AFCS / Spring 2015

Semigroups and Monoids

Prof.Dr. Ferucio Laurenţiu Ţiplea

"Al. I. Cuza" University of Iaşi

Department of Computer Science

Iasi 740083, Romania

E-mail: fltiplea@mail.dntis.ro

URL: http://www.infoiasi.ro/~fltiplea

Contents

1. 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 1

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.

1. Definitions and examples

Definition 2 A semigroup $(S, \circ)$ is called commutative if $\circ$ is a commutative operation.

Example 2 All semigroups in Example 1 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.

1. Definitions and examples

Definition 3 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.

1. Definitions and examples

Definition 4 A monoid $(M, \circ, e)$ is called commutative if its binary operation $\circ$ is commutative.

Example 3

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 $S_M = M - \{e\}$.

1. Definitions and examples

Definition 5 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 4 The principal ideal of $(\mathbb{Z}, \cdot, 1)$ generated by $n \in \mathbb{Z}$ is $n\mathbb{Z}$.

1. Definitions and examples

Definition 6

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 6(3)(4) is called a set of generators and its elements are called generators.

1. 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.

1. Definitions and examples

Example 5

- $(\mathbb{N}, +) \leq (\mathbb{Z}, +) \leq (\mathbb{Q}, +) \leq (\mathbb{R}, +)$;
- $(\mathbb{N}, +, 0) \leq (\mathbb{Z}, +, 0) \leq (\mathbb{Q}, +, 0) \leq (\mathbb{R}, +, 0)$;
- $(\mathbb{N}, \cdot) \leq (\mathbb{Z}, \cdot) \leq (\mathbb{Q}, \cdot) \leq (\mathbb{R}, \cdot)$;
- $(\mathbb{N}, \cdot, 1) \leq (\mathbb{Z}, \cdot, 1) \leq (\mathbb{Q}, \cdot, 1) \leq (\mathbb{R}, \cdot, 1)$;
- 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 $\{-3, 2\}$.

1. Definitions and examples

Definition 7

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 6

- $(\mathbb{Z}, +, 0)$ 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 ∞.

1. Definitions and examples

Definition 8

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;

1. 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 7

- 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).

2. Word semigroups

Definition 9 An alphabet is any non-empty set. The elements of an alphabet are called letters or symbols.

Example 8 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.

2. Word semigroups

Definition 10 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 9

- $w = abaa$ 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 $\Sigma_3 = \{\texttt{begin}, \texttt{end}, \texttt{if}, \texttt{then}, \texttt{else}, \texttt{while}, \texttt{do}\}$.

2. 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 11 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 $1 \leq i \leq k$.

2. Word semigroups

Definition 12 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 10

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

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

2. Word semigroups

Theorem 1 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.

2. Word semigroups

Theorem 2 (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 13 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.

2. Word semigroups

Definition 14

(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.

2. Word semigroups

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

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

3. Cyclic semigroups

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

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

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

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

Definition 15 A semigroup (monoid) generated by one of its elements is called a cyclic semigroup (cyclic monoid).

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

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

for some $a \in S$.

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

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

for some $a \in M$.

3. Cyclic semigroups

Theorem 3 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 3(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

4. 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 be uniquely written as a finite combination of 1's under $+$, namely,

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

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

Definition 16 A semigroup $(S, \circ)$ is freely generated by a subset $X \subseteq S$ if any element of $s \in S$ can be uniquely 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 $n \geq 1$.

4. Free semigroups and monoids

Definition 17 A monoid $(M, \circ, e)$ is freely generated by a subset $X \subseteq M$ if $(S_M, \circ)$ is freely generated by X.

Definition 18 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 11

- $(\mathbb{N}, +, 0)$ 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.

4. Free semigroups and monoids

Free semigroups (monoids) have very important properties.

Theorem 4 (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 1 Any free semigroup (monoid) is isomorphic with a word semigroup (monoid).

4. 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 12 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 $n \geq 0$.

4. 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 $S - S^2$ ($S_M - S_M^2$). $\square$

5. Variable-length codes

Definition 19 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
$$u_1 \cdots u_m = v_1 \cdots v_n \;\Rightarrow\; n = m \;\wedge\; (\forall i)(u_i = v_i),$$
for any $u_1, \ldots, u_m, v_1, \ldots, v_n \in C$;
3. C is a code over A if
$$u_1 \cdots u_m = v_1 \cdots v_n \;\Rightarrow\; u_1 = v_1,$$
for any $u_1, \ldots, u_m, v_1, \ldots, v_n \in C$.

5. Variable-length codes

Example 13

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

Definition 20

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 14

- ASCII is a block code.

5. 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 5 (Sardinas-Patterson Theorem)
C is a code over A iff $C \cap C_i = \emptyset$, for any $i \geq 1$.

5. 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.

6. Huffman codes

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.

6. Huffman codes

Definition 21 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 22 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 defined by

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

Definition 23 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.

6. Huffman codes

Example 15 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.

6. Huffman codes

Definition 24 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.

6. Huffman codes

Huffman algorithm:

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

$$\begin{array}{c|ccccc} A & a_1 & a_2 & \cdots & a_{n-1} & a_n \\ \hline \pi & p_1 & p_2 & \cdots & p_{n-1} & p_n \end{array}$$

   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

$$\begin{array}{c|ccccc} A' & a_1 & a_2 & \cdots & a_{n-2} & a_{n-1,n} \\ \hline \pi' & p_1 & p_2 & \cdots & p_{n-2} & p_{n-1,n} \end{array}$$

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

6. Huffman codes

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

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

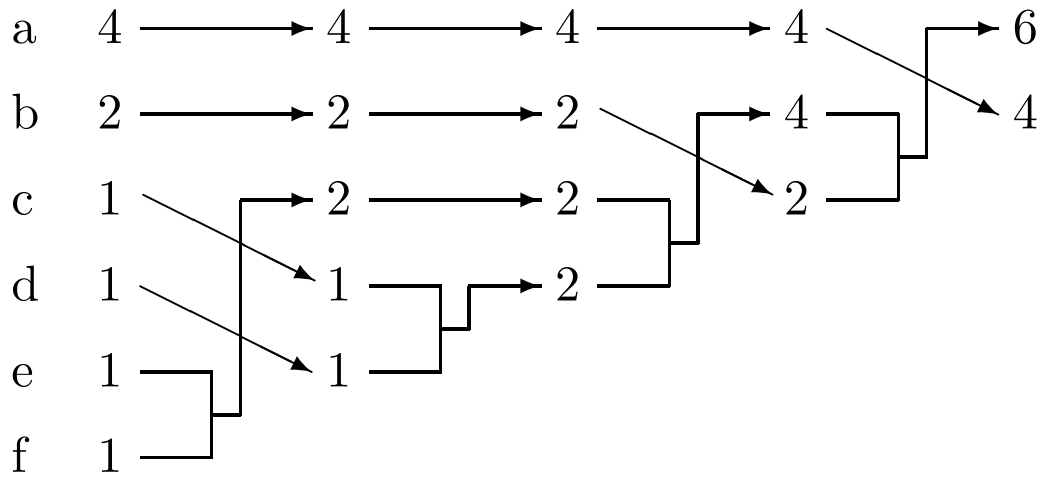
is a Huffman code for IS.

6. Huffman codes

Example 16 Let IS be the following information source:

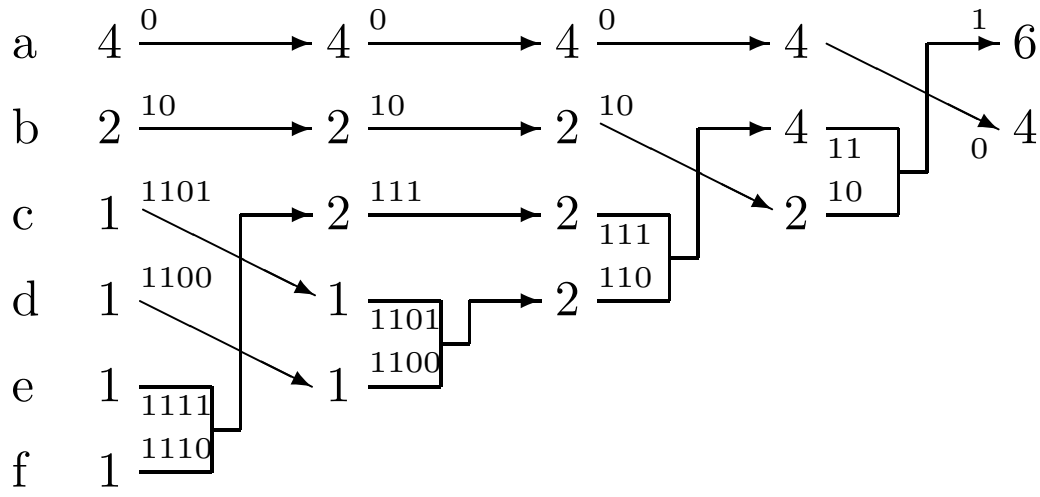
A	a	b	c	d	e	f
π	0.4	0.2	0.1	0.1	0.1	0.1

Compute a sequence of reduced sources for IS:



6. 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.

6. 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).

6. 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.

6. Huffman codes

Definition 25 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.

6. Huffman codes

Definition 26 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 $\underbrace{S \circ \cdots \circ S}_{k\ times}$ by S^k, where S is a finite information source.

Then,

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

6. Huffman codes

Theorem 6 (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.

6. 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

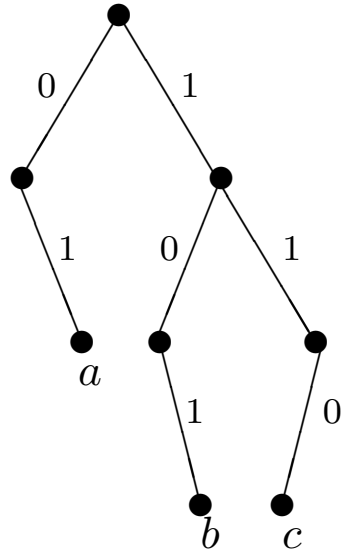
$$p_a = \frac{\text{frequency of } a \text{ in } w}{|w|}.$$

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.

6. 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.



6. 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 $\mathcal{T}_{n+1}$.

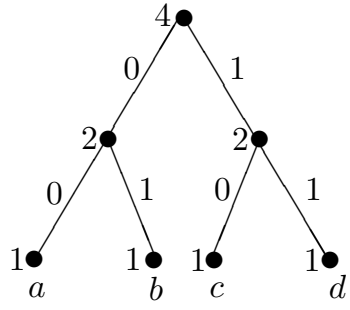
6. Huffman codes

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

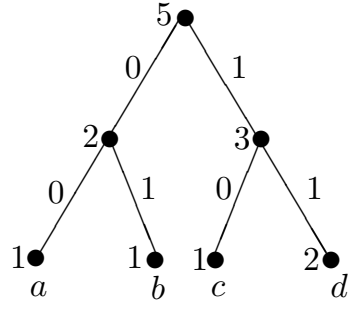
1. compare a to its successors in the tree (from left to right and from bottom to top). If the immediate successor has frequency $k+1$ or greater, where k is the frequency of a in $\mathcal{T}_n$, then the nodes are still in sorted order and there is no need to change anything. Otherwise, a should be 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.

6. Huffman codes

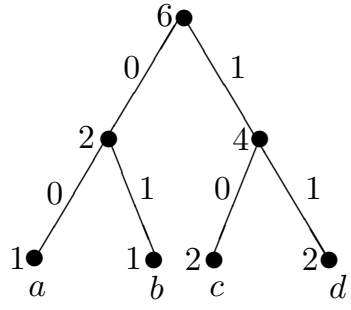
A sequence of Huffman trees used to encode the string dcd over the alphabet $\{a, b, c, d\}$:



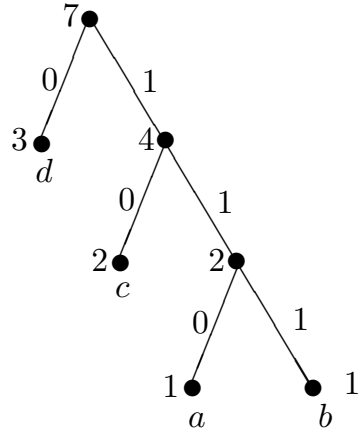
(a) T_0



(b) T_1



(d) T_2



(c) T_3

6. 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.

7. 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.