Introduction to Computer Science

Data representation

Piotr Fulmański

Faculty of Mathematics and Computer Science,
University of Łódź, Poland

November 3, 2010

Table of Contents

Remember!

- All information processed by computer systems is only a sequence of zeros and ones. Nothing more.
- In a system point of view ALL information is just a stream of zeros and ones.
- The same sequence of zeros and ones can be a photo of our friend, another time our favourite MP3 and another time a letter to uncle.
- We are, the user, who decide how to interprete given sequence of zeros and ones.
- What we have (read) depend on the way of interpretation.
- It is not a graphic file which inform us, that it is a graphic file but that we are, who interpret file as it is a graphic file.[a]

[a] Of course most of modern files enclose information about carried data but this is only to help us.

Remember!

- **All information processed by computer systems is only a sequence of zeros and ones. Nothing more.**
- In a system point of view ALL information is just a stream of zeros and ones.
- The same sequence of zeros and ones can be a photo of our friend, another time our favourite MP3 and another time a letter to uncle.
- We are, the user, who decide how to interprete given sequence of zeros and ones.
- What we have (read) depend on the way of interpretation.
- It is not a graphic file which inform us, that it is a graphic file but that we are, who interpret file as it is a graphic file.[a]

[a] Of course most of modern files enclose information about carried data but this is only to help us.

Remember!

- **All information processed by computer systems is only a sequence of zeros and ones. Nothing more.**
- **In a system point of view ALL information is just a stream of zeros and ones.**
- The same sequence of zeros and ones can be a photo of our friend, another time our favourite MP3 and another time a letter to uncle.
- We are, the user, who decide how to interprete given sequence of zeros and ones.
- What we have (read) depend on the way of interpretation.
- It is not a graphic file which inform us, that it is a graphic file but that we are, who interpret file as it is a graphic file.[a]

[a] Of course most of modern files enclose information about carried data but this is only to help us.

Remember!

- **All information processed by computer systems is only a sequence of zeros and ones. Nothing more.**
- **In a system point of view ALL information is just a stream of zeros and ones.**
- **The same sequence of zeros and ones can be a photo of our friend, another time our favourite MP3 and another time a letter to uncle.**
- **We are, the user, who decide how to interprete given sequence of zeros and ones.**
- **What we have (read) depend on the way of interpretation.**
- **It is not a graphic file which inform us, that it is a graphic file but that we are, who interpret file as it is a graphic file.**[a]

[a] Of course most of modern files enclose information about carried data but this is only to help us.

Remember!

- **All information processed by computer systems is only a sequence of zeros and ones. Nothing more.**
- **In a system point of view ALL information is just a stream of zeros and ones.**
- **The same sequence of zeros and ones can be a photo of our friend, another time our favourite MP3 and another time a letter to uncle.**
- **We are, the user, who decide how to interprete given sequence of zeros and ones.**
- **What we have (read) depend on the way of interpretation.**
- **It is not a graphic file which inform us, that it is a graphic file but that we are, who interpret file as it is a graphic file.**[a]

[a] Of course most of modern files enclose information about carried data but this is only to help us.

Remember!

- All information processed by computer systems is only a sequence of zeros and ones. Nothing more.
- In a system point of view ALL information is just a stream of zeros and ones.
- The same sequence of zeros and ones can be a photo of our friend, another time our favourite MP3 and another time a letter to uncle.
- We are, the user, who decide how to interprete given sequence of zeros and ones.
- What we have (read) depend on the way of interpretation.
- It is not a graphic file which inform us, that it is a graphic file but that we are, who interpret file as it is a graphic file.[a]

[a] Of course most of modern files enclose information about carried data but this is only to help us.

Remember!

- All information processed by computer systems is only a sequence of zeros and ones. Nothing more.
- In a system point of view ALL information is just a stream of zeros and ones.
- The same sequence of zeros and ones can be a photo of our friend, another time our favourite MP3 and another time a letter to uncle.
- We are, the user, who decide how to interprete given sequence of zeros and ones.
- What we have (read) depend on the way of interpretation.
- It is not a graphic file which inform us, that it is a graphic file but that we are, who interpret file as it is a graphic file.[a]

[a] Of course most of modern files enclose information about carried data but this is only to help us.

Some analogies

Linguistic analogy

What does it mean: ***para***

- if we know, that this is Polish word: ***two objects***;
- if we know, that this is Spanish word: ***for***.

Numeral analogy

Number ***eight*** can be written as

- ***8*** in decimal numeral system;
- ***VIII*** in roman numeral system;
- ***1000*** in binary numeral system.

Some analogies

Linguistic analogy

What does it mean: ***para***

- if we know, that this is Polish word: ***two objects***;
- if we know, that this is Spanish word: ***for***.

Numeral analogy

Number ***eight*** can be written as

- ***8*** in decimal numeral system;
- ***VIII*** in roman numeral system;
- ***1000*** in binary numeral system.

Some analogies

Linguistic analogy

What does it mean: ***para***

- if we know, that this is Polish word: ***two objects***;
- if we know, that this is Spanish word: ***for***.

Numeral analogy

Number ***eight*** can be written as

- ***8*** in decimal numeral system;
- ***VIII*** in roman numeral system;
- ***1000*** in binary numeral system.

Some analogies

Linguistic analogy

What does it mean: ***para***

- if we know, that this is Polish word: ***two objects***;
- if we know, that this is Spanish word: ***for***.

Numeral analogy

Number ***eight*** can be written as

- ***8*** in decimal numeral system;
- ***VIII*** in roman numeral system;
- ***1000*** in binary numeral system.

Some analogies

Linguistic analogy

What does it mean: ***para***

- if we know, that this is Polish word: ***two objects***;
- if we know, that this is Spanish word: ***for***.

Numeral analogy

Number ***eight*** can be written as

- ***8*** in decimal numeral system;
- ***VIII*** in roman numeral system;
- ***1000*** in binary numeral system.

Some analogies

Linguistic analogy

What does it mean: ***para***

- if we know, that this is Polish word: ***two objects***;
- if we know, that this is Spanish word: ***for***.

Numeral analogy

Number ***eight*** can be written as

- ***8*** in decimal numeral system;
- ***VIII*** in roman numeral system;
- ***1000*** in binary numeral system.

What we code?

What type of coding we will be talking about

- alphanumeric
- natural numbers
- integer numbers
- real numbers
- graphic file `bmp`
- TCP/IP packet

Alphanumeric

Alphanumeric is any letter, digit and some symbols like (, :, + etc., everything what we can write (input) using keyboard.

Coding

Coding is a process of changing character enteredy by keyboard or any input devices into digital representation, that is storing this character as a sequence of zeros and ones.

Alphanumeric

Alphanumeric is any letter, digit and some symbols like (, :, + etc., everything what we can write (input) using keyboard.

Coding

Coding is a process of changing character enteredy by keyboard or any input devices into digital representation, that is storing this character as a sequence of zeros and ones.

ASCII

ASCII American Standard Code for Information Interchange. In this type of coding there are codes defined for

- small (97-122) and capital (65-90) latin alphabet;
- digits (48-57);
- some symbols like (, :, + etc. (32-47, 58-64, 91-96, 123-126);
- nonprintable characters controling data flow, e.g. `ACK` – acknowledge or `BEL` – bell (sound signal) (0-31).

Alphanumeric

Good start...

The scope of ASCII codes stretch from 0 to 127. That means that we need at least 7 bits to write all this numbers. Because most computers in those days were using 8-bit bytes (that is, computers divides and process all information in 8-bit portions) so there were 128 spare place coded from 128 to 255.

ASCII characters did not cover the needs of nationality using latin characters with dots (Germany, Poland) or completely nonstandard characters (Greece, Russia).

Because of resultant needs, to represent dotted charactes this 128 spare numbers were used.

Alphanumeric

Good start...

The scope of ASCII codes stretch from 0 to 127. That means that we need at least 7 bits to write all this numbers. Because most computers in those days were using 8-bit bytes (that is, computers divides and process all information in 8-bit portions) so there were 128 spare place coded from 128 to 255.

ASCII characters did not cover the needs of nationality using latin characters with dots (Germany, Poland) or completely nonstandard characters (Greece, Russia).

Because of resultant needs, to represent dotted charactes this 128 spare numbers were used.

Alphanumeric

Good start...

The scope of ASCII codes stretch from 0 to 127. That means that we need at least 7 bits to write all this numbers. Because most computers in those days were using 8-bit bytes (that is, computers divides and process all information in 8-bit portions) so there were 128 spare place coded from 128 to 255.

ASCII characters did not cover the needs of nationality using latin characters with dots (Germany, Poland) or completely nonstandard characters (Greece, Russia).

Because of resultant needs, to represent dotted charactes this 128 spare numbers were used.

The trouble was, lots of people had this idea at the same time, and they had their own ideas of what should go where in the space from 128 to 255.

This is how **code pages** come into being. Code pages – a sets of 255 characters with shared firs half of characters and sometimes completely different second half.

This is why when manipulate any text, if we want correctly read nonstandard characters, we HAVE TO know code page used during coding.

The trouble was, lots of people had this idea at the same time, and they had their own ideas of what should go where in the space from 128 to 255.

This is how **code pages** come into being. Code pages – a sets of 255 characters with shared firs half of characters and sometimes completely different second half.

This is why when manipulate any text, if we want correctly read nonstandard characters, we HAVE TO know code page used during coding.

The trouble was, lots of people had this idea at the same time, and they had their own ideas of what should go where in the space from 128 to 255.

This is how **code pages** come into being. Code pages – a sets of 255 characters with shared firs half of characters and sometimes completely different second half.

This is why when manipulate any text, if we want correctly read nonstandard characters, we HAVE TO know code page used during coding.

Standard	ą	ć	ę	ł	ń	ó	ś	ż	ź
ISO 8859-2	B1	E6	EA	B3	F1	F3	B6	BF	BC
CP 1250	B9	E6	EA	B3	F1	F3	9C	BF	9F
Mazowia	86	8D	91	92	A4	A2	9E	A7	A6
Unicode	105	107	119	142	144	F3	15B	17C	17A

- ISO 8859-2, also known as latin2, typical for UNIX like systems.
- CP 1250, also known as win-1250, typical for Windows family systems.
- Mazowia – coding prepared for Polish computer Mazovia.
- Unicode – we will see soon.

Problems

- Obvious – many various code pages even for the same language.
- Difficulties with processing multilingua text.
- The scope of codes (255) was to small for some languages, e.g. Chinese.

Unicode – most importants facts

Explicitness. One code for fon characte and vice versa.

Universality. All well (and not well) known languages and symbols.

Effectiveness. Character identification does not depend on control sequence and preceding or following characters.

Identification not representation. What character but not how it look.

Sens/property. Characters properties (e.g. alphabetic order) are not depend on position in codes table but are define in properties table.

Plain text.

Logic order.

Unification. Identical characters for different meaning were replaced by one.

Kod FOO

Przyjmujemy następujący sposób kodowania znaków alfanumerycznych

a	b	c	d	e	f	g	h	i	j	k	l	m	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13
o	p	q	r	s	t	u	v	w	x	y	z	0	1
14	15	16	17	18	19	20	21	22	23	24	25	26	27
2	3	4	5	6	7	8	9	,	.	(	)	-	’
28	29	30	31	32	33	34	35	36	37	38	39	40	41

Znak „spacji” posiada kod 42.

Kod FOO

Dodatkowo wprowadzamy następujące sekwencje sterujące:

- `ESC1` (kod 43) służącą do zamiany litery małej występującej zaraz za sekwencją na dużą.
- `ESC2` (kod 44) służącą do uzyskania znaków diakrytycznych. Sekwencja
  - `ESC2 , litera` dodaje „ogonek” do litery,
  - `ESC2 . litera` dodaje „kropkę” do litery,
  - `ESC2 - litera` dodaje „przekreślenie” do litery,
  - `ESC2 ' litera` dodaje „kreskę nad” do litery.
- `NL` (kod 45) powodujący przejście do nowego wiersza.

Kod FOO

Jak widać w kodzie FOO liczba znaków jest mniejsza niż 64 ale większ niż 32. Stąd wniosek, że musimy używać co najmniej 6 bitów do zapisania kodów.
Spróbujmy zakodować następujące zdanie:

Miała (kiedyś) Zośka 371 kotów a teraz ma 1 szczura - „Mańka".

Kod FOO

Zdanie

Miała (kiedyś) Zośka 371 kotów a teraz ma 1 szczura - „Mańka”.

zapiszemy przy pomocy następującej sekwencji kodów:

43,12	M	101011,001100
8	i	001000
0	a	000000
44,40,11	ł	101100,101000,001011
0	a	000000
42		101010
38	(	100110
10	k	001010

Ciąg dalszy przykładu w skrypcie (do pobrania na stronie).

Natural numbers coding

To code natural numbers we use a well knonw from previous lectures method of coding decimal numbers as binary numbers.

Integer numbers

Integer numbers coding

- sign-and-magnitude (sign and absolute value)
- two's complement (U2)

Two's complement

Two's complement of some number x is calculate by formula $x_{U2} = 2^n - x$, where n is a number of bits used for coding x number.

Integer numbers coding

- sign-and-magnitude (sign and absolute value)
- two's complement (U2)

Two's complement

Two's complement of some number x is calculate by formula $x_{U2} = 2^n - x$, where n is a number of bits used for coding x number.

Real numbers coding

- fixed point real numbers
- floating point real numbers

Floating point real numbers

$$z_m M \cdot 2^{z_c C}$$

Floating point – example

Assumptions

- to represent a number we use 8 bits;
- the first bit from the left (7) is a sign of the number;
- bits (6-4) means mantissa;
- bits (3-0) means fractional (characteristic);
- constant K_C is equal to 7.

Real numbers coding

- fixed point real numbers
- floating point real numbers

Floating point real numbers

$$z_m M \cdot 2^{z_c C}$$

Floating point – example

Assumptions

- to represent a number we use 8 bits;
- the first bit from the left (7) is a sign of the number;
- bits (6-4) means mantissa;
- bits (3-0) means fractional (characteristic);
- constant K_C is equal to 7.

Real numbers coding

- fixed point real numbers
- floating point real numbers

Floating point real numbers

$$z_m M \cdot 2^{z_c C}$$

Floating point – example

Assumptions

- to represent a number we use 8 bits;
- the first bit from the left (7) is a sign of the number;
- bits (6-4) means mantissa;
- bits (3-0) means fractional (characteristic);
- constant K_C is equal to 7.

Graphic file bmp

TCP/IP packet