ is a homomorphism from a monoid $(M, \circ, e)$ to a monoid $(M', \circ', e')$ if
   - $f(a \circ b) = f(a) \circ' f(b)$, for any $a, b \in M$;
   - $f(e) = e'$.

Related concepts:

- injective homomorphism = monomorphism;
- surjective homomorphism = epimorphism;
- bijective homomorphism = isomorphism;

Definitions and examples

- Homomorphism from a semigroup (monoid) to the same semigroup (monoid) = endomorphism;
- Isomorphism from a semigroup (monoid) to the same semigroup (monoid) = automorphism.

Example 15

- The function $f(x) = 2^x$, for any $x \in \mathbb{N}$, is a homomorphism from $(\mathbb{N}, +, 0)$ to $(\mathbb{N}, \cdot, 1)$. Indeed,
  - $f(0) = 2^0 = 1$;
  - $f(x + y) = 2^{x+y} = 2^x \cdot 2^y = f(x) \cdot f(y)$, for any x, y.

  Moreover, f is injective but not surjective. Therefore, f is a monomorphism (but not an epimorphism).

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Word semigroups

Definition 16

An alphabet is any non-empty set. The elements of an alphabet are called letters or symbols.

Example 17

The following sets are alphabets:

- $\Sigma_1 = \{a, b, c\}$;
- $\Sigma_2 = \{0, 1, 2, 3\}$;
- $\Sigma_3 = \{\texttt{begin}, \texttt{end}, \texttt{if}, \texttt{then}, \texttt{else}, \texttt{while}, \texttt{do}\}$.

All letters of an alphabet are assumed indivisible.

Word semigroups

Definition 18

Let Σ be an alphabet. A word of length $k \geq 1$ over Σ is any function $w : \{1, \ldots, k\} \to \Sigma$. The empty function from $\emptyset$ into Σ is called the empty word over Σ and its length is 0.

We usually denote the word w by $w = w(1) \cdots w(k)$, if $k > 0$, and its length k by $|w|$. The empty word is usually denoted by λ.

Example 19

- $w = abaa$ is a word of length 4 over $\Sigma_1 = \{a, b, c\}$;
- 011033 is a word of length 6 over $\Sigma_2 = \{0, 1, 2, 3\}$;
- `begin end` is a word of length 2 over $\Sigma_3 = \{\texttt{begin}, \texttt{end}, \texttt{if}, \texttt{then}, \texttt{else}, \texttt{while}, \texttt{do}\}$.

Word semigroups

Let Σ be an alphabet. Denote:

- $\Sigma^0 = \{\lambda\}$;
- $\Sigma^+ = \bigcup_{k \geq 1} \Sigma^k$,
- $\Sigma^* = \bigcup_{k \geq 0} \Sigma^k = \Sigma^+ \cup \{\lambda\}$.

Words of length 1 are usually identified with letters. Therefore, we may write $\Sigma^1 = \Sigma$.

Definition 20

Two words u and v over the same alphabet Σ are called equal if they have the same length k and $u(i) = v(i)$, for each $1 \leq i \leq k$.

Word semigroups

Definition 21

Let Σ be an alphabet. The binary operation $\cdot : \Sigma^* \times \Sigma^* \to \Sigma^*$ given by

$$w_1 \cdot w_2 : \{i | 1 \leq i \leq |w_1| + |w_2|\} \to \Sigma$$

where

$$(w_1 \cdot w_2)(i) = \begin{cases} w_1(i), & \text{if } 1 \leq i \leq |w_1| \\ w_2(i - |w_1|), & \text{otherwise,} \end{cases}$$

for any i, is called the concatenation or catenation operation on Σ^*.

Example 22

- $abba \cdot bbaa = abbabbaa$;
- $\lambda \cdot w = w \cdot \lambda = w$, for any w.

The concatenation operation symbol is usually omitted. That is, we write uv instead of $u \cdot v$.

Word semigroups

Theorem 23

Let Σ be an alphabet. Then:

(1) $(\Sigma^+, \cdot)$ *is a semigroup generated by* Σ;

(2) $(\Sigma^*, \cdot, \lambda)$ *is a monoid generated by* Σ;

(3) $(\Sigma^*, \cdot, \lambda)$ *is a monoid with simplification;*

(4) $l : \Sigma^* \to \mathbb{N}$ *given by* $l(w) = |w|$, *for any* $w \in \Sigma^*$, *is a homomorphism from* $(\Sigma^*, \cdot, \lambda)$ *to the additive monoid* $(\mathbb{N}, +, 0)$. *Moreover,* $l^{-1}(0) = \{\lambda\}$;

(5) *The group of units of the monoid* $(\Sigma^*, \cdot, \lambda)$ *is trivial.*

Word semigroups

Theorem 24 (Levi's Theorem)

Let x, y, u, and v be words over Σ such that $xy = uv$.

(1) If $|x| < |u|$, then there exists a unique $z \in \Sigma^*$ such that $u = xz$.

(2) If $|x| = |u|$, then $x = u$ and $y = v$.

(3) If $|x| > |u|$, then there exists a unique $z \in \Sigma^*$ such that $x = uz$.

Definition 25

Let Σ be an alphabet and $u, v \in \Sigma^*$.

(1) u is called a prefix or left factor of v if $v = uw$ for some word w.

(2) u is called a suffix or right factor of v if $v = wu$ for some word w.

(3) u is called a sub-word of v if $v = xuy$ for some words x and y.

Word semigroups

Definition 26

(1) A pair $(\Sigma, \prec)$ which consists of an alphabet Σ and a total order $\prec$ on Σ is called an ordered alphabet.

(2) Let $(\Sigma, \prec)$ be an ordered alphabet. The binary relation $\leq_{(\Sigma,\prec)}$ given by

$$x \leq_{(\Sigma,\prec)} y$$

iff

- x is a prefix of y, or
- $x = uav$, $y = ubw$, and $a \prec b$, for some $u, v, w \in \Sigma^*$ and $a, b \in \Sigma$ with $a \neq b$,

is called the direct lexicographic order on $(\Sigma, \prec)$.

In a similar way one can define the inverse lexicographic order on ordered alphabets.

Word semigroups

λ	λ
a	a
aa	b
aaa	aa
$\dots$	ab
$aaaab$	ba
$aaab$	bb
aab	aaa
ab	aab
$\dots$	$\dots$
b	bbb
a)	b)

a) Lexicographic order
b) Lexicographic order on words of the same length.

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Cyclic semigroups

If $\mathbf{S} = (S, \circ)$ is a semigroup and $a \in S$, then

$$\langle a \rangle_{\mathbf{S}} = \{a, a^2, \ldots, a^n, \ldots\}$$

If $\mathbf{M} = (M, \circ, e)$ is a monoid $a \in M$, then

$$\langle a \rangle_{\mathbf{M}} = \{e = a^0, a, a^2, \ldots, a^n, \ldots\}$$

Definition 27

A semigroup (monoid) generated by one of its elements is called a cyclic semigroup (cyclic monoid).

$\mathbf{S}$ cyclic semigroup $\Rightarrow S = \{a, a^2, \ldots, a^n, \ldots\}$, for some $a \in S$.

$\mathbf{M}$ cyclic monoid $\Rightarrow M = \{e = a^0, a, a^2, \ldots, a^n, \ldots\}$, for some $a \in M$.

Cyclic semigroups

Theorem 28

Let a be an element of a semigroup $(S, \circ)$. Then, exactly one of the following two properties is satisfied:

(1) $a^n \neq a^m$ *for any* $n \neq m$, *and the semigroup generated by* a *is isomorphic with* $(\mathbb{N} - \{0\}, +)$;

(2) *there exists* $m > 0$ *and* $r > 0$ *such that :*

  (a) $a^m = a^{m+r}$;

  (b) $a^{m+u} = a^{m+v}$ *iff* $u \equiv v \mod r$, *for any* $u, v \in \mathbb{N}$;

  (c) $\langle a \rangle = \{a, a^2, \ldots, a^{m+r-1}\}$ *has exactly* $m + r - 1$ *elements;*

  (d) $K(a) = \{a^m, \ldots, a^{m+r-1}\}$ *is a cyclic subgroup of* $\langle a \rangle$.

The number m in Theorem 28(2) is called the index of a, and r is called the period of a, in $(S, \circ)$. The following property holds true:

order(a)=index(a)+period(a)-1

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Free semigroups and monoids

Remark 4

- *The monoid* $(\mathbb{N}, +, 0)$ *can be generated by* $\{1\}$*. Moreover, any number* $n \in \mathbb{N} - \{0\}$ *can uniquely be written as a finite combination of* 1*'s under* $+$*, namely,*

$$n = \underbrace{1 + 1 + \cdots + 1}_{n \text{ times}}.$$

We say that $\{1\}$ *freely generates the monoid.*

Definition 29

A semigroup $(S, \circ)$ is freely generated by a subset $X \subseteq S$ if any element of $s \in S$ can uniquely be written as a finite combination of elements of X,

$$s = x_1 \circ \cdots \circ x_n,$$

where $x_1, \ldots, x_n \in X$ and $n \geq 1$.

Free semigroups and monoids

Definition 30

A monoid $(M, \circ, e)$ is freely generated by a subset $X \subseteq M$ if $(S_M, \circ)$ is freely generated by X.

Definition 31

A free semigrup (free monoid) is a semigroup (monoid) which can be freely generated by some subset of it.

If A freely generates a semigroup, then it is called a set of free generators of the semigroup (monoid).

Example 32

- $(\mathbb{N}, +, 0)$ is a free monoid.
- X^+ (X^*) together with the concatenation operation is a free semigroup (monoid), for any non-empty set X.
- $(\mathbb{Z}, +, 0)$ is not a free monoid.

Free semigroups and monoids

Free semigroups (monoids) have very important properties.

Theorem 33 (The universality property)

If $(S, \circ)$ is a semigroup freely generated by X, then for any semigroup $(T, *)$ and any function $f : X \to T$, there exists a unique homomorphism $h : S \to T$ which extends f (that is, $h(x) = f(x)$, for any $x \in X$).

The universality property can be similarly formulated for free monoids.

Corollary 34

Any free semigroup (monoid) is isomorphic with a word semigroup (monoid).

Free semigroups and monoids

The universality property allows us to define homomorphisms from free semigroups $(S, \circ)$ to semigroups $(T, *)$ just by defining them on sets of free generators of $(S, \circ)$.

Example 35

To define a homomorphism from $(\mathbb{N}, +, 0)$ to $(\mathbb{N}, \cdot, 1)$ it is sufficient to consider an arbitrary function from $\{1\}$, which freely generates $(\mathbb{N}, +, 0)$, to $(\mathbb{N}, \cdot, 1)$. For example, if we consider the function $f(1) = 10$, then the unique homomorphism induced by f is:

- $h(0) = 1$;
- $h(1) = f(1) = 10$;
- $h(2) = h(1 + 1) = h(1) \cdot h(1) = 10^2$;
- $h(3) = h(1 + 1 + 1) = h(1) \cdot h(1) \cdot h(1) = 10^3$;
- $h(n) = 10^n$, for any $n \geq 0$.

Free semigroups and monoids

How many sets of free generators may have a free semigroup or monoid ?

Proposition 1

If a semigroup $(S, \circ)$ (monoid $(M, \circ, e)$) is free, then it has a unique set of free generates, and this set is $S - S^2$ ($S_M - S_M^2$).

Proof.

First, show that any set of generators should include $S - S^2$ ($S_M - S_M^2$). Then, show that any set X of generators should be a subset of $S - S^2$ ($S_M - S_M^2$). $\square$

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Variable-length codes

Definition 36

Let A be a non-empty set. A variable-length code (or simply code) over A is any subset $C \subseteq A^+$ such that C^* is a free sub-monoid of A^*. The elements of C are called code words.

Equivalent definitions:

1. C is a code over A if any code sequence $w \in C^+$ can be uniquely decomposed into code words
$$w = c_1 \cdots c_n;$$
2. C is a code over A if
$$u_1 \cdots u_m = v_1 \cdots v_n \quad \Rightarrow \quad n = m \ \wedge \ (\forall i)(u_i = v_i),$$
for any $u_1, \ldots, u_m, v_1, \ldots, v_n \in C$;
3. C is a code over A if
$$u_1 \cdots u_m = v_1 \cdots v_n \quad \Rightarrow \quad u_1 = v_1,$$
for any $u_1, \ldots, u_m, v_1, \ldots, v_n \in C$.

Variable-length codes

Example 37

- $C = \{a, ab, ba\}$ is not a code because $aba = (ab)a = a(ba)$;
- $C = \{a, bb, aab, bab\}$ is a code.

Definition 38

1. C is a prefix code if no code word of C is a prefix of any other code word.
2. C is a suffix code if no code word of C is a suffix of any other code word.
3. C is a block code if all code words of C have the same length.

Example 39

- ASCII is a block code.

Variable-length codes

Given a non-empty set $C \subseteq A^+$, define

- $C_1 = \{x \in A^+ | (\exists c \in C)(cx \in C)\}$,
- $C_{i+1} = \{x \in A^+ | (\exists c \in C)(cx \in C_i) \ \vee \ (\exists c \in C_i)(cx \in C)\}$, for any $i \geq 1$.

We get an infinite sequence of sets of words:

$$C_1, C_2, C_3, \ldots$$

Remark 5

If C is finite, then there are i and j such that $j < i$ and $C_i = C_j$.

Theorem 40 (Sardinas-Patterson Theorem)

C is a code over A iff $C \cap C_i = \emptyset$, for any $i \geq 1$.

Variable-length codes

Sardinas-Patterson Algorithm

input: finite non-empty set $C \subseteq A^+$;
output: $code(C) = 1$, if C is a code, and $code(C) = 0$, otherwise;
begin
  $C_1 := \{x \in A^+ | (\exists c \in C)(cx \in C)\}$;
  if $C \cap C_1 \neq \emptyset$ **then** $code(C) := 0$
    else begin
      $i := 1$; $cont := 1$;
      while $cont = 1$ **do**
        begin
          $i := i + 1$;
          $C_i := \{x \in A^+ | (\exists c \in C_{i-1})(cx \in C) \vee (\exists c \in C)(cx \in C_{i-1})\}$;
          if $C \cap C_i \neq \emptyset$ **then begin** $code(C) := 0$; $cont := 0$ **end**
            else if $(\exists j < i)(C_i = C_j)$
              then begin $code(C) := 1$; $cont := 0$ **end**;
        end;
    end;
end.

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Huffman codes

- Have been proposed by David Huffman in 1952;
- Are used to encode information sources (an information source is a device which outputs symbols from a given alphabet according to certain probabilities depending, in general, on preceding choices as well as the particular symbol in question);
- Are prefix codes of minimum length among all the prefix codes associated to a given information source;
- Associate short code words to highly probable symbols (which appear more frequently), and longer code words to symbols with smaller probabilities.

Huffman codes

Definition 41

An information source is a couple $IS = (A, \pi)$, where A is a non-empty and at most countable set, called the source alphabet, and π is a probability distribution on A.

Only finite information sources will be considered.

Definition 42

Let $IS = (A, \pi)$ be an information source and $h : A \to \Sigma^*$ be a homomorphism. The (average) length of h with respect to IS is

$$L_h(IS) = \sum_{a \in A} |h(a)|\pi(a).$$

Huffman codes

Definition 43

Let $IS = (A, \pi)$ be an information source and $h : A \to \Sigma^*$ be a homomorphism. h is called a code or encoding of IS if $C = \{h(a) | a \in A\}$ is a code.

Example 44

Let IS be the information source

A	a	b	c	d	e	f
π	0.4	0.1	0.1	0.1	0.2	0.1

and h be the encoding

A	a	b	c	d	e	f
h	0	1100	1101	1110	10	1111

Then, the length of h w.r.t. IS is $L_h(IS) = 2.4$. That is, on average, 2.4 bits are needed to encode any symbol of the information source.

Huffman codes

Definition 45

Let $IS = (A, \pi)$ be an information source and $h : A \to \Sigma^*$ be an encoding for IS. h is called a Huffman code or a Huffman encoding of IS if:

- $C = \{h(a) | a \in A\}$ is a prefix code;
- h has minimum length among all the prefix codes of IS.

Given an information source IS, are there Huffman encodings for IS ?
The answer is positive.

Huffman codes

Huffman algorithm:

1. let IS be an information source with $n \geq 2$ symbols

A	a_1	a_2	$\cdots$	a_{n-1}	a_n
π	p_1	p_2	$\cdots$	p_{n-1}	p_n

   where $p_1 \geq p_2 \geq \cdots \geq p_{n-1} \geq p_n$;

2. if $n = 2$, then $h(a_1) = 0$ and $h(a_2) = 1$ is a Huffman code for IS;
3. if $n \geq 3$, then compute a reduced source IS' for IS

A'	a_1	a_2	$\cdots$	a_{n-2}	$a_{n-1,n}$
π'	p_1	p_2	$\cdots$	p_{n-2}	$p_{n-1,n}$

   where $p_{n-1,n} = p_{n-1} + p_n$;

Huffman codes

4. if h' is a Huffman code for IS', then h given by

$$h(x) = \begin{cases} h'(x), & \text{if } x \notin \{a_{n-1}, a_n\} \\ h'(x)0, & \text{if } x = a_{n-1} \\ h'(x)1, & \text{if } x = a_n, \end{cases}$$

is a Huffman code for IS.

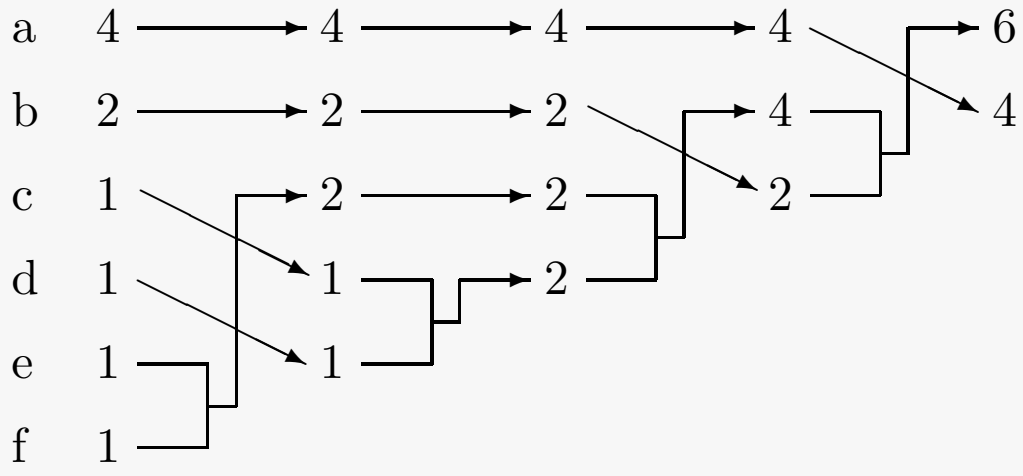
Huffman codes

Example 46

Let IS be the following information source:

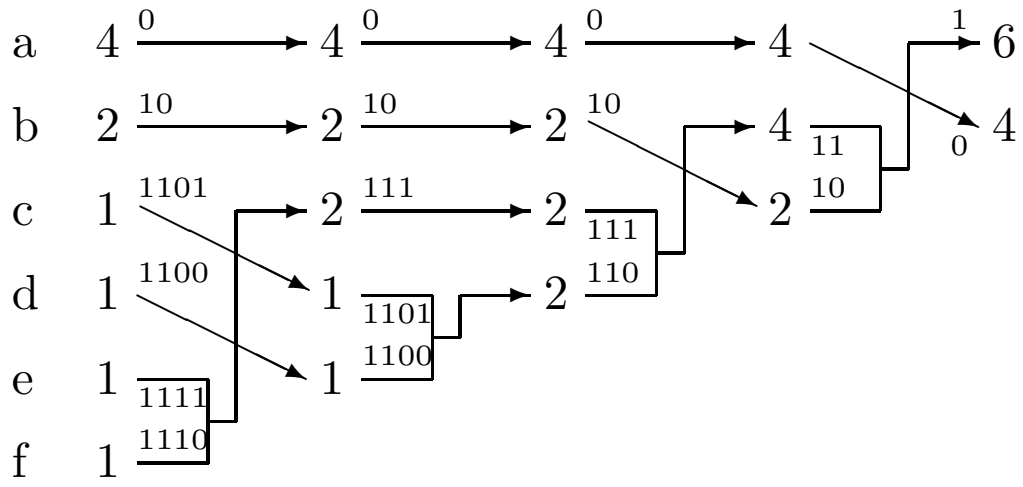
A	a	b	c	d	e	f
π	0.4	0.2	0.1	0.1	0.1	0.1

Compute a sequence of reduced sources for IS:



Huffman codes

Assign codes to each reduced source from right to left:



The Huffman code is $h(a) = 0$, $h(b) = 10$, $h(c) = 1101$, $h(d) = 1100$, $h(e) = 1111$, and $h(f) = 1110$. The length of h is 2.4. It is the minimum length code among all the prefix codes associated to IS.

Huffman codes

Huffman codes can be used to compress data as follows. Let α be a text:

1. parse α and, for each symbol a in α compute its number of occurrences;
2. let IS be the information source thus obtained. Compute a Huffman code h for IS;
3. encode α by $h(\alpha)$ (obtained by replacing each symbol a in α by $h(a)$).

Compression ratio = is the ratio of the size of the original data to the size of the compressed data.

Compression rate = is the rate of the compressed data (typically, it is in units of bits/sample, bits/character, bits/pixels, or bits/second).

Huffman codes

There are two types of data compression:

- lossless data compression – allows the exact original data to be reconstructed from the compressed data;
- lossy data compression – does not allow the exact original data to be reconstructed from the compressed data.

Data compression by Huffman codes is an example of lossless data compression.

Is there any limit to lossless data compression?

The answer is positive. The limit is called the entropy. The exact value of the entropy depends on the (statistical nature of the) information source. It is possible to compress the source, in a lossless manner, with compression rate close to its entropy. It is mathematically impossible to do better than that.

Huffman codes

Definition 47

Let S be an information source with n symbols and probabilities $p_1, \ldots, p_n$. The entropy of S, denoted $H(S)$ or $H(p_1, \ldots, p_n)$, is defined by

$$H(S) = \sum_{i=1}^{n} p_i \log(1/p_i)$$

(mathematical convention: $0 \cdot \log(1/0) = 0$).

Proposition 2

For any distribution of probability $p_1, \ldots, p_n$, the following properties hold:

1. $0 \leq H(p_1, \ldots, p_n) \leq \log n$;
2. $H(p_1, \ldots, p_n) = 0$ *iff* $p_i = 1$, *for some* i;
3. $H(p_1, \ldots, p_n) = \log n$ *iff* $p_i = 1/n$, *for any* i.

Huffman codes

Definition 48

Let $S_1 = (\{a_i | 1 \leq i \leq n\}, (p_i | 1 \leq i \leq n))$ and $S_2 = (\{b_j | 1 \leq j \leq m\}, (q_j | 1 \leq j \leq m))$ be two information sources. The product of S_1 and S_2, denoted $S_1 \circ S_2$, is the information source

$$S_1 \circ S_2 = (\{(a_i, a_j) | 1 \leq i \leq n,\ 1 \leq j \leq m\}, (p_i \cdot q_j | 1 \leq i \leq n,\ 1 \leq j \leq m)).$$

Proposition 3

For any finite information sources S_1 and S_2,

$$H(S_1 \circ S_2) = H(S_1) + H(S_2).$$

We denote $\underbrace{S \circ \cdots \circ S}_{k \text{ times}}$ by S^k, where S is a finite information source.

Then,

$$H(S^k) = kH(S).$$

Huffman codes

Theorem 49 (Shannon's noiseless coding theorem)

Let S be an information source. Then:

(1) $H(S) \leq L_h(S)$, *for any code h of S;*

(2) $H(S) \leq L_h(S) < H(S) + 1$, *for any Huffman code h of S;*

(3) $\lim_{k\to\infty} \frac{L_{min}(S^k)}{k} = H(S)$, *where $L_{min}(S')$ is the average length of some Huffman code for S'.*

Shannon's noiseless coding theorem places a lower bound on the minimal possible expected length of an encoding of a source S, as a function of the entropy of S.

Huffman codes

The design of a Huffman encoding for an input $w \in \Sigma^+$ requires two steps:

- determine the frequency of occurrences of each letter a in w;
- design a Huffman code for Σ w.r.t. the probability distribution

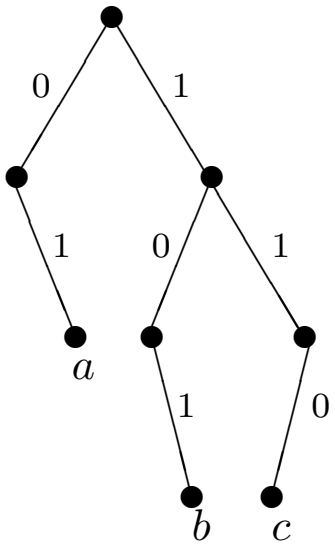
$$p_a = \frac{\text{frequency of } a \text{ in } w}{|w|}.$$

  Then, encode w by this Huffman code.

Because this procedure requires two parsings of the input, it is time-consuming for large inputs (although the compression rate by such an encoding is optimal). In practice, an alternative method which requires only one parsing of the input is used. It is called the adaptive Huffman encoding.

Huffman codes

A useful graphical representation of a finite code $C \subseteq A^+$ consists of a tree with nodes labeled by symbols in A such that the code words are exactly the sequences of labels collected from the root to leaves. For example, the tree below is the graphical representation of the prefix code $\{01, 110, 101\}$, where a is encoded by 01, b by 101, and c by 110.



Huffman codes

The encoding of an input w by the adaptive Huffman technique is based on the construction of a sequence of Huffman trees as follows:

- start initially with a Huffman tree $\mathcal{T}_0$ associated to the alphabet A (each symbol of A has frequency 1);
- if $\mathcal{T}_n$ is the current Huffman tree and the current input symbol is a (that is, $w = uav$ and u has been already processed), then output the code of a in $\mathcal{T}_n$ (this code is denoted by $code(a, \mathcal{T}_n)$) and update the tree $\mathcal{T}_n$ by applying to it the sibling transformation; the new tree is denoted $\mathcal{T}_{n+1}$.

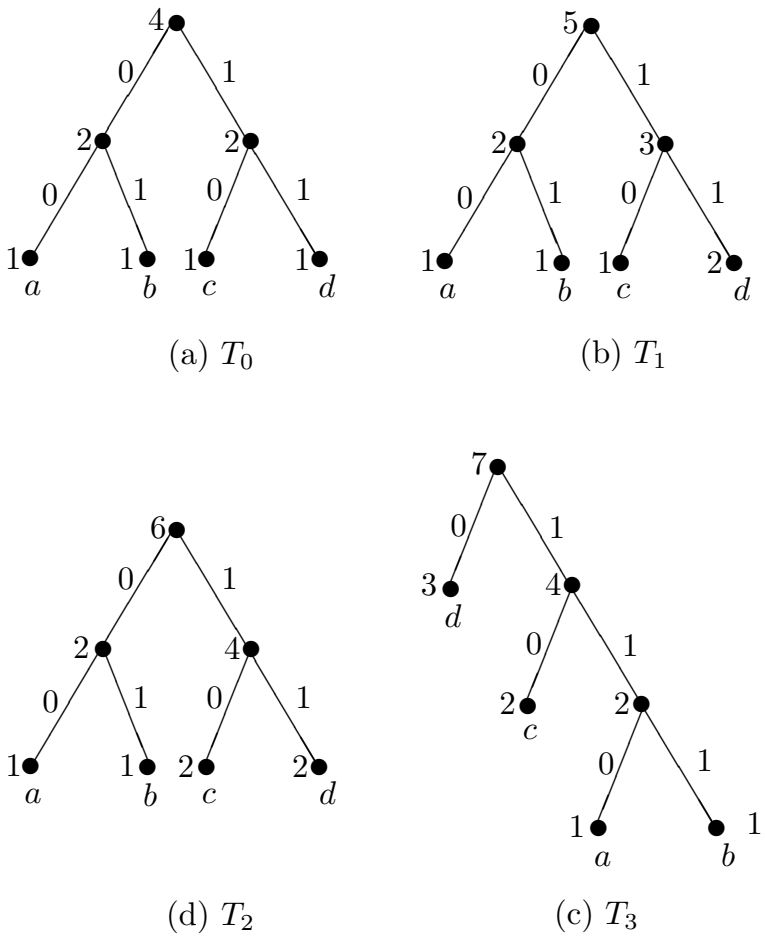
Huffman codes

The sibling transformation applied to symbol a and tree $\mathcal{T}_n$ consists of:

1. compare a to its successors in the tree (from left to right and from bottom to top). If the immediate successor has frequency $k+1$ or greater, where k is the frequency of a in $\mathcal{T}_n$, then the nodes are still in sorted order and there is no need to change anything. Otherwise, a should be swapped with the last successor which has frequency k or smaller (except that a should not be swapped with its parent);
2. increment the frequency of a (from k to $k+1$);
3. if a is the root, the loop halts; otherwise, the loop repeats with the parent of a.

Huffman codes

A sequence of Huffman trees used to encode the string *dcd* over the alphabet $\{a, b, c, d\}$:



(a) T_0 (b) T_1

(d) T_2 (c) T_3

Huffman codes

Huffman adaptive is not a variable-length code! The same character may be encoded by different code words!

Huffman adaptive is a time-varying code! For more details regarding time-varying codes see TMTE2002.pdf in the course web site.

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Course readings

1. F.L. Ţiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iaşi, 2006, pag. 179–243.

2. F.L. Ţiplea, E. Mäkinen, D. Trincă, C. Enea: *Characterization Results for Time-varying Codes*, Fundamenta Informaticae 53(2), 2002, 185-198.

Algebraic Foundations of Computer Science

Semigroups and Monoids. Variable-length Codes. Huffman Codes

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Definitions and examples

Definition 1

A semigroup is a pair $(S, \circ)$ which consists of a set S and an associative binary operation $\circ$ on S.

Example 2

1. $(\mathbb{N}, +)$, $(\mathbb{Z}, +)$, $(\mathbb{Q}, +)$, $(\mathbb{R}, +)$, and $(\mathbb{C}, +)$ are (additive) semigroups;
2. $(\mathbb{N}, \cdot)$, $(\mathbb{Z}, \cdot)$, $(\mathbb{Q}, \cdot)$, $(\mathbb{R}, \cdot)$, and $(\mathbb{C}, \cdot)$ are (multiplicative) semigroups;
3. Let $n \in \mathbb{Z}$ and $n\mathbb{Z} = \{n \cdot x | x \in \mathbb{Z}\}$. Then, $(n\mathbb{Z}, +)$ and $(n\mathbb{Z}, \cdot)$ are semigroups;
4. Let $m \in \mathbb{Z}$. Then, $(\mathbb{Z}_m, +)$ and $(\mathbb{Z}_m, \cdot)$, where $+$ and $\cdot$ are the addition and multiplication modulo m, are semigroups.

Definitions and examples

Definition 3

A semigroup $(S, \circ)$ is called commutative if $\circ$ is a commutative operation.

Example 4

All semigroups in Example 2 are commutative.

Remark 1

- *Associativity of a binary operation $\circ$ means that the order of evaluation of an expression $a_1 \circ a_2 \circ a_3$, without changing the order of the terms, is immaterial. In other words, no parenthesis is required for an associative operation;*
- *Commutativity of a binary operation $\circ$ means that the order of the operands in expressions like $a_1 \circ a_2$ is immaterial.*

Definitions and examples

Definition 5

A monoid is a triple $(M, \circ, e)$ which consists of a set M, an associative binary operation $\circ$ on M, and an element $e \in M$ such that

$$x \circ e = e \circ x = x,$$

for any $x \in M$. e is called the identity element or the unity of M.

Remark 2

The identity of any monoid $(M, \circ, e)$ is unique. For, if we assume that e' is an identity too, then:

$$e = e \circ e' = e'.$$

The identity of a monoid $(M, \circ, e)$ is usually denoted by 1_M or even 1.

Definitions and examples

Definition 6

A monoid $(M, \circ, e)$ is called commutative if its binary operation $\circ$ is commutative.

Example 7

1. $(\mathbb{N}, +, 0)$, $(\mathbb{Z}, +, 0)$, $(\mathbb{Q}, +, 0)$, and $(\mathbb{R}, +, 0)$ are commutative monoids;
2. $(\mathbb{N}, \cdot, 1)$, $(\mathbb{Z}, \cdot, 1)$, $(\mathbb{Q}, \cdot, 1)$, and $(\mathbb{R}, \cdot, 1)$ are commutative monoids;
3. $(n\mathbb{Z}, +, 0)$ is a commutative monoid and $(n\mathbb{Z}, \cdot)$ is a commutative semigroup. $(n\mathbb{Z}, \cdot)$ has unity only if $n = 0$ or $n = 1$ and, in such a case it becomes commutative monoid;
4. $(\mathbb{Z}_m, +, 0)$ and $(\mathbb{Z}_m, \cdot, 1)$ are commutative monoids. When $m = 1$, $\mathbb{Z}_1 = \{0\}$ and the multiplicative unity of this monoid is 0.

1. Definitions and examples

A few basic notations:

1. Let $(S, \circ)$ be a semigroup, $A, B \subseteq S$, and $a \in S$. Define:
   - $AB = \{a \circ b | a \in A \ \wedge \ b \in B\}$;
   - $A^1 = A$ and $A^{n+1} = A^n A$, for all $n \geq 1$;
   - $aB = \{a \circ b | b \in B\}$;
   - $a^1 = a$ and $a^{n+1} = a^n \circ a$, for all $n \geq 1$.
2. If $(M, \circ, e)$ is a monoid, $A \subseteq M$, and $a \in M$, we also define:
   - $A^0 = \{e\}$;
   - $a^0 = e$.
3. For any monoid $(M, \circ, e)$ define $S_M = M - \{e\}$.

Definitions and examples

Definition 8

Let $(S, \circ)$ be a semigroup and I a non-empty subset of S.

1. I is called a left ideal of $(S, \circ)$ if $SI \subseteq I$.
2. I is called a right ideal of $(S, \circ)$ if $IS \subseteq I$.
3. I is called an ideal of $(S, \circ)$ if I is a left and a right ideal of $(S, \circ)$.
4. The least (left, right) ideal of $(S, \circ)$ which includes I is called the (left, right) ideal of $(S, \circ)$ generated by I. It is denoted by $\langle I \rangle$.
5. If $I = \{a\}$, then $\langle I \rangle$ is called a (left, right) principal ideal of $(S, \circ)$. It is also denoted by $\langle a \rangle$.

(Left, Right) Ideals of monoids are defined in a similar way.

Example 9

The principal ideal of $(\mathbb{Z}, \cdot, 1)$ generated by $n \in \mathbb{Z}$ is $n\mathbb{Z}$.

Definitions and examples

Definition 10

1. A semigroup $(S', \circ')$ is a sub-semigroup of a semigroup $(S, \circ)$, denoted $(S', \circ') \leq (S, \circ)$, if $S' \subseteq S$ and $\circ' = \circ|_{S'}$.
2. A monoid $(M', \circ', e')$ is a sub-monoid of a monoid $(M, \circ, e)$, denoted $(M', \circ', e') \leq (M, \circ, e)$, if $M' \subseteq M$ and $\circ' = \circ|_{M'}$ and $e' = e$.
3. The least subsemigroup (monoid) of a semigroup (monoid) which includes a given subset A, denoted $\langle A \rangle$, is called the sub-semigroup (sub-monoid) generated by A.
4. A semigroup (monoid) is generated by a subset A of it if it coincides with the sub-semigroup (sub-monoid) generated by A.

The set A in Definition 10(3)(4) is called a set of generators and its elements are called generators.

Definitions and examples

Remark 3

The sub-semigroup (sub-monoid) of a semigroup (monoid), generated by a subset A, is the closure of A under the operation(s) of the host semigroup (monoid):

- *If $(S, \circ)$ is a semigroup and $A \subseteq S$, then the sub-semigroup generated by A is the set of all products*

$$a_1 \circ \cdots \circ a_n,$$

  where $n \geq 1$ and $a_1, \ldots, a_n \in A$;

- *If $(M, \circ, e)$ is a monoid and $A \subseteq S$, then the sub-monoid generated by A is the set of all products*

$$a_1 \circ \cdots \circ a_n,$$

  where $n \geq 1$ and $a_1, \ldots, a_n \in A$, together with the unity e of the host monoid.

Definitions and examples

Example 11

- $(\mathbb{N}, +) \leq (\mathbb{Z}, +) \leq (\mathbb{Q}, +) \leq (\mathbb{R}, +)$;
- $(\mathbb{N}, +, 0) \leq (\mathbb{Z}, +, 0) \leq (\mathbb{Q}, +, 0) \leq (\mathbb{R}, +, 0)$;
- $(\mathbb{N}, \cdot) \leq (\mathbb{Z}, \cdot) \leq (\mathbb{Q}, \cdot) \leq (\mathbb{R}, \cdot)$;
- $(\mathbb{N}, \cdot, 1) \leq (\mathbb{Z}, \cdot, 1) \leq (\mathbb{Q}, \cdot, 1) \leq (\mathbb{R}, \cdot, 1)$;
- The sub-monoid of $(\mathbb{Z}, +, 0)$, generated by $n \in \mathbb{Z}$, is $(n\mathbb{N}, +, 0)$;
- A semigroup (monoid) may have more than one set of generators. For instance, $(\mathbb{Z}, +, 0)$ can be generated by $\{-1, 1\}$ and by $\{-3, 2\}$.

Definitions and examples

Definition 12

1. The order of a semigroup (monoid) is the number of its elements if the semigroup (monoid) is finite, and ∞, otherwise.
2. The order of an element a of a semigroup (monoid) is the order of the sub-semigroup (sub-monoid) generated by a.

Example 13

- $(\mathbb{Z}, +, 0)$ has the order ∞;
- $(\mathbb{Z}_m, +, 0)$ has the order m, if $m \neq 0$. For $m = 0$, $(\mathbb{Z}_m, +, 0)$ has the order ∞.

Definitions and examples

Definition 14

1. A function $f : S \to S'$ is a homomorphism from a semigroup $(S, \circ)$ to a semigroup $(S', \circ')$ if
   - $f(a \circ b) = f(a) \circ' f(b)$, for any $a, b \in S$.
2. A function $f : M \to M'$ is a homomorphism from a monoid $(M, \circ, e)$ to a monoid $(M', \circ', e')$ if
   - $f(a \circ b) = f(a) \circ' f(b)$, for any $a, b \in M$;
   - $f(e) = e'$.

Related concepts:

- injective homomorphism = monomorphism;
- surjective homomorphism = epimorphism;
- bijective homomorphism = isomorphism;

Definitions and examples

- Homomorphism from a semigroup (monoid) to the same semigroup (monoid) = endomorphism;
- Isomorphism from a semigroup (monoid) to the same semigroup (monoid) = automorphism.

Example 15

- The function $f(x) = 2^x$, for any $x \in \mathbb{N}$, is a homomorphism from $(\mathbb{N}, +, 0)$ to $(\mathbb{N}, \cdot, 1)$. Indeed,
  - $f(0) = 2^0 = 1$;
  - $f(x + y) = 2^{x+y} = 2^x \cdot 2^y = f(x) \cdot f(y)$, for any x, y.

  Moreover, f is injective but not surjective. Therefore, f is a monomorphism (but not an epimorphism).

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Word semigroups

Definition 16

An alphabet is any non-empty set. The elements of an alphabet are called letters or symbols.

Example 17

The following sets are alphabets:

- $\Sigma_1 = \{a, b, c\}$;
- $\Sigma_2 = \{0, 1, 2, 3\}$;
- $\Sigma_3 = \{\texttt{begin}, \texttt{end}, \texttt{if}, \texttt{then}, \texttt{else}, \texttt{while}, \texttt{do}\}$.

All letters of an alphabet are assumed indivisible.

Word semigroups

Definition 18

Let Σ be an alphabet. A word of length $k \geq 1$ over Σ is any function $w : \{1, \ldots, k\} \to \Sigma$. The empty function from $\emptyset$ into Σ is called the empty word over Σ and its length is 0.

We usually denote the word w by $w = w(1) \cdots w(k)$, if $k > 0$, and its length k by $|w|$. The empty word is usually denoted by λ.

Example 19

- $w = abaa$ is a word of length 4 over $\Sigma_1 = \{a, b, c\}$;
- 011033 is a word of length 6 over $\Sigma_2 = \{0, 1, 2, 3\}$;
- `begin end` is a word of length 2 over $\Sigma_3 = \{\texttt{begin}, \texttt{end}, \texttt{if}, \texttt{then}, \texttt{else}, \texttt{while}, \texttt{do}\}$.

Word semigroups

Let Σ be an alphabet. Denote:

- $\Sigma^0 = \{\lambda\}$;
- $\Sigma^+ = \bigcup_{k \geq 1} \Sigma^k$,
- $\Sigma^* = \bigcup_{k \geq 0} \Sigma^k = \Sigma^+ \cup \{\lambda\}$.

Words of length 1 are usually identified with letters. Therefore, we may write $\Sigma^1 = \Sigma$.

Definition 20

Two words u and v over the same alphabet Σ are called equal if they have the same length k and $u(i) = v(i)$, for each $1 \leq i \leq k$.

Word semigroups

Definition 21

Let Σ be an alphabet. The binary operation $\cdot : \Sigma^* \times \Sigma^* \to \Sigma^*$ given by

$$w_1 \cdot w_2 : \{i | 1 \leq i \leq |w_1| + |w_2|\} \to \Sigma$$

where

$$(w_1 \cdot w_2)(i) = \begin{cases} w_1(i), & \text{if } 1 \leq i \leq |w_1| \\ w_2(i - |w_1|), & \text{otherwise,} \end{cases}$$

for any i, is called the concatenation or catenation operation on Σ^*.

Example 22

- $abba \cdot bbaa = abbabbaa$;
- $\lambda \cdot w = w \cdot \lambda = w$, for any w.

The concatenation operation symbol is usually omitted. That is, we write uv instead of $u \cdot v$.

Word semigroups

Theorem 23

Let Σ be an alphabet. Then:

(1) $(\Sigma^+, \cdot)$ *is a semigroup generated by* Σ;

(2) $(\Sigma^*, \cdot, \lambda)$ *is a monoid generated by* Σ;

(3) $(\Sigma^*, \cdot, \lambda)$ *is a monoid with simplification;*

(4) $l : \Sigma^* \to \mathbb{N}$ *given by* $l(w) = |w|$, *for any* $w \in \Sigma^*$, *is a homomorphism from* $(\Sigma^*, \cdot, \lambda)$ *to the additive monoid* $(\mathbb{N}, +, 0)$. *Moreover,* $l^{-1}(0) = \{\lambda\}$;

(5) *The group of units of the monoid* $(\Sigma^*, \cdot, \lambda)$ *is trivial.*

Word semigroups

Theorem 24 (Levi's Theorem)

Let x, y, u, and v be words over Σ such that $xy = uv$.

(1) If $|x| < |u|$, then there exists a unique $z \in \Sigma^*$ such that $u = xz$.

(2) If $|x| = |u|$, then $x = u$ and $y = v$.

(3) If $|x| > |u|$, then there exists a unique $z \in \Sigma^*$ such that $x = uz$.

Definition 25

Let Σ be an alphabet and $u, v \in \Sigma^*$.

(1) u is called a prefix or left factor of v if $v = uw$ for some word w.

(2) u is called a suffix or right factor of v if $v = wu$ for some word w.

(3) u is called a sub-word of v if $v = xuy$ for some words x and y.

Word semigroups

Definition 26

(1) A pair $(\Sigma, \prec)$ which consists of an alphabet Σ and a total order $\prec$ on Σ is called an ordered alphabet.

(2) Let $(\Sigma, \prec)$ be an ordered alphabet. The binary relation $\leq_{(\Sigma,\prec)}$ given by

$$x \leq_{(\Sigma,\prec)} y$$

iff

- x is a prefix of y, or
- $x = uav$, $y = ubw$, and $a \prec b$, for some $u, v, w \in \Sigma^*$ and $a, b \in \Sigma$ with $a \neq b$,

is called the direct lexicographic order on $(\Sigma, \prec)$.

In a similar way one can define the inverse lexicographic order on ordered alphabets.

Word semigroups

λ	λ
a	a
aa	b
aaa	aa
$\dots$	ab
$aaaab$	ba
$aaab$	bb
aab	aaa
ab	aab
$\dots$	$\dots$
b	bbb
a)	b)

a) Lexicographic order
b) Lexicographic order on words of the same length.

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Cyclic semigroups

If $\mathbf{S} = (S, \circ)$ is a semigroup and $a \in S$, then

$$\langle a \rangle_{\mathbf{S}} = \{a, a^2, \ldots, a^n, \ldots\}$$

If $\mathbf{M} = (M, \circ, e)$ is a monoid $a \in M$, then

$$\langle a \rangle_{\mathbf{M}} = \{e = a^0, a, a^2, \ldots, a^n, \ldots\}$$

Definition 27

A semigroup (monoid) generated by one of its elements is called a cyclic semigroup (cyclic monoid).

$\mathbf{S}$ cyclic semigroup $\Rightarrow S = \{a, a^2, \ldots, a^n, \ldots\}$, for some $a \in S$.

$\mathbf{M}$ cyclic monoid $\Rightarrow M = \{e = a^0, a, a^2, \ldots, a^n, \ldots\}$, for some $a \in M$.

Cyclic semigroups

Theorem 28

Let a be an element of a semigroup $(S, \circ)$. Then, exactly one of the following two properties is satisfied:

(1) $a^n \neq a^m$ *for any* $n \neq m$, *and the semigroup generated by* a *is isomorphic with* $(\mathbb{N} - \{0\}, +)$;

(2) *there exists* $m > 0$ *and* $r > 0$ *such that :*

  (a) $a^m = a^{m+r}$;

  (b) $a^{m+u} = a^{m+v}$ *iff* $u \equiv v \mod r$, *for any* $u, v \in \mathbb{N}$;

  (c) $\langle a \rangle = \{a, a^2, \ldots, a^{m+r-1}\}$ *has exactly* $m + r - 1$ *elements;*

  (d) $K(a) = \{a^m, \ldots, a^{m+r-1}\}$ *is a cyclic subgroup of* $\langle a \rangle$.

The number m in Theorem 28(2) is called the index of a, and r is called the period of a, in $(S, \circ)$. The following property holds true:

order(a)=index(a)+period(a)-1

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Free semigroups and monoids

Remark 4

- *The monoid* $(\mathbb{N}, +, 0)$ *can be generated by* $\{1\}$*. Moreover, any number* $n \in \mathbb{N} - \{0\}$ *can uniquely be written as a finite combination of* 1*'s under* $+$*, namely,*

$$n = \underbrace{1 + 1 + \cdots + 1}_{n \text{ times}}.$$

We say that $\{1\}$ *freely generates the monoid.*

Definition 29

A semigroup $(S, \circ)$ is freely generated by a subset $X \subseteq S$ if any element of $s \in S$ can uniquely be written as a finite combination of elements of X,

$$s = x_1 \circ \cdots \circ x_n,$$

where $x_1, \ldots, x_n \in X$ and $n \geq 1$.

Free semigroups and monoids

Definition 30

A monoid $(M, \circ, e)$ is freely generated by a subset $X \subseteq M$ if $(S_M, \circ)$ is freely generated by X.

Definition 31

A free semigrup (free monoid) is a semigroup (monoid) which can be freely generated by some subset of it.

If A freely generates a semigroup, then it is called a set of free generators of the semigroup (monoid).

Example 32

- $(\mathbb{N}, +, 0)$ is a free monoid.
- X^+ (X^*) together with the concatenation operation is a free semigroup (monoid), for any non-empty set X.
- $(\mathbb{Z}, +, 0)$ is not a free monoid.

Free semigroups and monoids

Free semigroups (monoids) have very important properties.

Theorem 33 (The universality property)

If $(S, \circ)$ is a semigroup freely generated by X, then for any semigroup $(T, *)$ and any function $f : X \to T$, there exists a unique homomorphism $h : S \to T$ which extends f (that is, $h(x) = f(x)$, for any $x \in X$).

The universality property can be similarly formulated for free monoids.

Corollary 34

Any free semigroup (monoid) is isomorphic with a word semigroup (monoid).

Free semigroups and monoids

The universality property allows us to define homomorphisms from free semigroups $(S, \circ)$ to semigroups $(T, *)$ just by defining them on sets of free generators of $(S, \circ)$.

Example 35

To define a homomorphism from $(\mathbb{N}, +, 0)$ to $(\mathbb{N}, \cdot, 1)$ it is sufficient to consider an arbitrary function from $\{1\}$, which freely generates $(\mathbb{N}, +, 0)$, to $(\mathbb{N}, \cdot, 1)$. For example, if we consider the function $f(1) = 10$, then the unique homomorphism induced by f is:

- $h(0) = 1$;
- $h(1) = f(1) = 10$;
- $h(2) = h(1 + 1) = h(1) \cdot h(1) = 10^2$;
- $h(3) = h(1 + 1 + 1) = h(1) \cdot h(1) \cdot h(1) = 10^3$;
- $h(n) = 10^n$, for any $n \geq 0$.

Free semigroups and monoids

How many sets of free generators may have a free semigroup or monoid ?

Proposition 1

If a semigroup $(S, \circ)$ (monoid $(M, \circ, e)$) is free, then it has a unique set of free generates, and this set is $S - S^2$ ($S_M - S_M^2$).

Proof.

First, show that any set of generators should include $S - S^2$ ($S_M - S_M^2$). Then, show that any set X of generators should be a subset of $S - S^2$ ($S_M - S_M^2$). $\square$

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Variable-length codes

Definition 36

Let A be a non-empty set. A variable-length code (or simply code) over A is any subset $C \subseteq A^+$ such that C^* is a free sub-monoid of A^*. The elements of C are called code words.

Equivalent definitions:

1. C is a code over A if any code sequence $w \in C^+$ can be uniquely decomposed into code words
$$w = c_1 \cdots c_n;$$
2. C is a code over A if
$$u_1 \cdots u_m = v_1 \cdots v_n \quad \Rightarrow \quad n = m \ \wedge \ (\forall i)(u_i = v_i),$$
for any $u_1, \ldots, u_m, v_1, \ldots, v_n \in C$;
3. C is a code over A if
$$u_1 \cdots u_m = v_1 \cdots v_n \quad \Rightarrow \quad u_1 = v_1,$$
for any $u_1, \ldots, u_m, v_1, \ldots, v_n \in C$.

Variable-length codes

Example 37

- $C = \{a, ab, ba\}$ is not a code because $aba = (ab)a = a(ba)$;
- $C = \{a, bb, aab, bab\}$ is a code.

Definition 38

1. C is a prefix code if no code word of C is a prefix of any other code word.
2. C is a suffix code if no code word of C is a suffix of any other code word.
3. C is a block code if all code words of C have the same length.

Example 39

- ASCII is a block code.

Variable-length codes

Given a non-empty set $C \subseteq A^+$, define

- $C_1 = \{x \in A^+ | (\exists c \in C)(cx \in C)\}$,
- $C_{i+1} = \{x \in A^+ | (\exists c \in C)(cx \in C_i) \ \vee \ (\exists c \in C_i)(cx \in C)\}$, for any $i \geq 1$.

We get an infinite sequence of sets of words:

$$C_1, C_2, C_3, \ldots$$

Remark 5

If C is finite, then there are i and j such that $j < i$ and $C_i = C_j$.

Theorem 40 (Sardinas-Patterson Theorem)

C is a code over A iff $C \cap C_i = \emptyset$, for any $i \geq 1$.

Variable-length codes

Sardinas-Patterson Algorithm

input: finite non-empty set $C \subseteq A^+$;
output: $code(C) = 1$, if C is a code, and $code(C) = 0$, otherwise;
begin
  $C_1 := \{x \in A^+ | (\exists c \in C)(cx \in C)\}$;
  if $C \cap C_1 \neq \emptyset$ **then** $code(C) := 0$
    else begin
      $i := 1$; $cont := 1$;
      while $cont = 1$ **do**
        begin
          $i := i + 1$;
          $C_i := \{x \in A^+ | (\exists c \in C_{i-1})(cx \in C) \vee (\exists c \in C)(cx \in C_{i-1})\}$;
          if $C \cap C_i \neq \emptyset$ **then begin** $code(C) := 0$; $cont := 0$ **end**
            else if $(\exists j < i)(C_i = C_j)$
              then begin $code(C) := 1$; $cont := 0$ **end**;
        end;
    end;
end.

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Huffman codes

- Have been proposed by David Huffman in 1952;
- Are used to encode information sources (an information source is a device which outputs symbols from a given alphabet according to certain probabilities depending, in general, on preceding choices as well as the particular symbol in question);
- Are prefix codes of minimum length among all the prefix codes associated to a given information source;
- Associate short code words to highly probable symbols (which appear more frequently), and longer code words to symbols with smaller probabilities.

Huffman codes

Definition 41

An information source is a couple $IS = (A, \pi)$, where A is a non-empty and at most countable set, called the source alphabet, and π is a probability distribution on A.

Only finite information sources will be considered.

Definition 42

Let $IS = (A, \pi)$ be an information source and $h : A \to \Sigma^*$ be a homomorphism. The (average) length of h with respect to IS is

$$L_h(IS) = \sum_{a \in A} |h(a)|\pi(a).$$

Huffman codes

Definition 43

Let $IS = (A, \pi)$ be an information source and $h : A \to \Sigma^*$ be a homomorphism. h is called a code or encoding of IS if $C = \{h(a) | a \in A\}$ is a code.

Example 44

Let IS be the information source

A	a	b	c	d	e	f
π	0.4	0.1	0.1	0.1	0.2	0.1

and h be the encoding

A	a	b	c	d	e	f
h	0	1100	1101	1110	10	1111

Then, the length of h w.r.t. IS is $L_h(IS) = 2.4$. That is, on average, 2.4 bits are needed to encode any symbol of the information source.

Huffman codes

Definition 45

Let $IS = (A, \pi)$ be an information source and $h : A \to \Sigma^*$ be an encoding for IS. h is called a Huffman code or a Huffman encoding of IS if:

- $C = \{h(a) | a \in A\}$ is a prefix code;
- h has minimum length among all the prefix codes of IS.

Given an information source IS, are there Huffman encodings for IS ?
The answer is positive.

Huffman codes

Huffman algorithm:

1. let IS be an information source with $n \geq 2$ symbols

A	a_1	a_2	$\cdots$	a_{n-1}	a_n
π	p_1	p_2	$\cdots$	p_{n-1}	p_n

   where $p_1 \geq p_2 \geq \cdots \geq p_{n-1} \geq p_n$;

2. if $n = 2$, then $h(a_1) = 0$ and $h(a_2) = 1$ is a Huffman code for IS;
3. if $n \geq 3$, then compute a reduced source IS' for IS

A'	a_1	a_2	$\cdots$	a_{n-2}	$a_{n-1,n}$
π'	p_1	p_2	$\cdots$	p_{n-2}	$p_{n-1,n}$

   where $p_{n-1,n} = p_{n-1} + p_n$;

Huffman codes

4. if h' is a Huffman code for IS', then h given by

$$h(x) = \begin{cases} h'(x), & \text{if } x \notin \{a_{n-1}, a_n\} \\ h'(x)0, & \text{if } x = a_{n-1} \\ h'(x)1, & \text{if } x = a_n, \end{cases}$$

is a Huffman code for IS.

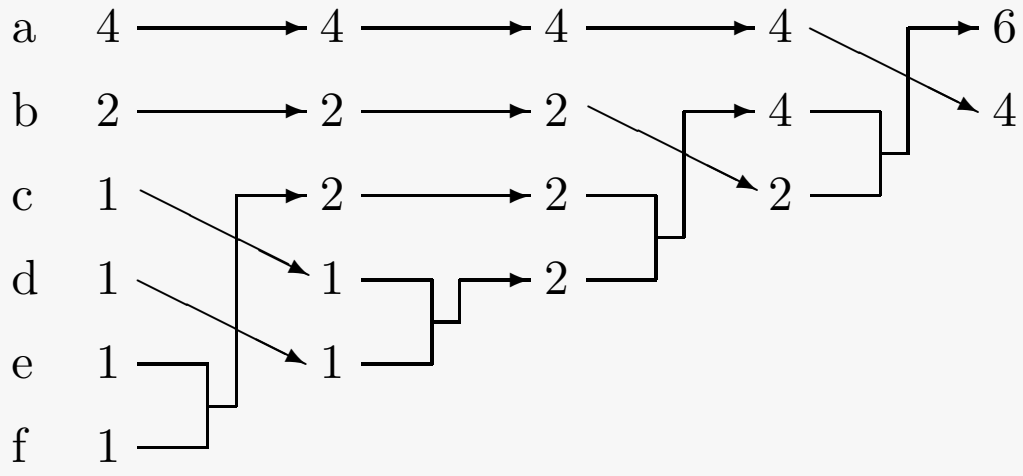
Huffman codes

Example 46

Let IS be the following information source:

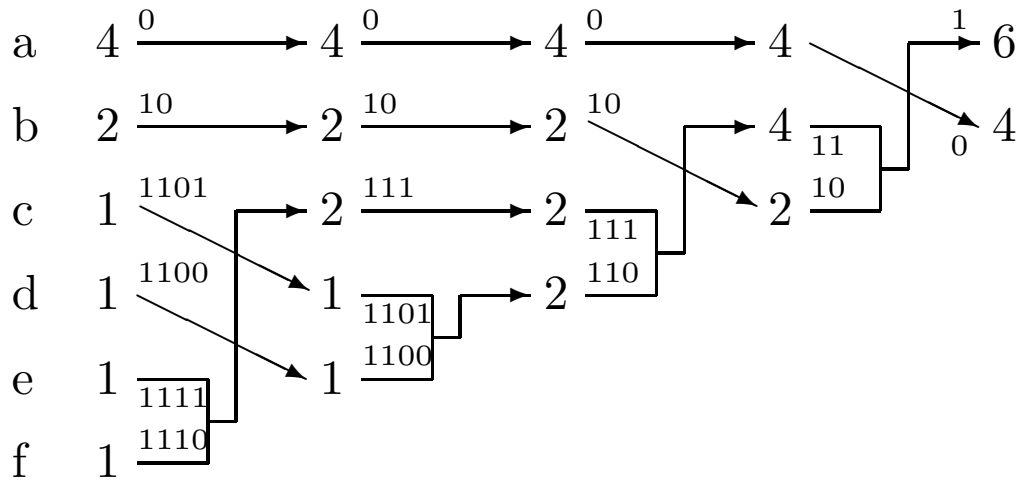
A	a	b	c	d	e	f
π	0.4	0.2	0.1	0.1	0.1	0.1

Compute a sequence of reduced sources for IS:



Huffman codes

Assign codes to each reduced source from right to left:



The Huffman code is $h(a) = 0$, $h(b) = 10$, $h(c) = 1101$, $h(d) = 1100$, $h(e) = 1111$, and $h(f) = 1110$. The length of h is 2.4. It is the minimum length code among all the prefix codes associated to IS.

Huffman codes

Huffman codes can be used to compress data as follows. Let α be a text:

1. parse α and, for each symbol a in α compute its number of occurrences;
2. let IS be the information source thus obtained. Compute a Huffman code h for IS;
3. encode α by $h(\alpha)$ (obtained by replacing each symbol a in α by $h(a)$).

Compression ratio = is the ratio of the size of the original data to the size of the compressed data.

Compression rate = is the rate of the compressed data (typically, it is in units of bits/sample, bits/character, bits/pixels, or bits/second).

Huffman codes

There are two types of data compression:

- lossless data compression – allows the exact original data to be reconstructed from the compressed data;
- lossy data compression – does not allow the exact original data to be reconstructed from the compressed data.

Data compression by Huffman codes is an example of lossless data compression.

Is there any limit to lossless data compression?

The answer is positive. The limit is called the entropy. The exact value of the entropy depends on the (statistical nature of the) information source. It is possible to compress the source, in a lossless manner, with compression rate close to its entropy. It is mathematically impossible to do better than that.

Huffman codes

Definition 47

Let S be an information source with n symbols and probabilities $p_1, \ldots, p_n$. The entropy of S, denoted $H(S)$ or $H(p_1, \ldots, p_n)$, is defined by

$$H(S) = \sum_{i=1}^{n} p_i \log(1/p_i)$$

(mathematical convention: $0 \cdot \log(1/0) = 0$).

Proposition 2

For any distribution of probability $p_1, \ldots, p_n$, the following properties hold:

1. $0 \leq H(p_1, \ldots, p_n) \leq \log n$;
2. $H(p_1, \ldots, p_n) = 0$ *iff* $p_i = 1$, *for some* i;
3. $H(p_1, \ldots, p_n) = \log n$ *iff* $p_i = 1/n$, *for any* i.

Huffman codes

Definition 48

Let $S_1 = (\{a_i | 1 \leq i \leq n\}, (p_i | 1 \leq i \leq n))$ and $S_2 = (\{b_j | 1 \leq j \leq m\}, (q_j | 1 \leq j \leq m))$ be two information sources. The product of S_1 and S_2, denoted $S_1 \circ S_2$, is the information source

$$S_1 \circ S_2 = (\{(a_i, a_j) | 1 \leq i \leq n,\ 1 \leq j \leq m\}, (p_i \cdot q_j | 1 \leq i \leq n,\ 1 \leq j \leq m)).$$

Proposition 3

For any finite information sources S_1 and S_2,

$$H(S_1 \circ S_2) = H(S_1) + H(S_2).$$

We denote $\underbrace{S \circ \cdots \circ S}_{k \text{ times}}$ by S^k, where S is a finite information source.

Then,

$$H(S^k) = kH(S).$$

Huffman codes

Theorem 49 (Shannon's noiseless coding theorem)

Let S be an information source. Then:

(1) $H(S) \leq L_h(S)$, *for any code h of S;*

(2) $H(S) \leq L_h(S) < H(S) + 1$, *for any Huffman code h of S;*

(3) $\lim_{k\to\infty} \frac{L_{min}(S^k)}{k} = H(S)$, *where $L_{min}(S')$ is the average length of some Huffman code for S'.*

Shannon's noiseless coding theorem places a lower bound on the minimal possible expected length of an encoding of a source S, as a function of the entropy of S.

Huffman codes

The design of a Huffman encoding for an input $w \in \Sigma^+$ requires two steps:

- determine the frequency of occurrences of each letter a in w;
- design a Huffman code for Σ w.r.t. the probability distribution

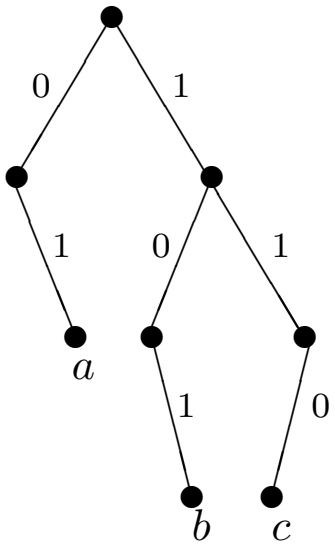
$$p_a = \frac{\text{frequency of } a \text{ in } w}{|w|}.$$

  Then, encode w by this Huffman code.

Because this procedure requires two parsings of the input, it is time-consuming for large inputs (although the compression rate by such an encoding is optimal). In practice, an alternative method which requires only one parsing of the input is used. It is called the adaptive Huffman encoding.

Huffman codes

A useful graphical representation of a finite code $C \subseteq A^+$ consists of a tree with nodes labeled by symbols in A such that the code words are exactly the sequences of labels collected from the root to leaves. For example, the tree below is the graphical representation of the prefix code $\{01, 110, 101\}$, where a is encoded by 01, b by 101, and c by 110.



Huffman codes

The encoding of an input w by the adaptive Huffman technique is based on the construction of a sequence of Huffman trees as follows:

- start initially with a Huffman tree $\mathcal{T}_0$ associated to the alphabet A (each symbol of A has frequency 1);
- if $\mathcal{T}_n$ is the current Huffman tree and the current input symbol is a (that is, $w = uav$ and u has been already processed), then output the code of a in $\mathcal{T}_n$ (this code is denoted by $code(a, \mathcal{T}_n)$) and update the tree $\mathcal{T}_n$ by applying to it the sibling transformation; the new tree is denoted $\mathcal{T}_{n+1}$.

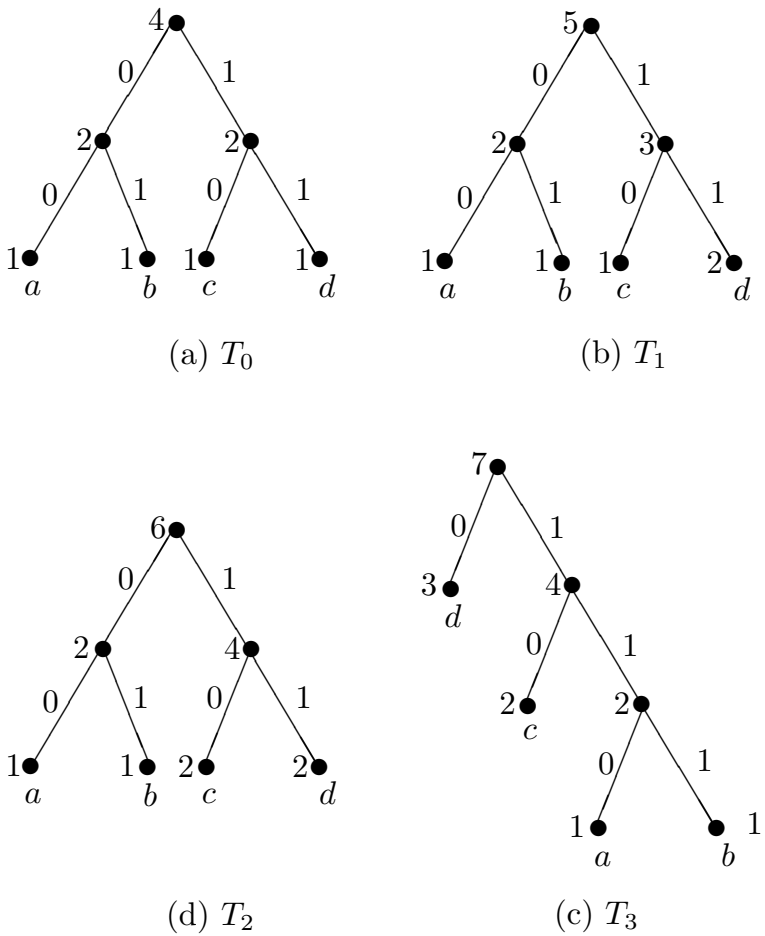
Huffman codes

The sibling transformation applied to symbol a and tree $\mathcal{T}_n$ consists of:

1. compare a to its successors in the tree (from left to right and from bottom to top). If the immediate successor has frequency $k+1$ or greater, where k is the frequency of a in $\mathcal{T}_n$, then the nodes are still in sorted order and there is no need to change anything. Otherwise, a should be swapped with the last successor which has frequency k or smaller (except that a should not be swapped with its parent);
2. increment the frequency of a (from k to $k+1$);
3. if a is the root, the loop halts; otherwise, the loop repeats with the parent of a.

Huffman codes

A sequence of Huffman trees used to encode the string *dcd* over the alphabet $\{a, b, c, d\}$:



(a) T_0 (b) T_1

(d) T_2 (c) T_3

Huffman codes

Huffman adaptive is not a variable-length code! The same character may be encoded by different code words!

Huffman adaptive is a time-varying code! For more details regarding time-varying codes see TMTE2002.pdf in the course web site.

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Course readings

1. F.L. Ţiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iaşi, 2006, pag. 179–243.

2. F.L. Ţiplea, E. Mäkinen, D. Trincă, C. Enea: *Characterization Results for Time-varying Codes*, Fundamenta Informaticae 53(2), 2002, 185-198.

Algebraic Foundations of Computer Science

Semigroups and Monoids. Variable-length Codes. Huffman Codes

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Definitions and examples

Definition 1

A semigroup is a pair $(S, \circ)$ which consists of a set S and an associative binary operation $\circ$ on S.

Example 2

1. $(\mathbb{N}, +)$, $(\mathbb{Z}, +)$, $(\mathbb{Q}, +)$, $(\mathbb{R}, +)$, and $(\mathbb{C}, +)$ are (additive) semigroups;
2. $(\mathbb{N}, \cdot)$, $(\mathbb{Z}, \cdot)$, $(\mathbb{Q}, \cdot)$, $(\mathbb{R}, \cdot)$, and $(\mathbb{C}, \cdot)$ are (multiplicative) semigroups;
3. Let $n \in \mathbb{Z}$ and $n\mathbb{Z} = \{n \cdot x | x \in \mathbb{Z}\}$. Then, $(n\mathbb{Z}, +)$ and $(n\mathbb{Z}, \cdot)$ are semigroups;
4. Let $m \in \mathbb{Z}$. Then, $(\mathbb{Z}_m, +)$ and $(\mathbb{Z}_m, \cdot)$, where $+$ and $\cdot$ are the addition and multiplication modulo m, are semigroups.

Definitions and examples

Definition 3

A semigroup $(S, \circ)$ is called commutative if $\circ$ is a commutative operation.

Example 4

All semigroups in Example 2 are commutative.

Remark 1

- *Associativity of a binary operation $\circ$ means that the order of evaluation of an expression $a_1 \circ a_2 \circ a_3$, without changing the order of the terms, is immaterial. In other words, no parenthesis is required for an associative operation;*
- *Commutativity of a binary operation $\circ$ means that the order of the operands in expressions like $a_1 \circ a_2$ is immaterial.*

Definitions and examples

Definition 5

A monoid is a triple $(M, \circ, e)$ which consists of a set M, an associative binary operation $\circ$ on M, and an element $e \in M$ such that

$$x \circ e = e \circ x = x,$$

for any $x \in M$. e is called the identity element or the unity of M.

Remark 2

The identity of any monoid $(M, \circ, e)$ is unique. For, if we assume that e' is an identity too, then:

$$e = e \circ e' = e'.$$

The identity of a monoid $(M, \circ, e)$ is usually denoted by 1_M or even 1.

Definitions and examples

Definition 6

A monoid $(M, \circ, e)$ is called commutative if its binary operation $\circ$ is commutative.

Example 7

1. $(\mathbb{N}, +, 0)$, $(\mathbb{Z}, +, 0)$, $(\mathbb{Q}, +, 0)$, and $(\mathbb{R}, +, 0)$ are commutative monoids;
2. $(\mathbb{N}, \cdot, 1)$, $(\mathbb{Z}, \cdot, 1)$, $(\mathbb{Q}, \cdot, 1)$, and $(\mathbb{R}, \cdot, 1)$ are commutative monoids;
3. $(n\mathbb{Z}, +, 0)$ is a commutative monoid and $(n\mathbb{Z}, \cdot)$ is a commutative semigroup. $(n\mathbb{Z}, \cdot)$ has unity only if $n = 0$ or $n = 1$ and, in such a case it becomes commutative monoid;
4. $(\mathbb{Z}_m, +, 0)$ and $(\mathbb{Z}_m, \cdot, 1)$ are commutative monoids. When $m = 1$, $\mathbb{Z}_1 = \{0\}$ and the multiplicative unity of this monoid is 0.

1. Definitions and examples

A few basic notations:

1. Let $(S, \circ)$ be a semigroup, $A, B \subseteq S$, and $a \in S$. Define:
   - $AB = \{a \circ b | a \in A \ \wedge \ b \in B\}$;
   - $A^1 = A$ and $A^{n+1} = A^n A$, for all $n \geq 1$;
   - $aB = \{a \circ b | b \in B\}$;
   - $a^1 = a$ and $a^{n+1} = a^n \circ a$, for all $n \geq 1$.
2. If $(M, \circ, e)$ is a monoid, $A \subseteq M$, and $a \in M$, we also define:
   - $A^0 = \{e\}$;
   - $a^0 = e$.
3. For any monoid $(M, \circ, e)$ define $S_M = M - \{e\}$.

Definitions and examples

Definition 8

Let $(S, \circ)$ be a semigroup and I a non-empty subset of S.

1. I is called a left ideal of $(S, \circ)$ if $SI \subseteq I$.
2. I is called a right ideal of $(S, \circ)$ if $IS \subseteq I$.
3. I is called an ideal of $(S, \circ)$ if I is a left and a right ideal of $(S, \circ)$.
4. The least (left, right) ideal of $(S, \circ)$ which includes I is called the (left, right) ideal of $(S, \circ)$ generated by I. It is denoted by $\langle I \rangle$.
5. If $I = \{a\}$, then $\langle I \rangle$ is called a (left, right) principal ideal of $(S, \circ)$. It is also denoted by $\langle a \rangle$.

(Left, Right) Ideals of monoids are defined in a similar way.

Example 9

The principal ideal of $(\mathbb{Z}, \cdot, 1)$ generated by $n \in \mathbb{Z}$ is $n\mathbb{Z}$.

Definitions and examples

Definition 10

1. A semigroup $(S', \circ')$ is a sub-semigroup of a semigroup $(S, \circ)$, denoted $(S', \circ') \leq (S, \circ)$, if $S' \subseteq S$ and $\circ' = \circ|_{S'}$.
2. A monoid $(M', \circ', e')$ is a sub-monoid of a monoid $(M, \circ, e)$, denoted $(M', \circ', e') \leq (M, \circ, e)$, if $M' \subseteq M$ and $\circ' = \circ|_{M'}$ and $e' = e$.
3. The least subsemigroup (monoid) of a semigroup (monoid) which includes a given subset A, denoted $\langle A \rangle$, is called the sub-semigroup (sub-monoid) generated by A.
4. A semigroup (monoid) is generated by a subset A of it if it coincides with the sub-semigroup (sub-monoid) generated by A.

The set A in Definition 10(3)(4) is called a set of generators and its elements are called generators.

Definitions and examples

Remark 3

The sub-semigroup (sub-monoid) of a semigroup (monoid), generated by a subset A, is the closure of A under the operation(s) of the host semigroup (monoid):

- *If $(S, \circ)$ is a semigroup and $A \subseteq S$, then the sub-semigroup generated by A is the set of all products*

$$a_1 \circ \cdots \circ a_n,$$

  where $n \geq 1$ and $a_1, \ldots, a_n \in A$;

- *If $(M, \circ, e)$ is a monoid and $A \subseteq S$, then the sub-monoid generated by A is the set of all products*

$$a_1 \circ \cdots \circ a_n,$$

  where $n \geq 1$ and $a_1, \ldots, a_n \in A$, together with the unity e of the host monoid.

Definitions and examples

Example 11

- $(\mathbb{N}, +) \leq (\mathbb{Z}, +) \leq (\mathbb{Q}, +) \leq (\mathbb{R}, +)$;
- $(\mathbb{N}, +, 0) \leq (\mathbb{Z}, +, 0) \leq (\mathbb{Q}, +, 0) \leq (\mathbb{R}, +, 0)$;
- $(\mathbb{N}, \cdot) \leq (\mathbb{Z}, \cdot) \leq (\mathbb{Q}, \cdot) \leq (\mathbb{R}, \cdot)$;
- $(\mathbb{N}, \cdot, 1) \leq (\mathbb{Z}, \cdot, 1) \leq (\mathbb{Q}, \cdot, 1) \leq (\mathbb{R}, \cdot, 1)$;
- The sub-monoid of $(\mathbb{Z}, +, 0)$, generated by $n \in \mathbb{Z}$, is $(n\mathbb{N}, +, 0)$;
- A semigroup (monoid) may have more than one set of generators. For instance, $(\mathbb{Z}, +, 0)$ can be generated by $\{-1, 1\}$ and by $\{-3, 2\}$.

Definitions and examples

Definition 12

1. The order of a semigroup (monoid) is the number of its elements if the semigroup (monoid) is finite, and ∞, otherwise.
2. The order of an element a of a semigroup (monoid) is the order of the sub-semigroup (sub-monoid) generated by a.

Example 13

- $(\mathbb{Z}, +, 0)$ has the order ∞;
- $(\mathbb{Z}_m, +, 0)$ has the order m, if $m \neq 0$. For $m = 0$, $(\mathbb{Z}_m, +, 0)$ has the order ∞.

Definitions and examples

Definition 14

1. A function $f : S \to S'$ is a homomorphism from a semigroup $(S, \circ)$ to a semigroup $(S', \circ')$ if
   - $f(a \circ b) = f(a) \circ' f(b)$, for any $a, b \in S$.
2. A function $f : M \to M'$ is a homomorphism from a monoid $(M, \circ, e)$ to a monoid $(M', \circ', e')$ if
   - $f(a \circ b) = f(a) \circ' f(b)$, for any $a, b \in M$;
   - $f(e) = e'$.

Related concepts:

- injective homomorphism = monomorphism;
- surjective homomorphism = epimorphism;
- bijective homomorphism = isomorphism;

Definitions and examples

- Homomorphism from a semigroup (monoid) to the same semigroup (monoid) = endomorphism;
- Isomorphism from a semigroup (monoid) to the same semigroup (monoid) = automorphism.

Example 15

- The function $f(x) = 2^x$, for any $x \in \mathbb{N}$, is a homomorphism from $(\mathbb{N}, +, 0)$ to $(\mathbb{N}, \cdot, 1)$. Indeed,
  - $f(0) = 2^0 = 1$;
  - $f(x + y) = 2^{x+y} = 2^x \cdot 2^y = f(x) \cdot f(y)$, for any x, y.

  Moreover, f is injective but not surjective. Therefore, f is a monomorphism (but not an epimorphism).

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Word semigroups

Definition 16

An alphabet is any non-empty set. The elements of an alphabet are called letters or symbols.

Example 17

The following sets are alphabets:

- $\Sigma_1 = \{a, b, c\}$;
- $\Sigma_2 = \{0, 1, 2, 3\}$;
- $\Sigma_3 = \{\texttt{begin}, \texttt{end}, \texttt{if}, \texttt{then}, \texttt{else}, \texttt{while}, \texttt{do}\}$.

All letters of an alphabet are assumed indivisible.

Word semigroups

Definition 18

Let Σ be an alphabet. A word of length $k \geq 1$ over Σ is any function $w : \{1, \ldots, k\} \to \Sigma$. The empty function from $\emptyset$ into Σ is called the empty word over Σ and its length is 0.

We usually denote the word w by $w = w(1) \cdots w(k)$, if $k > 0$, and its length k by $|w|$. The empty word is usually denoted by λ.

Example 19

- $w = abaa$ is a word of length 4 over $\Sigma_1 = \{a, b, c\}$;
- 011033 is a word of length 6 over $\Sigma_2 = \{0, 1, 2, 3\}$;
- `begin end` is a word of length 2 over $\Sigma_3 = \{\texttt{begin}, \texttt{end}, \texttt{if}, \texttt{then}, \texttt{else}, \texttt{while}, \texttt{do}\}$.

Word semigroups

Let Σ be an alphabet. Denote:

- $\Sigma^0 = \{\lambda\}$;
- $\Sigma^+ = \bigcup_{k \geq 1} \Sigma^k$,
- $\Sigma^* = \bigcup_{k \geq 0} \Sigma^k = \Sigma^+ \cup \{\lambda\}$.

Words of length 1 are usually identified with letters. Therefore, we may write $\Sigma^1 = \Sigma$.

Definition 20

Two words u and v over the same alphabet Σ are called equal if they have the same length k and $u(i) = v(i)$, for each $1 \leq i \leq k$.

Word semigroups

Definition 21

Let Σ be an alphabet. The binary operation $\cdot : \Sigma^* \times \Sigma^* \to \Sigma^*$ given by

$$w_1 \cdot w_2 : \{i | 1 \leq i \leq |w_1| + |w_2|\} \to \Sigma$$

where

$$(w_1 \cdot w_2)(i) = \begin{cases} w_1(i), & \text{if } 1 \leq i \leq |w_1| \\ w_2(i - |w_1|), & \text{otherwise,} \end{cases}$$

for any i, is called the concatenation or catenation operation on Σ^*.

Example 22

- $abba \cdot bbaa = abbabbaa$;
- $\lambda \cdot w = w \cdot \lambda = w$, for any w.

The concatenation operation symbol is usually omitted. That is, we write uv instead of $u \cdot v$.

Word semigroups

Theorem 23

Let Σ be an alphabet. Then:

(1) $(\Sigma^+, \cdot)$ *is a semigroup generated by* Σ;

(2) $(\Sigma^*, \cdot, \lambda)$ *is a monoid generated by* Σ;

(3) $(\Sigma^*, \cdot, \lambda)$ *is a monoid with simplification;*

(4) $l : \Sigma^* \to \mathbb{N}$ *given by* $l(w) = |w|$, *for any* $w \in \Sigma^*$, *is a homomorphism from* $(\Sigma^*, \cdot, \lambda)$ *to the additive monoid* $(\mathbb{N}, +, 0)$. *Moreover,* $l^{-1}(0) = \{\lambda\}$;

(5) *The group of units of the monoid* $(\Sigma^*, \cdot, \lambda)$ *is trivial.*

Word semigroups

Theorem 24 (Levi's Theorem)

Let x, y, u, and v be words over Σ such that $xy = uv$.

(1) If $|x| < |u|$, then there exists a unique $z \in \Sigma^*$ such that $u = xz$.

(2) If $|x| = |u|$, then $x = u$ and $y = v$.

(3) If $|x| > |u|$, then there exists a unique $z \in \Sigma^*$ such that $x = uz$.

Definition 25

Let Σ be an alphabet and $u, v \in \Sigma^*$.

(1) u is called a prefix or left factor of v if $v = uw$ for some word w.

(2) u is called a suffix or right factor of v if $v = wu$ for some word w.

(3) u is called a sub-word of v if $v = xuy$ for some words x and y.

Word semigroups

Definition 26

(1) A pair $(\Sigma, \prec)$ which consists of an alphabet Σ and a total order $\prec$ on Σ is called an ordered alphabet.

(2) Let $(\Sigma, \prec)$ be an ordered alphabet. The binary relation $\leq_{(\Sigma,\prec)}$ given by

$$x \leq_{(\Sigma,\prec)} y$$

iff

- x is a prefix of y, or
- $x = uav$, $y = ubw$, and $a \prec b$, for some $u, v, w \in \Sigma^*$ and $a, b \in \Sigma$ with $a \neq b$,

is called the direct lexicographic order on $(\Sigma, \prec)$.

In a similar way one can define the inverse lexicographic order on ordered alphabets.

Word semigroups

λ	λ
a	a
aa	b
aaa	aa
$\dots$	ab
$aaaab$	ba
$aaab$	bb
aab	aaa
ab	aab
$\dots$	$\dots$
b	bbb
a)	b)

a) Lexicographic order
b) Lexicographic order on words of the same length.

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Cyclic semigroups

If $\mathbf{S} = (S, \circ)$ is a semigroup and $a \in S$, then

$$\langle a \rangle_{\mathbf{S}} = \{a, a^2, \ldots, a^n, \ldots\}$$

If $\mathbf{M} = (M, \circ, e)$ is a monoid $a \in M$, then

$$\langle a \rangle_{\mathbf{M}} = \{e = a^0, a, a^2, \ldots, a^n, \ldots\}$$

Definition 27

A semigroup (monoid) generated by one of its elements is called a cyclic semigroup (cyclic monoid).

$\mathbf{S}$ cyclic semigroup $\Rightarrow S = \{a, a^2, \ldots, a^n, \ldots\}$, for some $a \in S$.

$\mathbf{M}$ cyclic monoid $\Rightarrow M = \{e = a^0, a, a^2, \ldots, a^n, \ldots\}$, for some $a \in M$.

Cyclic semigroups

Theorem 28

Let a be an element of a semigroup $(S, \circ)$. Then, exactly one of the following two properties is satisfied:

(1) $a^n \neq a^m$ *for any* $n \neq m$, *and the semigroup generated by* a *is isomorphic with* $(\mathbb{N} - \{0\}, +)$;

(2) *there exists* $m > 0$ *and* $r > 0$ *such that :*

  (a) $a^m = a^{m+r}$;

  (b) $a^{m+u} = a^{m+v}$ *iff* $u \equiv v \mod r$, *for any* $u, v \in \mathbb{N}$;

  (c) $\langle a \rangle = \{a, a^2, \ldots, a^{m+r-1}\}$ *has exactly* $m + r - 1$ *elements;*

  (d) $K(a) = \{a^m, \ldots, a^{m+r-1}\}$ *is a cyclic subgroup of* $\langle a \rangle$.

The number m in Theorem 28(2) is called the index of a, and r is called the period of a, in $(S, \circ)$. The following property holds true:

order(a)=index(a)+period(a)-1

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Free semigroups and monoids

Remark 4

- *The monoid* $(\mathbb{N}, +, 0)$ *can be generated by* $\{1\}$*. Moreover, any number* $n \in \mathbb{N} - \{0\}$ *can uniquely be written as a finite combination of* 1*'s under* $+$*, namely,*

$$n = \underbrace{1 + 1 + \cdots + 1}_{n \text{ times}}.$$

We say that $\{1\}$ *freely generates the monoid.*

Definition 29

A semigroup $(S, \circ)$ is freely generated by a subset $X \subseteq S$ if any element of $s \in S$ can uniquely be written as a finite combination of elements of X,

$$s = x_1 \circ \cdots \circ x_n,$$

where $x_1, \ldots, x_n \in X$ and $n \geq 1$.

Free semigroups and monoids

Definition 30

A monoid $(M, \circ, e)$ is freely generated by a subset $X \subseteq M$ if $(S_M, \circ)$ is freely generated by X.

Definition 31

A free semigrup (free monoid) is a semigroup (monoid) which can be freely generated by some subset of it.

If A freely generates a semigroup, then it is called a set of free generators of the semigroup (monoid).

Example 32

- $(\mathbb{N}, +, 0)$ is a free monoid.
- X^+ (X^*) together with the concatenation operation is a free semigroup (monoid), for any non-empty set X.
- $(\mathbb{Z}, +, 0)$ is not a free monoid.

Free semigroups and monoids

Free semigroups (monoids) have very important properties.

Theorem 33 (The universality property)

If $(S, \circ)$ is a semigroup freely generated by X, then for any semigroup $(T, *)$ and any function $f : X \to T$, there exists a unique homomorphism $h : S \to T$ which extends f (that is, $h(x) = f(x)$, for any $x \in X$).

The universality property can be similarly formulated for free monoids.

Corollary 34

Any free semigroup (monoid) is isomorphic with a word semigroup (monoid).

Free semigroups and monoids

The universality property allows us to define homomorphisms from free semigroups $(S, \circ)$ to semigroups $(T, *)$ just by defining them on sets of free generators of $(S, \circ)$.

Example 35

To define a homomorphism from $(\mathbb{N}, +, 0)$ to $(\mathbb{N}, \cdot, 1)$ it is sufficient to consider an arbitrary function from $\{1\}$, which freely generates $(\mathbb{N}, +, 0)$, to $(\mathbb{N}, \cdot, 1)$. For example, if we consider the function $f(1) = 10$, then the unique homomorphism induced by f is:

- $h(0) = 1$;
- $h(1) = f(1) = 10$;
- $h(2) = h(1 + 1) = h(1) \cdot h(1) = 10^2$;
- $h(3) = h(1 + 1 + 1) = h(1) \cdot h(1) \cdot h(1) = 10^3$;
- $h(n) = 10^n$, for any $n \geq 0$.

Free semigroups and monoids

How many sets of free generators may have a free semigroup or monoid ?

Proposition 1

If a semigroup $(S, \circ)$ (monoid $(M, \circ, e)$) is free, then it has a unique set of free generates, and this set is $S - S^2$ ($S_M - S_M^2$).

Proof.

First, show that any set of generators should include $S - S^2$ ($S_M - S_M^2$). Then, show that any set X of generators should be a subset of $S - S^2$ ($S_M - S_M^2$). $\square$

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Variable-length codes

Definition 36

Let A be a non-empty set. A variable-length code (or simply code) over A is any subset $C \subseteq A^+$ such that C^* is a free sub-monoid of A^*. The elements of C are called code words.

Equivalent definitions:

1. C is a code over A if any code sequence $w \in C^+$ can be uniquely decomposed into code words
$$w = c_1 \cdots c_n;$$
2. C is a code over A if
$$u_1 \cdots u_m = v_1 \cdots v_n \quad \Rightarrow \quad n = m \ \wedge \ (\forall i)(u_i = v_i),$$
for any $u_1, \ldots, u_m, v_1, \ldots, v_n \in C$;
3. C is a code over A if
$$u_1 \cdots u_m = v_1 \cdots v_n \quad \Rightarrow \quad u_1 = v_1,$$
for any $u_1, \ldots, u_m, v_1, \ldots, v_n \in C$.

Variable-length codes

Example 37

- $C = \{a, ab, ba\}$ is not a code because $aba = (ab)a = a(ba)$;
- $C = \{a, bb, aab, bab\}$ is a code.

Definition 38

1. C is a prefix code if no code word of C is a prefix of any other code word.
2. C is a suffix code if no code word of C is a suffix of any other code word.
3. C is a block code if all code words of C have the same length.

Example 39

- ASCII is a block code.

Variable-length codes

Given a non-empty set $C \subseteq A^+$, define

- $C_1 = \{x \in A^+ | (\exists c \in C)(cx \in C)\}$,
- $C_{i+1} = \{x \in A^+ | (\exists c \in C)(cx \in C_i) \ \vee \ (\exists c \in C_i)(cx \in C)\}$, for any $i \geq 1$.

We get an infinite sequence of sets of words:

$$C_1, C_2, C_3, \ldots$$

Remark 5

If C is finite, then there are i and j such that $j < i$ and $C_i = C_j$.

Theorem 40 (Sardinas-Patterson Theorem)

C is a code over A iff $C \cap C_i = \emptyset$, for any $i \geq 1$.

Variable-length codes

Sardinas-Patterson Algorithm

input: finite non-empty set $C \subseteq A^+$;
output: $code(C) = 1$, if C is a code, and $code(C) = 0$, otherwise;
begin
  $C_1 := \{x \in A^+ | (\exists c \in C)(cx \in C)\}$;
  if $C \cap C_1 \neq \emptyset$ **then** $code(C) := 0$
    else begin
      $i := 1$; $cont := 1$;
      while $cont = 1$ **do**
        begin
          $i := i + 1$;
          $C_i := \{x \in A^+ | (\exists c \in C_{i-1})(cx \in C) \vee (\exists c \in C)(cx \in C_{i-1})\}$;
          if $C \cap C_i \neq \emptyset$ **then begin** $code(C) := 0$; $cont := 0$ **end**
            else if $(\exists j < i)(C_i = C_j)$
              then begin $code(C) := 1$; $cont := 0$ **end**;
        end;
    end;
end.

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Huffman codes

- Have been proposed by David Huffman in 1952;
- Are used to encode information sources (an information source is a device which outputs symbols from a given alphabet according to certain probabilities depending, in general, on preceding choices as well as the particular symbol in question);
- Are prefix codes of minimum length among all the prefix codes associated to a given information source;
- Associate short code words to highly probable symbols (which appear more frequently), and longer code words to symbols with smaller probabilities.

Huffman codes

Definition 41

An information source is a couple $IS = (A, \pi)$, where A is a non-empty and at most countable set, called the source alphabet, and π is a probability distribution on A.

Only finite information sources will be considered.

Definition 42

Let $IS = (A, \pi)$ be an information source and $h : A \to \Sigma^*$ be a homomorphism. The (average) length of h with respect to IS is

$$L_h(IS) = \sum_{a \in A} |h(a)|\pi(a).$$

Huffman codes

Definition 43

Let $IS = (A, \pi)$ be an information source and $h : A \to \Sigma^*$ be a homomorphism. h is called a code or encoding of IS if $C = \{h(a) | a \in A\}$ is a code.

Example 44

Let IS be the information source

A	a	b	c	d	e	f
π	0.4	0.1	0.1	0.1	0.2	0.1

and h be the encoding

A	a	b	c	d	e	f
h	0	1100	1101	1110	10	1111

Then, the length of h w.r.t. IS is $L_h(IS) = 2.4$. That is, on average, 2.4 bits are needed to encode any symbol of the information source.

Huffman codes

Definition 45

Let $IS = (A, \pi)$ be an information source and $h : A \to \Sigma^*$ be an encoding for IS. h is called a Huffman code or a Huffman encoding of IS if:

- $C = \{h(a) | a \in A\}$ is a prefix code;
- h has minimum length among all the prefix codes of IS.

Given an information source IS, are there Huffman encodings for IS ?
The answer is positive.

Huffman codes

Huffman algorithm:

1. let IS be an information source with $n \geq 2$ symbols

A	a_1	a_2	$\cdots$	a_{n-1}	a_n
π	p_1	p_2	$\cdots$	p_{n-1}	p_n

   where $p_1 \geq p_2 \geq \cdots \geq p_{n-1} \geq p_n$;

2. if $n = 2$, then $h(a_1) = 0$ and $h(a_2) = 1$ is a Huffman code for IS;
3. if $n \geq 3$, then compute a reduced source IS' for IS

A'	a_1	a_2	$\cdots$	a_{n-2}	$a_{n-1,n}$
π'	p_1	p_2	$\cdots$	p_{n-2}	$p_{n-1,n}$

   where $p_{n-1,n} = p_{n-1} + p_n$;

Huffman codes

4. if h' is a Huffman code for IS', then h given by

$$h(x) = \begin{cases} h'(x), & \text{if } x \notin \{a_{n-1}, a_n\} \\ h'(x)0, & \text{if } x = a_{n-1} \\ h'(x)1, & \text{if } x = a_n, \end{cases}$$

is a Huffman code for IS.

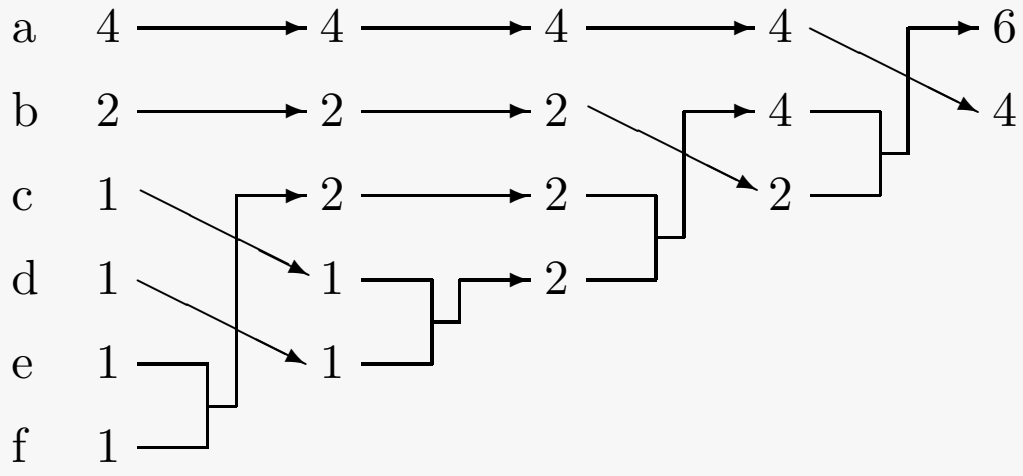
Huffman codes

Example 46

Let IS be the following information source:

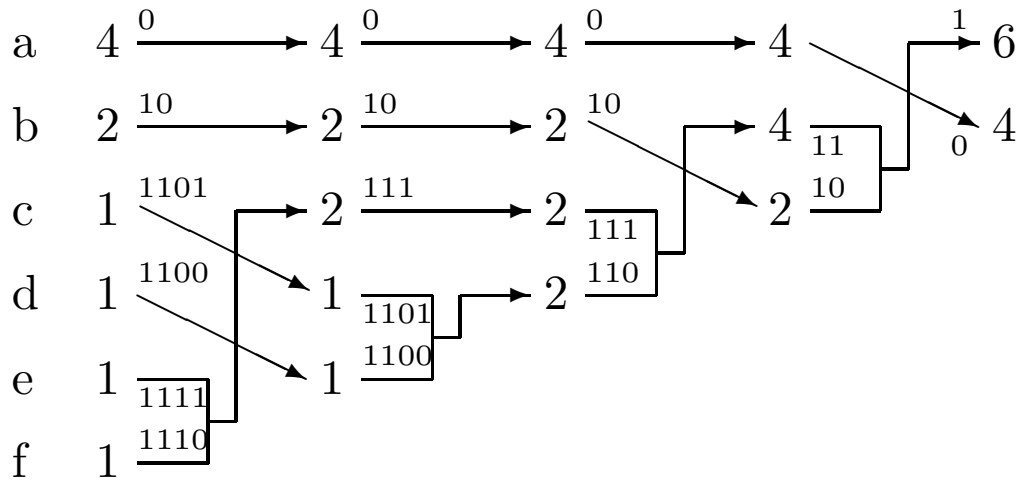
A	a	b	c	d	e	f
π	0.4	0.2	0.1	0.1	0.1	0.1

Compute a sequence of reduced sources for IS:



Huffman codes

Assign codes to each reduced source from right to left:



The Huffman code is $h(a) = 0$, $h(b) = 10$, $h(c) = 1101$, $h(d) = 1100$, $h(e) = 1111$, and $h(f) = 1110$. The length of h is 2.4. It is the minimum length code among all the prefix codes associated to IS.

Huffman codes

Huffman codes can be used to compress data as follows. Let α be a text:

1. parse α and, for each symbol a in α compute its number of occurrences;
2. let IS be the information source thus obtained. Compute a Huffman code h for IS;
3. encode α by $h(\alpha)$ (obtained by replacing each symbol a in α by $h(a)$).

Compression ratio = is the ratio of the size of the original data to the size of the compressed data.

Compression rate = is the rate of the compressed data (typically, it is in units of bits/sample, bits/character, bits/pixels, or bits/second).

Huffman codes

There are two types of data compression:

- lossless data compression – allows the exact original data to be reconstructed from the compressed data;
- lossy data compression – does not allow the exact original data to be reconstructed from the compressed data.

Data compression by Huffman codes is an example of lossless data compression.

Is there any limit to lossless data compression?

The answer is positive. The limit is called the entropy. The exact value of the entropy depends on the (statistical nature of the) information source. It is possible to compress the source, in a lossless manner, with compression rate close to its entropy. It is mathematically impossible to do better than that.

Huffman codes

Definition 47

Let S be an information source with n symbols and probabilities $p_1, \ldots, p_n$. The entropy of S, denoted $H(S)$ or $H(p_1, \ldots, p_n)$, is defined by

$$H(S) = \sum_{i=1}^{n} p_i \log(1/p_i)$$

(mathematical convention: $0 \cdot \log(1/0) = 0$).

Proposition 2

For any distribution of probability $p_1, \ldots, p_n$, the following properties hold:

1. $0 \leq H(p_1, \ldots, p_n) \leq \log n$;
2. $H(p_1, \ldots, p_n) = 0$ *iff* $p_i = 1$, *for some* i;
3. $H(p_1, \ldots, p_n) = \log n$ *iff* $p_i = 1/n$, *for any* i.

Huffman codes

Definition 48

Let $S_1 = (\{a_i | 1 \leq i \leq n\}, (p_i | 1 \leq i \leq n))$ and $S_2 = (\{b_j | 1 \leq j \leq m\}, (q_j | 1 \leq j \leq m))$ be two information sources. The product of S_1 and S_2, denoted $S_1 \circ S_2$, is the information source

$$S_1 \circ S_2 = (\{(a_i, a_j) | 1 \leq i \leq n,\ 1 \leq j \leq m\}, (p_i \cdot q_j | 1 \leq i \leq n,\ 1 \leq j \leq m)).$$

Proposition 3

For any finite information sources S_1 and S_2,

$$H(S_1 \circ S_2) = H(S_1) + H(S_2).$$

We denote $\underbrace{S \circ \cdots \circ S}_{k \text{ times}}$ by S^k, where S is a finite information source.

Then,

$$H(S^k) = kH(S).$$

Huffman codes

Theorem 49 (Shannon's noiseless coding theorem)

Let S be an information source. Then:

(1) $H(S) \leq L_h(S)$, *for any code h of S;*

(2) $H(S) \leq L_h(S) < H(S) + 1$, *for any Huffman code h of S;*

(3) $\lim_{k\to\infty} \frac{L_{min}(S^k)}{k} = H(S)$, *where $L_{min}(S')$ is the average length of some Huffman code for S'.*

Shannon's noiseless coding theorem places a lower bound on the minimal possible expected length of an encoding of a source S, as a function of the entropy of S.

Huffman codes

The design of a Huffman encoding for an input $w \in \Sigma^+$ requires two steps:

- determine the frequency of occurrences of each letter a in w;
- design a Huffman code for Σ w.r.t. the probability distribution

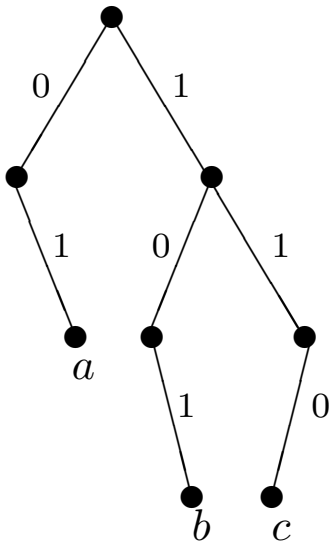
$$p_a = \frac{\text{frequency of } a \text{ in } w}{|w|}.$$

  Then, encode w by this Huffman code.

Because this procedure requires two parsings of the input, it is time-consuming for large inputs (although the compression rate by such an encoding is optimal). In practice, an alternative method which requires only one parsing of the input is used. It is called the adaptive Huffman encoding.

Huffman codes

A useful graphical representation of a finite code $C \subseteq A^+$ consists of a tree with nodes labeled by symbols in A such that the code words are exactly the sequences of labels collected from the root to leaves. For example, the tree below is the graphical representation of the prefix code $\{01, 110, 101\}$, where a is encoded by 01, b by 101, and c by 110.



Huffman codes

The encoding of an input w by the adaptive Huffman technique is based on the construction of a sequence of Huffman trees as follows:

- start initially with a Huffman tree $\mathcal{T}_0$ associated to the alphabet A (each symbol of A has frequency 1);
- if $\mathcal{T}_n$ is the current Huffman tree and the current input symbol is a (that is, $w = uav$ and u has been already processed), then output the code of a in $\mathcal{T}_n$ (this code is denoted by $code(a, \mathcal{T}_n)$) and update the tree $\mathcal{T}_n$ by applying to it the sibling transformation; the new tree is denoted $\mathcal{T}_{n+1}$.

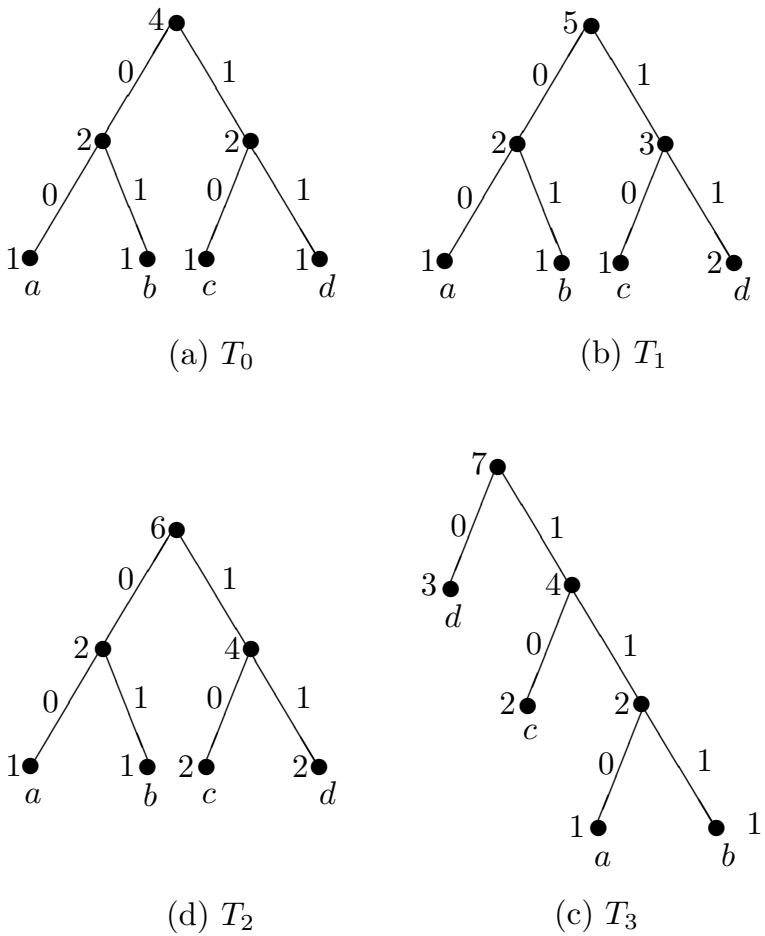
Huffman codes

The sibling transformation applied to symbol a and tree $\mathcal{T}_n$ consists of:

1. compare a to its successors in the tree (from left to right and from bottom to top). If the immediate successor has frequency $k+1$ or greater, where k is the frequency of a in $\mathcal{T}_n$, then the nodes are still in sorted order and there is no need to change anything. Otherwise, a should be swapped with the last successor which has frequency k or smaller (except that a should not be swapped with its parent);
2. increment the frequency of a (from k to $k+1$);
3. if a is the root, the loop halts; otherwise, the loop repeats with the parent of a.

Huffman codes

A sequence of Huffman trees used to encode the string *dcd* over the alphabet $\{a, b, c, d\}$:



(a) T_0 (b) T_1

(d) T_2 (c) T_3

Huffman codes

Huffman adaptive is not a variable-length code! The same character may be encoded by different code words!

Huffman adaptive is a time-varying code! For more details regarding time-varying codes see TMTE2002.pdf in the course web site.

Outline

Definition and examples

Word semigroups

Cyclic semigroups

Free semigroups and monoids

Variable-length codes

Huffman codes

Course readings

Course readings

1. F.L. Ţiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iaşi, 2006, pag. 179–243.

2. F.L. Ţiplea, E. Mäkinen, D. Trincă, C. Enea: *Characterization Results for Time-varying Codes*, Fundamenta Informaticae 53(2), 2002, 185-198.

Algebraic Foundations of Computer Science

Groups

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Definitions

Definition 1

A group is a 4-tuple $(G, \cdot, ', e)$ which consists of a set G, a binary operation $\cdot$ on G, a unary operation $'$ on G, and a nullary operation $e \in G$ such that:

- *$\cdot$ is associative;*
- $(\forall x \in G)(x \cdot e = e \cdot x = x)$;
- $(\forall x \in G)(x \cdot x' = x' \cdot x = e)$.

Remark 1

Let $(G, \cdot, ', e)$ be a group.

1. The element e is called the unity of G. It is unique and it is also denoted by 1_G or even 1;
2. For any x, x' is unique with the property $x \cdot x' = x' \cdot x = e$. x' is called the inverse of x and it is also denoted by x^{-1}.

Definitions

Conventions to be used when no confusions may arise:

- We will usually denote groups just by their carrier sets. That is, we will often say "Let G be a group";
- When the binary operation of a group is denoted additively (by $+$), then the unary operation will be denoted by "$-$" and the nullary operation by 0. However, in such a case, "$-$" should not be confused with the subtraction operation, and 0 with the number zero.
- We will often omit the symbol of the binary operation when two or more elements of the group are operated by it. That is, we will write ab instead of $a \cdot b$.

Definition 2

A group $(G, \cdot, ', e)$ is called commutative if $\cdot$ is a commutative operation.

Powers and multiples

1. Multiplicatively denoted groups:
   - $a^0 = e$;
   - $a^n = a^{n-1} \cdot a$, for any $n \geq 1$;
   - $a^{-1} = a'$, where a' is the inverse of a;
   - $a^{-n} = (a^{-1})^n$, for any $n \geq 1$;
2. Additively denoted groups:
   - $0a = 0$;
   - $na = (n-1)a + a$, for any $n \geq 1$;
   - $(-1)a = -a$, where $-a$ is the inverse of a;
   - $(-n)a = n(-a)$, for any $n \geq 1$.

Some basic properties of powers

Proposition 1

Let G be a group, $a, b \in G$, and $m, n \in \mathbb{Z}$. Then, the following properties hold true:

(1) $(a^{-1})^{-1} = a$;

(2) $(ab)^{-1} = b^{-1}a^{-1}$;

(3) $a^m a^n = a^{m+n} = a^n a^m$;

(4) $(a^m)^n = a^{mn} = (a^n)^m$;

(5) $a^{-m} = (a^{-1})^m = (a^m)^{-1}$.

You are invited to rewrite these properties under the additive notation.

Examples of groups

Example 3

1. $(\mathbb{Z}, +, -, 0)$, $(\mathbb{Q}, +, -, 0)$, $(\mathbb{R}, +, -, 0)$, and $(\mathbb{C}, +, -, 0)$ are commutative groups.
2. $(\mathbb{Q}^*, \cdot, ^{-1}, 1)$, $(\mathbb{R}^*, \cdot, ^{-1}, 1)$, and $(\mathbb{C}^*, \cdot, ^{-1}, 1)$ are commutative groups.
3. $(n\mathbb{Z}, +, -, 0)$ is a commutative group, and $(n\mathbb{Z}, \cdot, 1)$ is a commutative monoid.
4. $(\mathbb{Z}_m, +, -, 0)$ is a cyclic commutative group, and $(\mathbb{Z}_m^*, \cdot, ^{-1}, 1)$ is a commutative group, for any $m \geq 1$.
5. Let A be a set. The set of all bijective function from A to A, together with the function composition operation, the function inverse operation, and the identity function from A to A, forms a groups called the permutations group of A or the symmetric group of A. It is usually denoted by $Sym(A)$.

Solving equations in groups

Proposition 2

Let G be a semigroup.

(1) *If G is a group, then, for any $a, b \in G$, the equations $ax = b$ and $ya = b$ have unique solutions in G.*

(2) *If, for any $a, b \in G$, the equations $ax = b$ and $ya = b$ have unique solutions in G, then G is a group.*

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Subgroup

Definition 4

A group $(H, \circ, '', e_H)$ is a subgroup of a group $(G, \cdot, ', e_G)$ if $H \subseteq G$, $\circ = \cdot|_H$, $'' = '|_H$, and $e_H = e_G$.

When H is a subgroup of G we will write $H \leq G$.

Example 5

Considering the groups in Example 3, it follows:

- $\mathbb{Z} \leq \mathbb{Q} \leq \mathbb{R} \leq \mathbb{C}$;
- $\mathbb{Q}^* \leq \mathbb{R}^* \leq \mathbb{C}^*$;
- $n\mathbb{Z} \leq \mathbb{Z}$, for any $n \in \mathbb{Z}$. Moreover, any subgroup of $\mathbb{Z}$ is of the form $n\mathbb{Z}$, for some $n \geq 0$.

Subgroup: characterization

Proposition 3

Let $(G, \cdot, ', e)$ be a group and $H \subseteq G$ a non-empty subset. The following statements are equivalent:

(1) $H \leq G$;

(2) $ab \in H$ *and* $a' \in H$, *for any* $a, b \in H$;

(3) $ab' \in H$, *for any* $a, b \in H$.

Corollary 6

Let $(G, \cdot, ', e)$ be a finite group. Then, a non-empty subset H of G is a subgroup of G iff $ab \in H$, for any $a, b \in H$.

Equivalences induced by subgroups

Let G be a group and $H \leq G$. Define two binary relations on G, $\sim_H$ and ${}_H\!\sim$, by

- $a \sim_H b \quad \Leftrightarrow \quad (\exists c \in H)(b = ac)$
- $a \,{}_H\!\sim b \quad \Leftrightarrow \quad (\exists c \in H)(b = ca)$,

for $a, b \in G$.

Proposition 4

Let G be a group, $H \leq G$, and $a, b \in G$.

1. *$a \sim_H b$ iff $a'b \in H$.*
2. *$a \,{}_H\!\sim b$ iff $ba' \in H$.*
3. *$\sim_H$ and ${}_H\!\sim$ are equivalence relations on G.*
4. *$[a]_{\sim_H} = aH$ and $[a]_{{}_H\!\sim} = Ha$.*
5. *H, aH, and Ha are pairwise equipotent sets.*
6. *$\{Ha | a \in G\}$ and $\{aH | a \in G\}$ are equipotent sets.*

Lagrange's theorem

Let G be a finite group and $H \leq G$. The index of H in G is defined by

$$(G : H) = |\{Ha | a \in G\}| = |\{aH | a \in G\}|.$$

Theorem 7 (Lagrange's Theorem)

For any finite group G and $H \leq G$,

$$|G| = (G : H)|H|.$$

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Definitions

A group G is cyclic if it can be generated by one of its elements. That is,

- if G is written multiplicatively, then G is cyclic if

$$G = \langle a \rangle = \{a^n | n \in \mathbb{Z}\},$$

  for some $a \in G$;

- if G is written additively, then G is cyclic if

$$G = \langle a \rangle = \{na | n \in \mathbb{Z}\},$$

  for some $a \in G$.

Examples

Example 8

1. $(\mathbb{Z}, +, -, 0)$ is an infinite cyclic group generated by 1.
2. For any $m \geq 1$, $(\mathbb{Z}_m, +, -, 0)$ is a finite cyclic group:
   - if $m = 1$, then the group is generated by 0;
   - if $m > 1$, then the group is generated by 1.

Characterization of cyclic groups

Theorem 9

Let a be an element of a group $(G, \cdot, ', e)$. Then, exactly one of the following two properties holds true:

(1) $a^n \neq a^m$ *for any integers* $n \neq m$*, and the cyclic subgroup generated by* a *is isomorphic to* $(\mathbb{Z}, +, -, 0)$*;*

(2) *there exists* $r > 0$ *such that:*

  (a) $a^r = e$*;*

  (b) $a^u = a^v$ *iff* $u \equiv v \bmod r$*, for any* $u, v \in \mathbb{Z}$*;*

  (c) $\langle a \rangle = \{a^0, a^1, \ldots, a^{r-1}\}$ *has exactly* r *elements;*

  (d) *the subgroup* $\langle a \rangle$ *is isomorphic to the cyclic group* $(\mathbb{Z}_r, +, -, 0)$*.*

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Definitions and properties

The order of an element a of a group G, denoted $ord_G(a)$, is the order of the subgroup generated by a.

Theorem 10

Let $(G, \cdot, ', e)$ be a group and $a \in G$ be an element of finite order. Then:

(1) $ord_G(a) = min\{r \geq 1 | a^r = e\}$;

(2) *if G is finite, then* $ord_G(a) || G|$;

(3) *if G is finite, then* $a^{|G|} = e$;

(4) $(\forall s, t \in \mathbb{Z})(a^s = a^t \quad \Leftrightarrow \quad s \equiv t \ mod \ ord_G(a))$;

(5) $(\forall s \in \mathbb{Z})(a^s = e \quad \Leftrightarrow \quad ord_G(a)|s)$;

(6) $(\forall t \in \mathbb{Z})(ord_G(a^t) = ord_G(a)/(t, ord_G(a)))$;

(7) *if* $ord_G(a) = r_1 r_2$ *and* $r_1, r_2 > 1$, *then* $ord_G(a^{r_1}) = r_2$.

More properties

Corollary 11

Let $(G, \cdot, ', e)$ be a group and $a, b \in G$ be elements of finite order. If $ab = ba$ and $(ord_G(a), ord_G(b)) = 1$, then $ord_G(ab) = ord_G(a)ord_G(b)$.

Theorem 12

Let $(G, \cdot, ', e)$ be a finite group and $a \in G$. Then,

(1) $G = \langle a \rangle$ iff $ord_G(a) = |G|$;

(2) a generates G iff $a^{|G|/q} \neq e$, for any prime factor q of $|G|$;

(3) if a is a generator of G, then for any $t \in \mathbb{Z}$, a^t is a generator of G iff $(t, |G|) = 1$;

(4) if G is cyclic, then it has $\phi(|G|)$ generators.

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

The group $\mathbb{Z}_m^*$

Let $m \geq 1$. Recall that

$$\mathbb{Z}_m^* = \{a \in \mathbb{Z}_m | (a, m) = 1\}$$

and $(\mathbb{Z}_m^*, \cdot, ^{-1}, 1)$ is a commutative group. Moreover, $|\mathbb{Z}_m^*| = \phi(m)$.

Given $a \in \mathbb{Z}_m^*$, denote

$$ord_m(a) = ord_{\mathbb{Z}_m^*}(a).$$

$ord_m(a)$ is called the order of a modulo m.

Order of an element in $\mathbb{Z}_m^*$

Proposition 5

Let $m \geq 1$ and $a \in \mathbb{Z}_m^*$. Then:

(1) $ord_m(a) = min\{k \geq 1 | a^k \equiv 1 \ mod \ m\}$;

(2) *if* $a^k \equiv 1 \ mod \ m$, *then* $ord_m(a)|k$. *In particular,* $ord_m(a)|\phi(m)$;

(3) $ord_m(a) = \phi(m)$ *iff* $a^{\phi(m)/q} \not\equiv 1 \ mod \ m$, *for any prime factor* q *of* $\phi(m)$;

(4) $a^k \equiv a^l \ mod \ m$ *iff* $k \equiv l \ mod \ ord_m(a)$;

(5) $a^0 \ mod \ m, \ a^1 \ mod \ m, \ldots, a^{ord_m(a)-1} \ mod \ m$ *are pairwise distinct;*

(6) $ord_m(a^k \ mod \ m) = ord_m(a)/(k, ord_m(a))$, *for any* $k \geq 1$;

(7) *if* $ord_m(a) = d_1 d_2$, *then* $ord_m(a^{d_1} \ mod \ m) = d_2$.

Corollary 13

Let $m \geq 1$ and $a, b \in \mathbb{Z}_m^*$. If $ord_m(a)$ and $ord_m(b)$ are co-prime, then $ord_m(ab \ mod \ m) = ord_m(a) ord_m(b)$.

Primitive roots

When $\mathbb{Z}_m^*$ is a cyclic group, its generators are also called primitive roots modulo m.

Proposition 6

Let $m \geq 1$ and $a \in \mathbb{Z}_m^*$. Then:

(1) *a is a primitive root modulo m iff $ord_m(a) = \phi(m)$;*

(2) *a is a primitive root modulo m iff*

$$(\forall q)(q \text{ prime factor of } \phi(m) \quad \Rightarrow \quad a^{\phi(m)/q} \not\equiv 1 \bmod m);$$

(3) *if a is a primitive root modulo m, then, for any $k \geq 1$, a^k is a primitive root modulo m iff $(k, \phi(m)) = 1$;*

(4) *if there are primitive roots modulo m, then there are exactly $\phi(\phi(m))$ primitive roots.*

Primitive roots

Theorem 14 (Gauss)

There are primitive roots modulo m iff $m = 1, 2, 4, p^k, 2p^k$, where $p \geq 3$ is a prime number and $k \geq 1$.

Example 15

- There are primitive roots modulo 50 because $50 = 2 \cdot 5^2$. Moreover, there are $\phi(\phi(50)) = \phi(20) = 8$ primitive roots modulo 50.
- There is no primitive root modulo 150.

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

The discrete logarithm problem

If G is a finite cyclic group and a is a generator of G, then

$$G = \{a^0 = e, a^1, \ldots, a^{|G|-1}\}.$$

Given $b \in G$, there exists $k < |G|$ such that $b = a^k$. k is called the index of b w.r.t. a or the discrete logarithm of b to base a. When $G = \mathbb{Z}_m^*$, k is called the discrete logarithm of b to base a modulo m and it is usually denoted by $\log_a b \ mod \ m$.

Discrete Logarithm Problem (DLP)

Instance: finite cyclic group G, generator a of G, and $b \in G$;

Question: find $k < |G|$ such that $b = a^k$.

The discrete logarithm problem

- No efficient algorithm for computing general discrete algorithms is known;
- The naive approach is to raise a to powers i until the desired b is found (this method is sometimes called trial multiplication). The complexity of this method is linear in the size of the group and, therefore, it is exponential in the number of bits of the size of the group;
- While computing discrete logarithms is apparently difficult, the inverse problem of discrete exponentiation is easy (polynomial). This asymmetry has been exploited in the construction of cryptographic schemes: ElGamal encryption and digital signature, Diffie-Hellman key exchange protocol etc.

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

ElGamal digital signature

- let p be a (large) prime and α be a primitive root in $\mathbb{Z}_p^*$;
- $\mathcal{P} = \mathbb{Z}_p^*$;
- $\mathcal{S} = \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$;
- $\mathcal{K} = \{(p, \alpha, a, \beta) | a \in \mathbb{Z}_{p-1},\ \beta = \alpha^a \ mod\ p\}$;
- for any $K = (p, \alpha, a, \beta)$ and $k \in \mathbb{Z}_{p-1}^*$, and any $x \in \mathbb{Z}_p^*$,
  - the message x is signed by
    $$sig_K(x, k) = (\gamma, \delta),$$
    where $\gamma = \alpha^k \ mod\ p$ and $\delta = (x - a\gamma)k^{-1} \ mod\ (p-1)$
  - the verification of the signature (γ, δ) for x is performed by
    $$ver_K(x, (\gamma, \delta)) = 1 \quad \Leftrightarrow \quad \beta^\gamma \gamma^\delta \equiv \alpha^x \ mod\ p;$$
- p, α and β are public, and a and k are secret.

Example

Example 16

Let $p = 467$, $\alpha = 2$, and $a = 127$. Then,

$$\beta = \alpha^a \ mod \ p = 2^{127} \ mod \ 467 = 132.$$

Assume that we want to sign $x = 100$ using $k = 213$ ($k \in \mathbb{Z}_{466}^*$ and $k^{-1} = 431$). Then:

$$\gamma = 2^{213} \ mod \ 467 = 29,$$

and

$$\delta = (100 - 127 \cdot 29) \cdot 431 \ mod \ 466 = 51.$$

Therefore, $sig_K(x, k) = (29, 51)$.
In order to verify the signature we compute

$$132^{29} \cdot 29^{51} \ mod \ 467 \quad \text{and} \quad 2^{100} \ mod \ 467$$

and accept the signature if they are equal.

Signing two messages by the same k

If the secret value k is used to sign two distinct messages x_1 and x_2, then the secret parameter a could be easily computed.

Let $sig_K(x_1) = (\gamma, \delta_1)$ and $sig_K(x_2) = (\gamma, \delta_2)$ (the same k has been used). Therefore,

$$\beta^\gamma \gamma^{\delta_1} \equiv \alpha^{x_1} \ mod \ p$$

and

$$\beta^\gamma \gamma^{\delta_2} \equiv \alpha^{x_2} \ mod \ p,$$

which lead to

$$\alpha^{x_1 - x_2} \equiv \gamma^{\delta_1 - \delta_2} \ mod \ p.$$

Because $\gamma = \alpha^k \ mod \ p$, we get

$$\alpha^{x_1 - x_2} \equiv \alpha^{k(\delta_1 - \delta_2)} \ mod \ p,$$

which is equivalent to

$$k(\delta_1 - \delta_2) \equiv x_1 - x_2 \ mod \ (p-1).$$

Signing two messages by the same k

The solutions modulo $p-1$ to the above equation are of the form

$$(k_0 + i(p-1)/d) \ mod \ (p-1),$$

where k_0 is an arbitrary solution, $d = (\delta_1 - \delta_2, p-1)$, and $0 \leq i < d$. k_0 can be obtained by the extended Euclidean algorithm, and k can be obtained by checking the equation $\gamma \equiv \alpha^k \ mod \ p$.
If k is recovered, then the parameter a can be easily recovered from the equation $\delta = (x - a\gamma)k^{-1} \ mod \ (p-1)$, and the signature scheme is broken.

Digital signature standard

- Digital Signature Standard (DSS) is the American standard for digital signatures;
- DSS was proposed by NIST in 1991, and adopted in 1994;
- DSS is a variation of the ElGamal digital signature. This variation is based on the following remark: the prime p in the ElGamal digital signature should be a 512-bit or 1024-bit number in order to ensure security. This fact leads to signatures that are too large to be used on smart cards;
- DSS modifies ElGamal digital signature so that the computations are done in a subgroup $\mathbb{Z}_q$ of $\mathbb{Z}_p^*$ by using an element $\alpha \in \mathbb{Z}_p^*$ of order q.

Digital signature standard

- Let p a prime, q a prime factor of $p-1$, and α an element of order q in $\mathbb{Z}_p^*$;
- $\mathcal{P} = \mathbb{Z}_p^*$;
- $\mathcal{S} = \mathbb{Z}_q \times \mathbb{Z}_q$;
- $\mathcal{K} = \{(p, q, \alpha, a, \beta) | a \in \mathbb{Z}_q \ \wedge \ \beta = \alpha^a \ mod \ p\}$;
- For any $K = (p, q, \alpha, a, \beta)$ and $k \in \mathbb{Z}_q^*$, and any $x \in \mathbb{Z}_p^*$,
  - $sig_K(x, k) = (\gamma, \delta)$, where $\gamma = (\alpha^k \ mod \ p) \ mod \ q$ and $\delta = (x + a\gamma)k^{-1} \ mod \ q$;
  - $ver_K(x, (\gamma, \delta)) = 1 \quad \Leftrightarrow \quad (\alpha^{e_1}\beta^{e_2} \ mod \ p) \ mod \ q = \gamma$, where $e_1 = x\delta^{-1} \ mod \ q$ and $e_2 = \gamma\delta^{-1} \ mod \ q$;
- p, q, α, and β are public, and a is secret.

Computing primitive roots

Recall that an element $\alpha \in \mathbb{Z}_m^*$ is a primitive root modulo m iff $\alpha^{\phi(m)/q} \not\equiv 1 \ mod \ m$, for any prime factor q of $\phi(m)$.

If $p = 2q + 1$ and p and q are primes, then $\alpha \in \mathbb{Z}_p^*$ is a primitive root modulo p iff $\alpha^2 \not\equiv 1 \ mod \ p$ and $\alpha^q \not\equiv 1 \ mod \ p$. Moreover, there are $\phi(\phi(p)) = q - 1$ primitive roots modulo p, which shows that the probability that a randomly generated number $\alpha \in \mathbb{Z}_p^*$ is a primitive root is approximately $1/2$.

If α is a primitive root modulo a prime p and q is a prime factor of $p - 1$, then $\alpha^{\frac{p-1}{q}} \ mod \ p$ is an element or order q.

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Course readings

1. F.L. Ţiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iaşi, 2006, pag. 245–289.

Algebraic Foundations of Computer Science

Groups

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Definitions

Definition 1

A group is a 4-tuple $(G, \cdot, ', e)$ which consists of a set G, a binary operation $\cdot$ on G, a unary operation $'$ on G, and a nullary operation $e \in G$ such that:

- *$\cdot$ is associative;*
- $(\forall x \in G)(x \cdot e = e \cdot x = x)$;
- $(\forall x \in G)(x \cdot x' = x' \cdot x = e)$.

Remark 1

Let $(G, \cdot, ', e)$ be a group.

1. The element e is called the unity of G. It is unique and it is also denoted by 1_G or even 1;
2. For any x, x' is unique with the property $x \cdot x' = x' \cdot x = e$. x' is called the inverse of x and it is also denoted by x^{-1}.

Definitions

Conventions to be used when no confusions may arise:

- We will usually denote groups just by their carrier sets. That is, we will often say "Let G be a group";
- When the binary operation of a group is denoted additively (by $+$), then the unary operation will be denoted by "$-$" and the nullary operation by 0. However, in such a case, "$-$" should not be confused with the subtraction operation, and 0 with the number zero.
- We will often omit the symbol of the binary operation when two or more elements of the group are operated by it. That is, we will write ab instead of $a \cdot b$.

Definition 2

A group $(G, \cdot, ', e)$ is called commutative if $\cdot$ is a commutative operation.

Powers and multiples

1. Multiplicatively denoted groups:
   - $a^0 = e$;
   - $a^n = a^{n-1} \cdot a$, for any $n \geq 1$;
   - $a^{-1} = a'$, where a' is the inverse of a;
   - $a^{-n} = (a^{-1})^n$, for any $n \geq 1$;
2. Additively denoted groups:
   - $0a = 0$;
   - $na = (n-1)a + a$, for any $n \geq 1$;
   - $(-1)a = -a$, where $-a$ is the inverse of a;
   - $(-n)a = n(-a)$, for any $n \geq 1$.

Some basic properties of powers

Proposition 1

Let G be a group, $a, b \in G$, and $m, n \in \mathbb{Z}$. Then, the following properties hold true:

(1) $(a^{-1})^{-1} = a$;

(2) $(ab)^{-1} = b^{-1}a^{-1}$;

(3) $a^m a^n = a^{m+n} = a^n a^m$;

(4) $(a^m)^n = a^{mn} = (a^n)^m$;

(5) $a^{-m} = (a^{-1})^m = (a^m)^{-1}$.

You are invited to rewrite these properties under the additive notation.

Examples of groups

Example 3

1. $(\mathbb{Z}, +, -, 0)$, $(\mathbb{Q}, +, -, 0)$, $(\mathbb{R}, +, -, 0)$, and $(\mathbb{C}, +, -, 0)$ are commutative groups.
2. $(\mathbb{Q}^*, \cdot, ^{-1}, 1)$, $(\mathbb{R}^*, \cdot, ^{-1}, 1)$, and $(\mathbb{C}^*, \cdot, ^{-1}, 1)$ are commutative groups.
3. $(n\mathbb{Z}, +, -, 0)$ is a commutative group, and $(n\mathbb{Z}, \cdot, 1)$ is a commutative monoid.
4. $(\mathbb{Z}_m, +, -, 0)$ is a cyclic commutative group, and $(\mathbb{Z}_m^*, \cdot, ^{-1}, 1)$ is a commutative group, for any $m \geq 1$.
5. Let A be a set. The set of all bijective function from A to A, together with the function composition operation, the function inverse operation, and the identity function from A to A, forms a groups called the permutations group of A or the symmetric group of A. It is usually denoted by $Sym(A)$.

Solving equations in groups

Proposition 2

Let G be a semigroup.

(1) *If G is a group, then, for any $a, b \in G$, the equations $ax = b$ and $ya = b$ have unique solutions in G.*

(2) *If, for any $a, b \in G$, the equations $ax = b$ and $ya = b$ have unique solutions in G, then G is a group.*

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Subgroup

Definition 4

A group $(H, \circ, '', e_H)$ is a subgroup of a group $(G, \cdot, ', e_G)$ if $H \subseteq G$, $\circ = \cdot|_H$, $'' = '|_H$, and $e_H = e_G$.

When H is a subgroup of G we will write $H \leq G$.

Example 5

Considering the groups in Example 3, it follows:

- $\mathbb{Z} \leq \mathbb{Q} \leq \mathbb{R} \leq \mathbb{C}$;
- $\mathbb{Q}^* \leq \mathbb{R}^* \leq \mathbb{C}^*$;
- $n\mathbb{Z} \leq \mathbb{Z}$, for any $n \in \mathbb{Z}$. Moreover, any subgroup of $\mathbb{Z}$ is of the form $n\mathbb{Z}$, for some $n \geq 0$.

Subgroup: characterization

Proposition 3

Let $(G, \cdot, ', e)$ be a group and $H \subseteq G$ a non-empty subset. The following statements are equivalent:

(1) $H \leq G$;

(2) $ab \in H$ *and* $a' \in H$, *for any* $a, b \in H$;

(3) $ab' \in H$, *for any* $a, b \in H$.

Corollary 6

Let $(G, \cdot, ', e)$ be a finite group. Then, a non-empty subset H of G is a subgroup of G iff $ab \in H$, for any $a, b \in H$.

Equivalences induced by subgroups

Let G be a group and $H \leq G$. Define two binary relations on G, $\sim_H$ and ${}_H\!\sim$, by

- $a \sim_H b \quad \Leftrightarrow \quad (\exists c \in H)(b = ac)$
- $a \,{}_H\!\sim b \quad \Leftrightarrow \quad (\exists c \in H)(b = ca)$,

for $a, b \in G$.

Proposition 4

Let G be a group, $H \leq G$, and $a, b \in G$.

1. *$a \sim_H b$ iff $a'b \in H$.*
2. *$a \,{}_H\!\sim b$ iff $ba' \in H$.*
3. *$\sim_H$ and ${}_H\!\sim$ are equivalence relations on G.*
4. *$[a]_{\sim_H} = aH$ and $[a]_{{}_H\!\sim} = Ha$.*
5. *H, aH, and Ha are pairwise equipotent sets.*
6. *$\{Ha | a \in G\}$ and $\{aH | a \in G\}$ are equipotent sets.*

Lagrange's theorem

Let G be a finite group and $H \leq G$. The index of H in G is defined by

$$(G : H) = |\{Ha | a \in G\}| = |\{aH | a \in G\}|.$$

Theorem 7 (Lagrange's Theorem)

For any finite group G and $H \leq G$,

$$|G| = (G : H)|H|.$$

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Definitions

A group G is cyclic if it can be generated by one of its elements. That is,

- if G is written multiplicatively, then G is cyclic if

$$G = \langle a \rangle = \{a^n | n \in \mathbb{Z}\},$$

  for some $a \in G$;

- if G is written additively, then G is cyclic if

$$G = \langle a \rangle = \{na | n \in \mathbb{Z}\},$$

  for some $a \in G$.

Examples

Example 8

1. $(\mathbb{Z}, +, -, 0)$ is an infinite cyclic group generated by 1.
2. For any $m \geq 1$, $(\mathbb{Z}_m, +, -, 0)$ is a finite cyclic group:
   - if $m = 1$, then the group is generated by 0;
   - if $m > 1$, then the group is generated by 1.

Characterization of cyclic groups

Theorem 9

Let a be an element of a group $(G, \cdot, ', e)$. Then, exactly one of the following two properties holds true:

(1) $a^n \neq a^m$ *for any integers* $n \neq m$*, and the cyclic subgroup generated by* a *is isomorphic to* $(\mathbb{Z}, +, -, 0)$*;*

(2) *there exists* $r > 0$ *such that:*

  (a) $a^r = e$*;*

  (b) $a^u = a^v$ *iff* $u \equiv v \bmod r$*, for any* $u, v \in \mathbb{Z}$*;*

  (c) $\langle a \rangle = \{a^0, a^1, \ldots, a^{r-1}\}$ *has exactly* r *elements;*

  (d) *the subgroup* $\langle a \rangle$ *is isomorphic to the cyclic group* $(\mathbb{Z}_r, +, -, 0)$*.*

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Definitions and properties

The order of an element a of a group G, denoted $ord_G(a)$, is the order of the subgroup generated by a.

Theorem 10

Let $(G, \cdot, ', e)$ be a group and $a \in G$ be an element of finite order. Then:

(1) $ord_G(a) = min\{r \geq 1 | a^r = e\}$;

(2) *if G is finite, then* $ord_G(a) || G|$;

(3) *if G is finite, then* $a^{|G|} = e$;

(4) $(\forall s, t \in \mathbb{Z})(a^s = a^t \quad \Leftrightarrow \quad s \equiv t \ mod \ ord_G(a))$;

(5) $(\forall s \in \mathbb{Z})(a^s = e \quad \Leftrightarrow \quad ord_G(a)|s)$;

(6) $(\forall t \in \mathbb{Z})(ord_G(a^t) = ord_G(a)/(t, ord_G(a)))$;

(7) *if* $ord_G(a) = r_1 r_2$ *and* $r_1, r_2 > 1$, *then* $ord_G(a^{r_1}) = r_2$.

More properties

Corollary 11

Let $(G, \cdot, ', e)$ be a group and $a, b \in G$ be elements of finite order. If $ab = ba$ and $(ord_G(a), ord_G(b)) = 1$, then $ord_G(ab) = ord_G(a)ord_G(b)$.

Theorem 12

Let $(G, \cdot, ', e)$ be a finite group and $a \in G$. Then,

(1) $G = \langle a \rangle$ iff $ord_G(a) = |G|$;

(2) a generates G iff $a^{|G|/q} \neq e$, for any prime factor q of $|G|$;

(3) if a is a generator of G, then for any $t \in \mathbb{Z}$, a^t is a generator of G iff $(t, |G|) = 1$;

(4) if G is cyclic, then it has $\phi(|G|)$ generators.

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

The group $\mathbb{Z}_m^*$

Let $m \geq 1$. Recall that

$$\mathbb{Z}_m^* = \{a \in \mathbb{Z}_m | (a, m) = 1\}$$

and $(\mathbb{Z}_m^*, \cdot, ^{-1}, 1)$ is a commutative group. Moreover, $|\mathbb{Z}_m^*| = \phi(m)$.

Given $a \in \mathbb{Z}_m^*$, denote

$$ord_m(a) = ord_{\mathbb{Z}_m^*}(a).$$

$ord_m(a)$ is called the order of a modulo m.

Order of an element in $\mathbb{Z}_m^*$

Proposition 5

Let $m \geq 1$ and $a \in \mathbb{Z}_m^*$. Then:

(1) $ord_m(a) = min\{k \geq 1 | a^k \equiv 1 \ mod \ m\}$;

(2) *if* $a^k \equiv 1 \ mod \ m$, *then* $ord_m(a)|k$. *In particular,* $ord_m(a)|\phi(m)$;

(3) $ord_m(a) = \phi(m)$ *iff* $a^{\phi(m)/q} \not\equiv 1 \ mod \ m$, *for any prime factor* q *of* $\phi(m)$;

(4) $a^k \equiv a^l \ mod \ m$ *iff* $k \equiv l \ mod \ ord_m(a)$;

(5) $a^0 \ mod \ m, \ a^1 \ mod \ m, \ldots, a^{ord_m(a)-1} \ mod \ m$ *are pairwise distinct;*

(6) $ord_m(a^k \ mod \ m) = ord_m(a)/(k, ord_m(a))$, *for any* $k \geq 1$;

(7) *if* $ord_m(a) = d_1 d_2$, *then* $ord_m(a^{d_1} \ mod \ m) = d_2$.

Corollary 13

Let $m \geq 1$ and $a, b \in \mathbb{Z}_m^*$. If $ord_m(a)$ and $ord_m(b)$ are co-prime, then $ord_m(ab \ mod \ m) = ord_m(a) ord_m(b)$.

Primitive roots

When $\mathbb{Z}_m^*$ is a cyclic group, its generators are also called primitive roots modulo m.

Proposition 6

Let $m \geq 1$ and $a \in \mathbb{Z}_m^*$. Then:

(1) *a is a primitive root modulo m iff $ord_m(a) = \phi(m)$;*

(2) *a is a primitive root modulo m iff*

$$(\forall q)(q \text{ prime factor of } \phi(m) \quad \Rightarrow \quad a^{\phi(m)/q} \not\equiv 1 \bmod m);$$

(3) *if a is a primitive root modulo m, then, for any $k \geq 1$, a^k is a primitive root modulo m iff $(k, \phi(m)) = 1$;*

(4) *if there are primitive roots modulo m, then there are exactly $\phi(\phi(m))$ primitive roots.*

Primitive roots

Theorem 14 (Gauss)

There are primitive roots modulo m iff $m = 1, 2, 4, p^k, 2p^k$, where $p \geq 3$ is a prime number and $k \geq 1$.

Example 15

- There are primitive roots modulo 50 because $50 = 2 \cdot 5^2$. Moreover, there are $\phi(\phi(50)) = \phi(20) = 8$ primitive roots modulo 50.
- There is no primitive root modulo 150.

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

The discrete logarithm problem

If G is a finite cyclic group and a is a generator of G, then

$$G = \{a^0 = e, a^1, \ldots, a^{|G|-1}\}.$$

Given $b \in G$, there exists $k < |G|$ such that $b = a^k$. k is called the index of b w.r.t. a or the discrete logarithm of b to base a. When $G = \mathbb{Z}_m^*$, k is called the discrete logarithm of b to base a modulo m and it is usually denoted by $\log_a b \ mod \ m$.

Discrete Logarithm Problem (DLP)

Instance: finite cyclic group G, generator a of G, and $b \in G$;

Question: find $k < |G|$ such that $b = a^k$.

The discrete logarithm problem

- No efficient algorithm for computing general discrete algorithms is known;
- The naive approach is to raise a to powers i until the desired b is found (this method is sometimes called trial multiplication). The complexity of this method is linear in the size of the group and, therefore, it is exponential in the number of bits of the size of the group;
- While computing discrete logarithms is apparently difficult, the inverse problem of discrete exponentiation is easy (polynomial). This asymmetry has been exploited in the construction of cryptographic schemes: ElGamal encryption and digital signature, Diffie-Hellman key exchange protocol etc.

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

ElGamal digital signature

- let p be a (large) prime and α be a primitive root in $\mathbb{Z}_p^*$;
- $\mathcal{P} = \mathbb{Z}_p^*$;
- $\mathcal{S} = \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$;
- $\mathcal{K} = \{(p, \alpha, a, \beta) | a \in \mathbb{Z}_{p-1},\ \beta = \alpha^a \ mod\ p\}$;
- for any $K = (p, \alpha, a, \beta)$ and $k \in \mathbb{Z}_{p-1}^*$, and any $x \in \mathbb{Z}_p^*$,
  - the message x is signed by
    $$sig_K(x, k) = (\gamma, \delta),$$
    where $\gamma = \alpha^k \ mod\ p$ and $\delta = (x - a\gamma)k^{-1} \ mod\ (p-1)$
  - the verification of the signature (γ, δ) for x is performed by
    $$ver_K(x, (\gamma, \delta)) = 1 \quad \Leftrightarrow \quad \beta^\gamma \gamma^\delta \equiv \alpha^x \ mod\ p;$$
- p, α and β are public, and a and k are secret.

Example

Example 16

Let $p = 467$, $\alpha = 2$, and $a = 127$. Then,

$$\beta = \alpha^a \ mod \ p = 2^{127} \ mod \ 467 = 132.$$

Assume that we want to sign $x = 100$ using $k = 213$ ($k \in \mathbb{Z}_{466}^*$ and $k^{-1} = 431$). Then:

$$\gamma = 2^{213} \ mod \ 467 = 29,$$

and

$$\delta = (100 - 127 \cdot 29) \cdot 431 \ mod \ 466 = 51.$$

Therefore, $sig_K(x, k) = (29, 51)$.
In order to verify the signature we compute

$$132^{29} \cdot 29^{51} \ mod \ 467 \quad \text{and} \quad 2^{100} \ mod \ 467$$

and accept the signature if they are equal.

Signing two messages by the same k

If the secret value k is used to sign two distinct messages x_1 and x_2, then the secret parameter a could be easily computed.

Let $sig_K(x_1) = (\gamma, \delta_1)$ and $sig_K(x_2) = (\gamma, \delta_2)$ (the same k has been used). Therefore,

$$\beta^\gamma \gamma^{\delta_1} \equiv \alpha^{x_1} \ mod \ p$$

and

$$\beta^\gamma \gamma^{\delta_2} \equiv \alpha^{x_2} \ mod \ p,$$

which lead to

$$\alpha^{x_1 - x_2} \equiv \gamma^{\delta_1 - \delta_2} \ mod \ p.$$

Because $\gamma = \alpha^k \ mod \ p$, we get

$$\alpha^{x_1 - x_2} \equiv \alpha^{k(\delta_1 - \delta_2)} \ mod \ p,$$

which is equivalent to

$$k(\delta_1 - \delta_2) \equiv x_1 - x_2 \ mod \ (p-1).$$

Signing two messages by the same k

The solutions modulo $p-1$ to the above equation are of the form

$$(k_0 + i(p-1)/d) \ mod \ (p-1),$$

where k_0 is an arbitrary solution, $d = (\delta_1 - \delta_2, p-1)$, and $0 \leq i < d$. k_0 can be obtained by the extended Euclidean algorithm, and k can be obtained by checking the equation $\gamma \equiv \alpha^k \ mod \ p$.
If k is recovered, then the parameter a can be easily recovered from the equation $\delta = (x - a\gamma)k^{-1} \ mod \ (p-1)$, and the signature scheme is broken.

Digital signature standard

- Digital Signature Standard (DSS) is the American standard for digital signatures;
- DSS was proposed by NIST in 1991, and adopted in 1994;
- DSS is a variation of the ElGamal digital signature. This variation is based on the following remark: the prime p in the ElGamal digital signature should be a 512-bit or 1024-bit number in order to ensure security. This fact leads to signatures that are too large to be used on smart cards;
- DSS modifies ElGamal digital signature so that the computations are done in a subgroup $\mathbb{Z}_q$ of $\mathbb{Z}_p^*$ by using an element $\alpha \in \mathbb{Z}_p^*$ of order q.

Digital signature standard

- Let p a prime, q a prime factor of $p-1$, and α an element of order q in $\mathbb{Z}_p^*$;
- $\mathcal{P} = \mathbb{Z}_p^*$;
- $\mathcal{S} = \mathbb{Z}_q \times \mathbb{Z}_q$;
- $\mathcal{K} = \{(p, q, \alpha, a, \beta) | a \in \mathbb{Z}_q \ \wedge \ \beta = \alpha^a \ mod \ p\}$;
- For any $K = (p, q, \alpha, a, \beta)$ and $k \in \mathbb{Z}_q^*$, and any $x \in \mathbb{Z}_p^*$,
  - $sig_K(x, k) = (\gamma, \delta)$, where $\gamma = (\alpha^k \ mod \ p) \ mod \ q$ and $\delta = (x + a\gamma)k^{-1} \ mod \ q$;
  - $ver_K(x, (\gamma, \delta)) = 1 \quad \Leftrightarrow \quad (\alpha^{e_1}\beta^{e_2} \ mod \ p) \ mod \ q = \gamma$, where $e_1 = x\delta^{-1} \ mod \ q$ and $e_2 = \gamma\delta^{-1} \ mod \ q$;
- p, q, α, and β are public, and a is secret.

Computing primitive roots

Recall that an element $\alpha \in \mathbb{Z}_m^*$ is a primitive root modulo m iff $\alpha^{\phi(m)/q} \not\equiv 1 \ mod \ m$, for any prime factor q of $\phi(m)$.

If $p = 2q + 1$ and p and q are primes, then $\alpha \in \mathbb{Z}_p^*$ is a primitive root modulo p iff $\alpha^2 \not\equiv 1 \ mod \ p$ and $\alpha^q \not\equiv 1 \ mod \ p$. Moreover, there are $\phi(\phi(p)) = q - 1$ primitive roots modulo p, which shows that the probability that a randomly generated number $\alpha \in \mathbb{Z}_p^*$ is a primitive root is approximately $1/2$.

If α is a primitive root modulo a prime p and q is a prime factor of $p - 1$, then $\alpha^{\frac{p-1}{q}} \ mod \ p$ is an element or order q.

Outline

Definitions, examples, basic properties

Subgroups. Lagrange's theorem

Cyclic groups

Order of an element

The group $\mathbb{Z}_m^*$

The discrete logarithm problem

Applications to cryptography

Course readings

Course readings

1. F.L. Ţiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iaşi, 2006, pag. 245–289.

Algebraic Foundations of Computer Science

Rings and Fields

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Definitions and examples

Ring homomorphisms

Characteristic of a ring

Finite fields

Course readings

Outline

Definitions and examples

Ring homomorphisms

Characteristic of a ring

Finite fields

Course readings

Definitions and examples

Definition 1

A ring is an algebraic structure $(R, +, -, 0, \cdot)$ such that:

- $(R, +, -, 0)$ is commutative group;
- $(R, \cdot)$ is a semigroup;
- $\cdot$ is left- and right-distributive over $+$.

$a \cdot (b + c) = a \cdot b + a \cdot c$

Remark 1

Let $(R, +, -, 0, \cdot)$ be a ring.

1. *0 is called the zero element of R; it is unique;*
2. *If $\cdot$ is commutative then the ring is called commutative;*
3. *We will usually denote rings just by their carrier sets. That is, we will often say "Let R be a ring".*

Definitions and examples

Proposition 1

Let $(R, +, -, 0, \cdot)$ be a ring. Then:

(1) $a0 = 0a = 0$, *for any* $a \in R$;

(2) $(-a)b = a(-b) = -(ab)$, *for any* $a, b \in R$;

(3) $(-a)(-b) = ab$, *for any* $a, b \in R$;

(4) $a(b - c) = ab - ac$ *and* $(b - c)a = ba - ca$, *for any* $a, b, c \in R$;

(5) $(\sum_{i=1}^{n} a_i)(\sum_{j=1}^{m} b_j) = \sum_{i=1}^{n} \sum_{j=1}^{m} a_i b_j$, *for any* $n, m \geq 1$ *and* $a_i, b_j \in R$, $1 \leq i \leq n$, *and* $1 \leq j \leq m$.

ind

Definitions and examples

$ma = \underbrace{a + a + \dots + a}_{m \text{ ori}}$

$0 \cdot a = 0$

$(-m)a = \underbrace{(-a) + \dots + (-a)}_{m \text{ ori}}$

$m \geq 0$

Proposition 2

Let R be a ring. Then:

(1) $(-m)a = -(ma)$;

(2) $(m+n)a = ma + na$;

(3) $m(a+b) = ma + mb$;

(4) $(mn)a = m(na)$;

(5) $m(ab) = (ma)b = a(mb)$;

(6) $(ma)(nb) = (mn)(ab)$,

for any $a, b \in R$ and $m, n \geq 1$.

Definitions and examples

Proposition 3

Let R be a commutative ring. Then,

$$(a+b)^n = \sum_{k=0}^{n} C_n^k a^{n-k} b^k,$$

for any $a, b \in R$ and $n \geq 1$, where $C_n^k = n!/(k!(n-k)!)$, for any $0 \leq k \leq n$.

Proof.

By mathematical induction on $n \geq 1$. □

$$\boxed{C_n^k + C_n^{k-1} = C_{n+1}^{k+1}}$$

$$(a+b)^{n+1} = (a+b)^n (a+b) = \left(\sum \cdots\right)(a+b)$$

Definitions and examples

Definition 2

A ring with unity/identity is an algebraic structure $(R, +, -, 0, \cdot, e)$ which satisfies:

- $(R, +, -, 0)$ is a commutative group;
- $(R, \cdot, e)$ is a monoid;
- $\cdot$ is left- and right-distributive over $+$.

The element e, also denoted by 1_R or 1, is called the unity/identity of R.

Proposition 4

$a = ae = a0 = 0$

If $(R, +, -, 0, \cdot, e)$ *is a ring with unity then* $e = 0$ *iff* $R = \{0\}$.

A ring with unity $(R, +, -, 0, \cdot, e)$ which satisfies $e = 0$ is called a trivial/null ring.

Definitions and examples

Definition 3

(1) A division ring is an algebraic structure $(R, +, -, 0, \cdot, ', e)$ which satisfies:

- $(R, +, -, 0)$ is a commutative group;
- $(R, \cdot, e)$ is a monoid and $e \neq 0$;
- $'$ is a unary operation which satisfies $aa' = a'a = e$, for any $a \neq 0$;
- $\cdot$ is left- and right-distributive over $+$.

(2) A commutative division ring is called a field.

Definition 4

(1) An element $a \in R - \{0\}$ of a ring R is called a zero divisor if there exists $b \in R - \{0\}$ such that $ab = 0$ or $ba = 0$.

(2) A commutative ring R with unity $e \neq 0$ and with no zero divisors is called an integral domain.

Definitions and examples

Proposition 5

1. If R is a ring and $c \in R - \{0\}$ is not a zero divisor, then $ac = bc$ ($ca = cb$) implies $a = b$, for any $a, b \in R$.
2. Division rings do not have zero divisors.
3. Any field is an integral domain.
4. Any finite integral domain is a field.
5. Let $p \geq 2$. $\mathbb{Z}_p$ is a field iff p is a prime.

$ac = bc \Rightarrow (a-b)c = 0$, $c \neq 0$, fără div de zero $\Rightarrow a - b = 0 \Rightarrow a = b$

Definitions and examples

Example 5

(1) Let $(R, +, -, 0)$ be a commutative group. Define on R the binary operation $\cdot$ by $a \cdot b = 0$, for any $a, b \in R$. Then, $(R, +, -, 0, \cdot)$ is a ring.

(2) $\mathbb{Z}$, together with addition and multiplication, form an integral domain, but not a field.

(3) $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$, together with addition and multiplication, form fields.

(4) $n\mathbb{Z}$ is a commutative ring with no zero divisors. This ring has unity only if $n = -1$, $n = 0$, or $n = 1$ (for $n = 0$, the ring is null).

(5) $\mathbb{Z}_n$ is a commutative ring with unity. If n is a prime, then $\mathbb{Z}_n$ is a field.

Outline

Definitions and examples

Ring homomorphisms

Characteristic of a ring

Finite fields

Course readings

Ring homomorphisms

Definition 6

Let R_1 and R_2 be rings. A function $h : R_1 \to R_2$ is a ring homomorphism if

- $h(a + b) = h(a) + h(b)$;
- $h(ab) = h(a)h(b)$,

for any $a, b \in R_1$

Avînd în vedere
$h(-a) = -h(a)$
$h(0_{R_1}) = 0_{R_2}$

The second property in the definition above may only be required for $a, b \in R_1 - \{0\}$. Indeed, if, for instance, $b = 0$, then

$$h(a0) = h(0) = 0 = h(a)0 = h(a)h(0).$$

If R_1 and R_2 have units e_1 and e_2, then the property

- $h(e_1) = e_2$

is required too.

Outline

Definitions and examples

Ring homomorphisms

Characteristic of a ring

Finite fields

Course readings

Characteristic of a ring

Definition 7

We say that a ring R has characteristic $n \geq 1$, if n is the smallest natural number such that $na = 0$, for any $a \in R$ (if there is such a number n).

If does not exist $n \geq 1$ with $na = 0$ for any $a \in R$, then we say that R has characteristic zero.

The characteristic of a ring R will be denoted by $char(R)$.

Remark 2

A ring with unity $e \neq 0$ cannot have the characteristic 1. Therefore, the only ring of characteristic 1 is the null ring.

Characteristic of a ring

Example 8

(1) $\mathbb{Z}_m$ has characteristic m, for any $m \geq 1$.

(2) $\mathbb{Z}$ is an integral domain of characteristic zero.

(3) $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ are fields of characteristic zero.

Characteristic of a ring

Theorem 9

Let $(R, +, -, 0, \cdot, e)$ be a ring with unity of characteristic $n \geq 1$.

(1) *n is the smallest non-zero natural number which satisfies $ne = 0$.*

(2) *If $e \neq 0$ and R does not have zero divisors, then n is a prime.*

dim 2 (3) *If the characteristic of an integral domain is not zero, then it is a prime.*

dim 3 (4) *The characteristic of a finite field is a prime number.*

(1) char(R), dacă este $\neq 0$, este cel mai mic n a.î. $n \cdot e = 0$

$$na = n \cdot (e \cdot a) = (n \cdot e) a = 0$$

(2) Ca la "$\mathbb{Z}_p$ corp $\Rightarrow$ p prim".

Outline

Definitions and examples

Ring homomorphisms

Characteristic of a ring

Finite fields

Course readings

Finite fields

Theorem 10

If R is a finite field of characteristic p, then $|R| = p^n$, for some $n \geq 1$.

Theorem 11

For any prime p and $n \geq 1$, there exists a field with p^n elements.

Theorem 12

Any two finite fields with the same number of elements are isomorphic.

The finite field with p^n elements, which is unique up to isomorphism, is denoted by $GF(p^n)$ or F_{p^n} and it is called the Galois field with p^n elements.

Theorem 13

Any subfield of a field $GF(p^n)$ is of the form $GF(p^m)$, where $m|n$, and vice versa.

Finite fields

Constructing $GF(p^n)$ – p is a prime and $n \geq 1$

- let f be a polynomial of degree n with coefficients in $\mathbb{Z}_p$;
- the set F of all polynomials over $\mathbb{Z}_p$ of degree at most $n-1$ has exactly p^n elements;
- If f is irreducible over $\mathbb{Z}_p[x]$, then F together with the following operations form a field:
  - the addition of two polynomial in F is the component-wise addition modulo p;
  - the multiplication of two polynomials in F is performed modulo p for coefficients and modulo f for the entire result;
  - the zero element is the zero polynomial, and the unity of F is the constant polynomial 1;
  - the additive (multiplicative) inverse exists for any polynomial (non-zero polynomial) in F.

$f(x) = a_p x^p + \dots + a_1 x + a_0 \pmod p$

Finite fields

Constructing $GF(p^n)$ – Example

- we want to construct $GF(2^8)$;
- let $f(x) = x^8 + x^4 + x^3 + x + 1 \in \mathbb{Z}_2[x]$ be an irreducible polynomial of degree 8 over $\mathbb{Z}_2[x]$;
- $GF(2^8)$ consists of all polynomial of degree at most 7 with coefficients in $\mathbb{Z}_2 = \{0, 1\}$;
- example of addition in $GF(2^8)$:

$$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2$$

- example of multiplication in $GF(2^8)$:

$$(x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1) = x^7 + x^6 + 1$$

Finite fields

Irreducible polynomials – Example

- $f(x) = x^2 + x + 1$ is irreducible over $\mathbb{Z}_2[x]$. This polynomial can be used to define $GF(2^2)$;
- $f(x) = x^3 + x^2 + x + 2$ is irreducible over $\mathbb{Z}_3[x]$. This polynomial can be used to define $GF(3^3)$;
- $f(x) = x^8 + x^4 + x^3 + x + 1 \in \mathbb{Z}_2[x]$ is irreducible over $\mathbb{Z}_2[x]$. This polynomial is used by the cryptosystem Rijndael (AES) to define $GF(2^8)$.

Pt AES $\exists$ 16 pol ired ce se pot utiliza

Outline

Definitions and examples

Ring homomorphisms

Characteristic of a ring

Finite fields

Course readings

Course readings

1. F.L. Ţiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iaşi, 2006, pag. 291–325.

SAGEMATH

Algebraic Foundations of Computer Science

Vector Spaces

Ferucio Laurențiu Țiplea

Department of Computer Science
Alexandru Ioan Cuza University of Iași
Iași 700506, Romania

e-mail: ferucio.tiplea@uaic.ro

Spring 2021

Outline

Definitions and examples

Sums and direct sums

Linear combinations

Basis and dimension

Linear transformations

Outline

Definitions and examples

Sums and direct sums

Linear combinations

Basis and dimension

Linear transformations

Definitions and examples

Definition 1

Let $(F, +, -, 0, \circ, ', e)$ be a field. A vector space over F is an algebraic system $(V, \oplus, \ominus, \mathbf{0}, \cdot)$ which consists of a commutative group $(V, \oplus, \ominus, \mathbf{0})$ and a function $\cdot : F \times V \to V$ such that:

1. $\alpha \cdot (x \oplus y) = \alpha \cdot x \oplus \alpha \cdot y$, for any $\alpha \in F$ and $x, y \in V$;
2. $(\alpha + \beta) \cdot x = \alpha \cdot x \oplus \beta \cdot x$, for any $\alpha, \beta \in F$ and $x \in V$;
3. $(\alpha \circ \beta) \cdot x = \alpha \cdot (\beta \cdot x)$, for any $\alpha, \beta \in F$ and $x \in V$;
4. $e \cdot x = x$, for any $x \in V$.

The elements of V are called vectors, the elements of F are called scalars, and F is called the field of scalars of V. The operation $\oplus$ is called the vector addition and the operation $\cdot$ is called the scalar multiplication.

Definitions and examples

Remark 1

To simplify the notation, we will denote the operations of F by $(F, +, -, 0, \cdot, ', 1)$ and the operations of V by $(V, +, -, 0, \cdot)$. Moreover, the symbol of the operation $\cdot$ will be mostly omitted. Therefore, the axioms of V can be rewritten as follows:

1. $\alpha(x + y) = \alpha x + \alpha y$, *for any* $\alpha \in F$ *and* $x, y \in V$;
2. $(\alpha + \beta)x = \alpha x + \beta x$, *for any* $\alpha, \beta \in F$ *and* $x \in V$;
3. $(\alpha\beta)x = \alpha(\beta x)$, *for any* $\alpha, \beta \in F$ *and* $x \in V$;
4. $1x = x$, *for any* $x \in V$.

Vector subtraction is defined by $x - y = x + (-y)$, for any $x, y \in V$.

The vector space which consists of the only element 0 is called the trivial vector space (it is unique up to isomorphism).

Definitions and examples

Example 2

1. Let F be a field and $n \geq 1$. Denote by F^n the set of all n-dimensional vectors over F. Define vector addition by

$$(a_1, \ldots, a_n) + (b_1, \ldots, b_n) = (a_1 + b_1, \ldots, a_n + b_n)$$

and scalar multiplication by

$$b(a_1, \ldots, a_n) = (ba_1, \ldots, ba_n),$$

for any $(a_1, \ldots, a_n), (b_1, \ldots, b_n) \in F^n$ and $b \in F$.
With these operations, F^n is a vector space over F. If we identify F^1 with F, then F can be viewed as a vector space over itself.

2. The set of all $m \times n$ matrices over F, denoted ${}^mF^n$, can be organized as a vector space over F. Vector addition is matrix addition, and scalar multiplication is the usual multiplication with scalars.

Definitions and examples

Example 3

1. $\mathbb{Q}^n$, $\mathbb{R}^n$, and $\mathbb{C}^n$ are vector spaces.
2. $\mathbb{C}$ can be viewed as a vector space over $\mathbb{R}$, and both $\mathbb{C}$ and $\mathbb{R}$ can be viewed as vector spaces over $\mathbb{Q}$.
3. The set of all functions from $\mathbb{R}$ to $\mathbb{R}$, together with the addition $f + g$ and scalar multiplication αf ($(\alpha f)(x) = \alpha f(x)$, for any x), form a vector space over $\mathbb{R}$.

Definitions and examples

Proposition 1

Let V be a vector space over a field F. Then, for any $x, y \in V$ and $\alpha, \beta \in F$, the following properties hold

1. $0x = 0$;
2. $(-1)x = -x$;
3. $(-\alpha)x = \alpha(-x) = -\alpha x$;
4. $\alpha 0 = 0$;
5. *if* $\alpha x = 0$, *then* $\alpha = 0$ *or* $x = 0$;
6. *if* $\alpha x = \alpha y$, *then* $\alpha = 0$ *or* $x = y$;
7. *if* $\alpha x = \beta x$, *then* $\alpha = \beta$ *or* $x = 0$.

Definitions and examples

Definition 4

Let V and U be vector spaces over a field F. We say that U is a subspace of V, denoted $U \leq V$, if $U \subseteq V$ and the restriction of V's operations to U coincide with U's operations.

Example 5

1. If V is a vector space over F, then $\{0\}$ and V are subspaces of V.
2. Let F be a field and $n \geq 1$. The set U of all vectors of F^n whose first coordinate is 0 is a subspace of F^n. When $n \geq 2$, this subspace can be identified with F^{n-1}.

Outline

Definitions and examples

Sums and direct sums

Linear combinations

Basis and dimension

Linear transformations

Sums and direct sums

Given V a vector space over a field F and $U_1, U_2 \subseteq V$, define

$$U_1 + U_2 = \{u_1 + u_2 \mid u_1 \in U_1,\ u_2 \in U_2\}$$

- If U_1 and U_2 are sub-spaces of V, then $U_1 + U_2$ is a sub-space of V;
- If $V = U_1 + U_2$, then V is called the *sum* of U_1 and U_2, and U_2 is called a *complement* of U_1 in V;
- If $V = U_1 + U_2$ and each non-zero vector $x \in V$ has a unique expression as a sum $x = x_1 + x_2$ of vectors in U_1 and U_2, resp., except for the order of terms, then V is called the *direct sum* of U_1 and U_2. In this case we write $V = U_1 \oplus U_2$. Moreover, U_2 is called a *direct complement* of U_1 in V;

Theorem 6

Given V a vector space over a field F and $U_1, U_2 \leq V$, $V = U_1 \oplus U_2$ iff $V = U_1 + U_2$ and $U_1 \cap U_2 = \{0\}$.

Outline

Definitions and examples

Sums and direct sums

Linear combinations

Basis and dimension

Linear transformations

Definitions and examples

Let V be a vector space over a field F, $x_1, \ldots, x_k \in V$, and $\alpha_1, \ldots, \alpha_k \in F$, where $k \geq 1$. An expression

$$\alpha_1 x_1 + \ldots + \alpha_k x_k$$

is called a linear combination of $x_1, \ldots, x_k$.

The set of all linear combinations of $x_1, \ldots, x_k$ forms a subspace of V; this subspace is called the subspace generated by $x_1, \ldots, x_k$. It is usually denoted by $\langle x_1, \ldots, x_k \rangle_V$ or $\langle x_1, \ldots, x_k \rangle$. Therefore,

$$\langle x_1, \ldots, x_k \rangle = \{\alpha_1 x_1 + \cdots + \alpha_k x_k | \alpha_1, \ldots, \alpha_k \in F\}.$$

If $x = \sum \alpha_i x_i$ then we say that x is a linear combination of $x_1, \ldots, x_k$ or that x is linearly dependent of $x_1, \ldots, x_k$.

Definitions and examples

Definition 7

Let V be a vector space over a field F. The vectors $x_1, \ldots, x_k$ from V are called linearly dependent if there exist $\alpha_1, \ldots, \alpha_k \in F$, not all 0, such that $\sum \alpha_i x_i = 0$.

If $x_1, \ldots, x_k$ are not linearly dependent, then they are called linearly independent. That is, $x_1, \ldots, x_k$ are linearly independent if for any $\alpha_1, \ldots, \alpha_k \in F$, the relation $\sum \alpha_i x_i = 0$ leads to $\alpha_1 = \cdots = \alpha_k = 0$.

Remark 2

Let V be a vector space over a field F.

1. *$x \in V$ is linearly independent iff $x \neq 0$.*
2. *If $x_1, \ldots, x_k \in V$ are linearly independent, then $x_i \neq 0$, for any i. Moreover, $x_i \neq x_j$, for any $i \neq j$.*

Definitions and examples

Proposition 2

Let V be a vector space over a field F. $x_1, \ldots, x_k$ from V are linearly dependent iff there exists $1 \leq i \leq k$ such that x_i is a linear combination of the other vectors.

Outline

Definitions and examples

Sums and direct sums

Linear combinations

Basis and dimension

Linear transformations

Basis and dimension

Definition 8

Let V be a non-trivial vector space over a field F. A finite subset $B \subseteq V$ is called a basis of V if it is linearly independent and generates V (each element in V is a linear combination of vectors in B).

Remark 3

- *If $x_1, \ldots, x_k$ form a basis for V, then $x_i \neq x_j$, for any $i \neq j$. Therefore, $\{x_1, \ldots, x_k\}$ has exactly k vectors.*
- *We have considered only finite basis. There are approaches for infinite basis too.*

Basis and dimension

Example 9

1. Let F be a field and $n \geq 1$. The vector space F^n can be generated by

$$\begin{array}{rcl} \mathbf{e_1} & = & (1,0,0,\ldots,0,0) \\ \mathbf{e_2} & = & (0,1,0,\ldots,0,0) \\ & \ldots & \\ \mathbf{e_n} & = & (0,0,0,\ldots,0,1). \end{array}$$

2. Let F be a field and $m, n \geq 1$. The vector space ${}^mF^n$ can be generated by E_{ij}, where

$$E_{ij}(u,v) = \begin{cases} 1, & \text{if } u = i \text{ and } v = j \\ 0, & \text{otherwise,} \end{cases}$$

for any $i, u \in \{1,\ldots,m\}$ and $v, j \in \{1,\ldots,n\}$.

Basis and dimension

Theorem 10

Let V be a vector space over a field F. $B = \{x_1, \ldots, x_k\} \subseteq V$ is a basis of V iff any $x \in V$ can be uniquely written as a linear combination of vectors in B.

Given a basis $B = \{x_1, \ldots, x_n\}$ of V, each element $x \in V$ can be uniquely written as

$$x = \alpha_1 x_1 + \cdots + \alpha_n x_n,$$

with $\alpha_1, \ldots, \alpha_n \in F$. We will say that α_i in the *i-th coordinate of* x, and $(\alpha_1, \ldots, \alpha_n)$ is the *coordinate of* x, in the basis B.

Basis and dimension

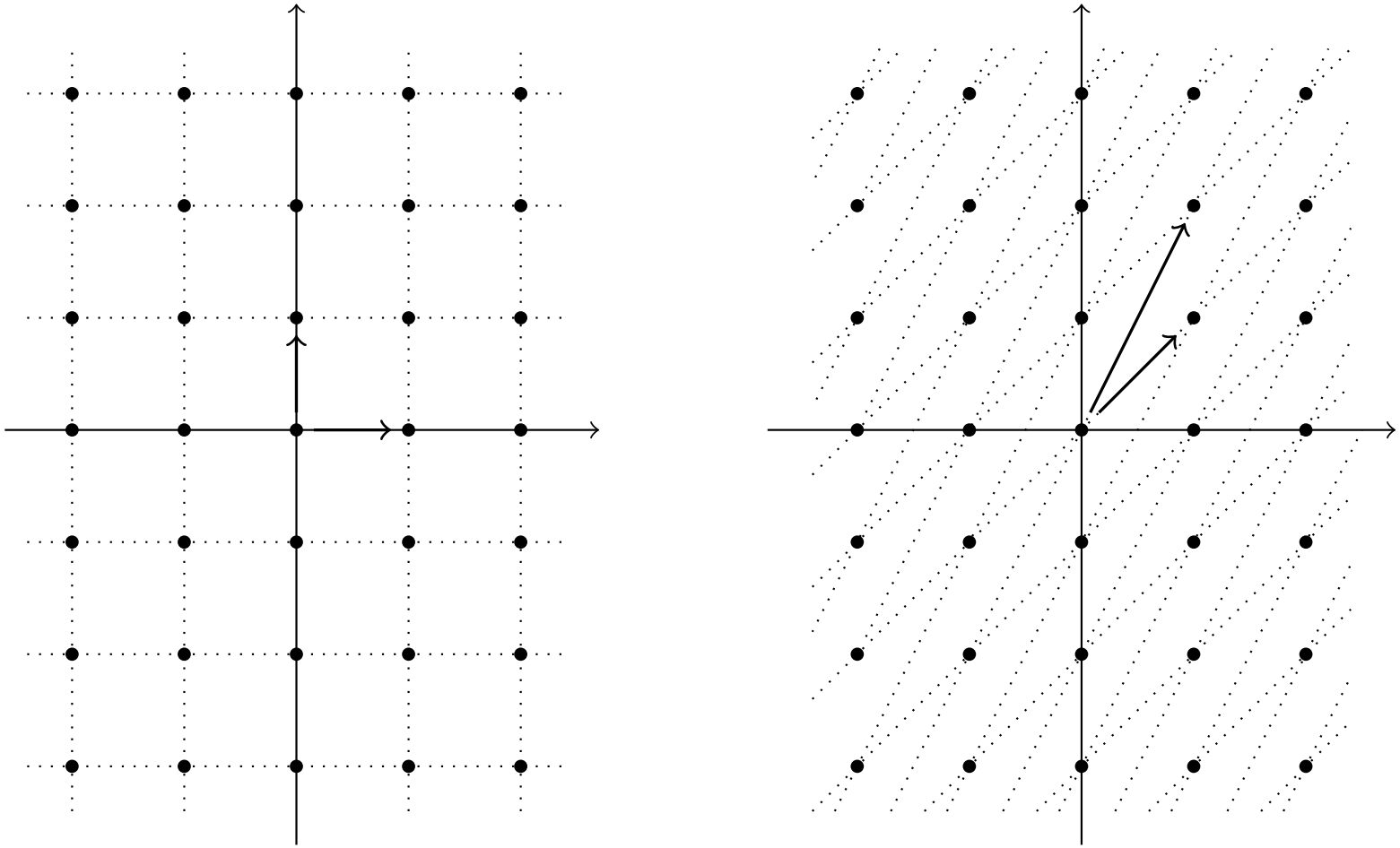


Figure: Generic examples of bases

Basis and dimension

Lemma 11 (Steinitz Exchange Lemma)

Let V be a vector space over a field F. If V is generated by $A = \{x_1, \ldots, x_k\}$ and $B \subseteq \{x_1, \ldots, x_m\}$ is a subset of linearly independent vectors, then $m \leq k$. Moreover, some vectors from A can be replaced by vectors from B so that the set thus obtained contains exactly k vectors, still generates V, and includes B.

Corollary 12

If A and B are finite linearly independent sets that generate a vector space V, then $|A| = |B|$.

Basis and dimension

Definition 13

Let V be a vector space over a field F. V is called finite dimensional if there exists a (finite) basis B for V. In this case, $|B|$ is called the dimension of V, denoted $dim(V)$. If V is not finite dimensional then it is called infinite dimensional.

Example 14

1. $dim(F^n) = n$;
2. $dim({}^mF^n) = mn$;
3. $F^{\mathbb{N}}$ is infinite dimensional.

Basis and dimension

Theorem 15

Let V be a vector space over a field F and $U_1, U_2 \leq V$. Then,

$$dim(U_1) + dim(U_2) = dim(U_1 + U_2) + dim(U_1 \cap U_2)$$

In particular, if U_2 is the complement of U_1 in V, then

$$dim(U_1) + dim(U_2) = dim(V)$$

Outline

Definitions and examples

Sums and direct sums

Linear combinations

Basis and dimension

Linear transformations

Linear transformation

Definition 16

Let U and V be vector spaces over a field F. A linear transformation/map from U to V is a mapping $f : U \to V$ such that

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y),$$

for all $\alpha, \beta \in F$ and $x, y \in U$.

Proposition 3

If U and V are two vector spaces of the same (finite) dimension over a field F, then U and V are isomorphic.

Linear transformation

Let U and V be vector spaces over a field F, with bases $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_m\}$, respectively.
Any linear map $f : U \to V$ is completely specified by a matrix

$$A_f = \begin{array}{c} \\ y_1 \\ \vdots \\ y_m \end{array} \begin{array}{c} \begin{array}{ccc} f(x_1) & \cdots & f(x_n) \end{array} \\ \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \vdots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} \end{array}$$

where $f(x_j) = a_{1,j}y_1 + \cdots + a_{m,j}y_m$, for all j. Therefore, if

$$x = \alpha_1 x_1 + \cdots + \alpha_n x_n,$$

then

$$f(x) = \alpha_1 f(x_1) + \cdots + \alpha_n f(x_n) = \begin{pmatrix} f(x_1) & \cdots & f(x_n) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = A_f x$$

Linear transformation

Let U and V be vector spaces over a field F. Denote by $Hom_F(U, V)$ the set of all linear maps from U to V.
$Hom_F(U, V)$ can be structured as a vector space over F, considering:

- $(f + g)(x) = f(x) + g(x)$;
- $(\alpha f)(x) = \alpha f(x)$,

for any $f, g \in Hom_F(U, V)$, $x \in U$, and $\alpha \in F$.

Proposition 4

Let U and V be vector spaces over a field F with $dim(U) = m$ and $dim(V) = n$. Then, $Hom_F(U, V)$ is isomorphic to ${}^mF^n$.