

Représenter un texte

Yann Strozecki
yann.strozecki@uvsq.fr

Université de Versailles St-Quentin-en-Yvelines

Année universitaire 2022-2023

Théorie des langages allégée

On veut représenter des mots sur ordinateur. Qu'est-ce qu'un mot formellement ?

Théorie des langages allégée

On veut représenter des **mots** sur ordinateur. Qu'est-ce qu'un mot formellement ?

On a un **alphabet** fini : $\Sigma = \{a, b, c, d\}$.

C'est un ensemble de *symboles*.

Théorie des langages allégée

On veut représenter des **mots** sur ordinateur. Qu'est-ce qu'un mot formellement ?

On a un **alphabet** fini : $\Sigma = \{a, b, c, d\}$.

C'est un ensemble de *symboles*.

On crée des mots par **concaténation** de symboles : *abacadaba*.

Un mot est une *suite finie* de symboles de l'alphabet.

Exemple : Les nombres entiers positifs sont des mots, dont les symboles sont les chiffres.

Théorie des langages allégée

On veut représenter des **mots** sur ordinateur. Qu'est-ce qu'un mot formellement ?

On a un **alphabet** fini : $\Sigma = \{a, b, c, d\}$.

C'est un ensemble de *symboles*.

On crée des mots par **concaténation** de symboles : *abacadaba*.

Un mot est une *suite finie* de symboles de l'alphabet.

Exemple : Les nombres entiers positifs sont des mots, dont les symboles sont les chiffres.

Σ^k est l'ensemble des mots de k lettres sur Σ . Σ^* est l'ensemble de tous les mots sur Σ .

Exemple : $abc \in \Sigma^3$ et $ab \notin \Sigma^4$.

Notre objectif est de représenter toute information par un mot de $\{0, 1\}^*$.

Encodage de mots et décodage

Un **codage** est une fonction **injective** de $\Sigma \rightarrow \{0, 1\}^*$.

Par exemple $E(a) = 00, E(b) = 01, E(c) = 10, E(d) = 11$.

Encodage de mots et décodage

Un **codage** est une fonction **injective** de $\Sigma \rightarrow \{0, 1\}^*$.

Par exemple $E(a) = 00, E(b) = 01, E(c) = 10, E(d) = 11$.

On **étend** cette fonction en une fonction de $\Sigma^* \rightarrow \{0, 1\}^*$ par concaténation :

$$E(w_1 w_2 \dots w_k) = E(w_1) E(w_2) \dots E(w_k)$$

Par exemple $E(abacadaba) = 000100100011000100$.

Décodage

L'application donnée par

$E(a) = 0, E(b) = 10, E(c) = 01, E(d) = 110$, n'est **pas acceptable** pour définir un codage.

Ambiguïté le codage de ab donne 010 et le codage de ca également.

Décodage

L'application donnée par

$E(a) = 0, E(b) = 10, E(c) = 01, E(d) = 110$, n'est **pas acceptable** pour définir un codage.

Ambiguïté le codage de ab donne 010 et le codage de ca également.

Pour résoudre ce problème, on utilise des codages à longueur fixe.
Proposer un tel codage pour $\{a, b, c, d, e\}$. Est-il efficace en taille ?

Codage et communication

L'ancêtre de tous les "codages binaires", le *Morse* International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A • ■■
B ■■ • •
C ■■ • ■■ •
D ■■ • •
E •
F • • ■■ •
G ■■ ■■ •
H • • • •
I • •
J • ■■ ■■ ■■
K ■■ • ■■
L • ■■ • •
M ■■ ■■
N ■■ •
O ■■ ■■ ■■
P • ■■ ■■ •
Q ■■ ■■ • ■■
R • ■■ •
S • • •
T ■■

U • • ■■
V • • • ■■
W • ■■ ■■
X ■■ • • ■■
Y ■■ • ■■ ■■
Z ■■ ■■ • •

1 • ■■ ■■ ■■ ■■
2 • • ■■ ■■ ■■
3 • • • ■■ ■■
4 • • • • ■■
5 • • • • •
6 ■■ • • • •
7 ■■ ■■ • • •
8 ■■ ■■ ■■ • •
9 ■■ ■■ ■■ ■■ •
0 ■■ ■■ ■■ ■■ ■■

Codage et communication

L'ancêtre de tous les "codages binaires", le *Morse* International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A • —
B — • • •
C — • — •
D — • •
E •
F • • — •
G — — •
H • • • •
I • •
J • — — —
K — • —
L • — • •
M — —
N — •
O — — —
P • — — •
Q — — • —
R • — •
S • • •
T —

U • • —
V • • • —
W • — —
X — • • —
Y — • — —
Z — — • •

1 • — — — —
2 • • — — —
3 • • • — —
4 • • • • —
5 • • • • •
6 — • • • •
7 — — • • •
8 — — — • •
9 — — — — •
0 — — — — —

Le mot S.O.S. se note :

... — — — ...

Codage et communication

L'ancêtre de tous les "codages binaires", le *Morse* International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A • —
B — • • •
C — • — •
D — • •
E •
F • • — •
G — — •
H • • • •
I • •
J • — — —
K — • —
L • — • •
M — —
N — •
O — — —
P • — — •
Q — — • —
R • — •
S • • •
T —

U • • —
V • • • —
W • — —
X — • • —
Y — • — —
Z — — • •

1 • — — — —
2 • • — — —
3 • • • — —
4 • • • • —
5 • • • • •
6 — • • • •
7 — — • • •
8 — — — • •
9 — — — — •
0 — — — — —

Le mot S.O.S. se note :

... — — — ...

Code de longueur **variable**.

La taille des caractères dépend de leur fréquence.

Codage et communication

L'ancêtre de tous les "codages binaires", le *Morse* International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A • —
B — • • •
C — • — •
D — • •
E •
F • • — •
G — — •
H • • • •
I • •
J • — — —
K — • —
L • — • •
M — —
N — •
O — — —
P • — — •
Q — — • —
R • — •
S • • •
T —

U • • —
V • • • —
W • — —
X — • • —
Y — • — —
Z — — • •

1 • — — — —
2 • • — — —
3 • • • — —
4 • • • • —
5 • • • • •
6 — • • • •
7 — — • • •
8 — — — • •
9 — — — — •
0 — — — — —

Le mot S.O.S. se note :

... — — — ...

Code de longueur **variable**.

La taille des caractères dépend de leur fréquence.

Non ambigu car il y a un *troisième* symbole, le silence.

Le codage ASCII

Le **code ASCII** représente chaque caractère sur 7 bits, c'est un code de **taille fixe**. À chaque caractère est associée une configuration de 8 bits (1 octet), le bit de poids fort (le plus à gauche) est toujours égal à zéro.

Le codage ASCII

Le **code ASCII** représente chaque caractère sur 7 bits, c'est un code de **taille fixe**. À chaque caractère est associée une configuration de 8 bits (1 octet), le bit de poids fort (le plus à gauche) est toujours égal à zero.

- Les codes compris entre 0 et 31 sont des **caractères de contrôle**. Ils sont utilisés pour indiquer des actions telles que passer à la ligne (CR, LF), émettre un bip sonore (BEL), etc.

Le codage ASCII

Le **code ASCII** représente chaque caractère sur 7 bits, c'est un code de **taille fixe**. À chaque caractère est associée une configuration de 8 bits (1 octet), le bit de poids fort (le plus à gauche) est toujours égal à zero.

- ▶ Les codes compris entre 0 et 31 sont des **caractères de contrôle**. Ils sont utilisés pour indiquer des actions telles que passer à la ligne (CR, LF), émettre un bip sonore (BEL), etc.
- ▶ Les lettres se suivent dans l'ordre alphabétique (codes 65 à 90 pour les majuscules, 97 à 122 pour les minuscules), ce qui simplifie les comparaisons et le passage d'une lettre à la suivante.

Le codage ASCII

Le **code ASCII** représente chaque caractère sur 7 bits, c'est un code de **taille fixe**. À chaque caractère est associée une configuration de 8 bits (1 octet), le bit de poids fort (le plus à gauche) est toujours égal à zéro.

- ▶ Les codes compris entre 0 et 31 sont des **caractères de contrôle**. Ils sont utilisés pour indiquer des actions telles que passer à la ligne (CR, LF), émettre un bip sonore (BEL), etc.
- ▶ Les lettres se suivent dans l'ordre alphabétique (codes 65 à 90 pour les majuscules, 97 à 122 pour les minuscules), ce qui simplifie les comparaisons et le passage d'une lettre à la suivante.
- ▶ On passe des majuscules au minuscules en modifiant le 5^{ième} bit, ce qui revient à ajouter 32 au code ASCII décimal.

Le codage ASCII

Le **code ASCII** représente chaque caractère sur 7 bits, c'est un code de **taille fixe**. À chaque caractère est associée une configuration de 8 bits (1 octet), le bit de poids fort (le plus à gauche) est toujours égal à zéro.

- ▶ Les codes compris entre 0 et 31 sont des **caractères de contrôle**. Ils sont utilisés pour indiquer des actions telles que passer à la ligne (CR, LF), émettre un bip sonore (BEL), etc.
- ▶ Les lettres se suivent dans l'ordre alphabétique (codes 65 à 90 pour les majuscules, 97 à 122 pour les minuscules), ce qui simplifie les comparaisons et le passage d'une lettre à la suivante.
- ▶ On passe des majuscules au minuscules en modifiant le 5^{ième} bit, ce qui revient à ajouter 32 au code ASCII décimal.
- ▶ Les chiffres sont rangés dans l'ordre croissant (codes 48 à 57), et les 4 bits de poids faible définissent la valeur en binaire du chiffre.

The table

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[59	3B	;	91	5B	[123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-`	63	3F	?	95	5F	`	127	7F	DEL

Un exemple

La phrase **ASCII 7 bits** s'écrit :

- ▶ En hexadécimal 41 53 43 49 49 20 37 20 62 69 74 73
- ▶ En binaire 01000001 01010011 01000011 01001001 01001001
00100000 00110111 00100000 01100010 01101001 01110100
01110011

La table

Mais où sont les caractères accentués ?

La table

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	Ł	224	E0	α
129	81	ü	161	A1	í	193	C1	⌞	225	E1	β
130	82	e	162	A2	ó	194	C2	⌟	226	E2	γ
131	83	â	163	A3	ú	195	C3	⌠	227	E3	π
132	84	ä	164	A4	ñ	196	C4	⌡	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	⌢	229	E5	σ
134	86	ç	166	A6	à	198	C6	⌣	230	E6	μ
135	87	ê	167	A7	ó	199	C7	⌤	231	E7	γ
136	88	ë	168	A8	¿	200	C8	⌥	232	E8	ϕ
137	89	è	169	A9	¸	201	C9	⌦	233	E9	θ
138	8A	ë	170	AA		202	CA	⌧	234	EA	Ω
139	8B	ì	171	AB	½	203	CB	⌨	235	EB	δ
140	8C	î	172	AC	¼	204	CC	〈	236	EC	∞
141	8D	ï	173	AD	¡	205	CD	=	237	ED	φ
142	8E	Ä	174	AE	«	206	CE	〉	238	EE	ε
143	8F	Å	175	AF	»	207	CF	⌫	239	EF	∩
144	90	É	176	B0		208	D0	⌬	240	F0	≡
145	91	æ	177	B1		209	D1	⌭	241	F1	±
146	92	œ	178	B2		210	D2	⌮	242	F2	Σ
147	93	ô	179	B3	—	211	D3	⌯	243	F3	∠
148	94	ö	180	B4	⌰	212	D4	⌱	244	F4	└
149	95	o	181	B5	⌱	213	D5	⌲	245	F5	└
150	96	û	182	B6	⌲	214	D6	⌳	246	F6	÷
151	97	ÿ	183	B7	⌳	215	D7	⌴	247	F7	≈
152	98	ÿ	184	B8	⌴	216	D8	⌵	248	F8	◊
153	99	ö	185	B9	⌵	217	D9	⌶	249	F9	•
154	9A	Ü	186	BA	⌶	218	DA	⌷	250	FA	·
155	9B	£	187	BB	⌷	219	DB	■	251	FB	√
156	9C	£	188	BC	⌸	220	DC	■	252	FC	n
157	9D	¥	189	BD	⌸	221	DD	■	253	FD	z
158	9E	℔	190	BE	⌹	222	DE	■	254	FE	■
159	9F	f	191	BF	⌹	223	DF	■	255	FF	

La table

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	Ł	224	E0	α
129	81	ü	161	A1	í	193	C1	⊥	225	E1	β
130	82	ë	162	A2	ó	194	C2	⌞	226	E2	γ
131	83	â	163	A3	ú	195	C3	⌏	227	E3	π
132	84	à	164	A4	ñ	196	C4	—	228	E4	Σ
133	85	ä	165	A5	ñ	197	C5	⌞	229	E5	σ
134	86	å	166	A6	æ	198	C6	⌏	230	E6	μ
135	87	ç	167	A7	ø	199	C7	⌞	231	E7	γ
136	88	è	168	A8	é	200	C8	⌏	232	E8	δ
137	89	ê	169	A9	ı	201	C9	⌏	233	E9	θ
138	8A	ë	170	AA	ı	202	CA	⌏	234	EA	Ω
139	8B	ì	171	AB	½	203	CB	⌏	235	EB	δ
140	8C	î	172	AC	¼	204	CC	⌏	236	EC	∞
141	8D	ï	173	AD	ı	205	CD	=	237	ED	φ
142	8E	â	174	AE	«	206	CE	⌏	238	EE	€
143	8F	Ä	175	AF	»	207	CF	⌏	239	EF	ü
144	90	Å	176	B0	☼	208	D0	⌏	240	F0	≡
145	91	æ	177	B1	▤	209	D1	⌏	241	F1	±
146	92	œ	178	B2	▤	210	D2	⌏	242	F2	≥
147	93	ø	179	B3	⌏	211	D3	⌏	243	F3	≤
148	94	ö	180	B4	⌏	212	D4	⌏	244	F4	∫
149	95	ó	181	B5	⌏	213	D5	⌏	245	F5	∫
150	96	û	182	B6	⌏	214	D6	⌏	246	F6	÷
151	97	ü	183	B7	⌏	215	D7	⌏	247	F7	≈
152	98	ý	184	B8	⌏	216	D8	⌏	248	F8	◊
153	99	ÿ	185	B9	⌏	217	D9	⌏	249	F9	●
154	9A	Û	186	BA	⌏	218	DA	⌏	250	FA	·
155	9B	Ɔ	187	BB	⌏	219	DB	▀	251	FB	√
156	9C	Ɔ	188	BC	⌏	220	DC	▀	252	FC	∞
157	9D	Ɔ	189	BD	⌏	221	DD	▀	253	FD	z
158	9E	Ɔ	190	BE	⌏	222	DE	▀	254	FE	ı
159	9F	Ɔ	191	BF	⌏	223	DF	▀	255	FF	

Problème d'encodage : @ ë □ ...

Le codage Unicode

Problèmes :

- ▶ Comment encoder les alphabets non-occidentaux avec de nombreux caractères ?
- ▶ Comment permettre de rajouter des caractères (extensibilité) ?
- ▶ Comment mettre tout le monde d'accord ?
- ▶ Comment être compatible avec les anciens formats, de type ASCII ?

Le codage Unicode

Problèmes :

- ▶ Comment encoder les alphabets non-occidentaux avec de nombreux caractères ?
- ▶ Comment permettre de rajouter des caractères (extensibilité) ?
- ▶ Comment mettre tout le monde d'accord ?
- ▶ Comment être compatible avec les anciens formats, de type ASCII ?

La solution : le standard Unicode.

- ▶ On utilise jusqu'à 4 octets : UTF8, UTF16, UTF32.
- ▶ On utilise un encodage de longueur **variable** (les 1 du début du premier octet donnent cette longueur).
- ▶ Une base de donnée officielle contenant le nom et le numéro de chaque symbole.
- ▶ Les caractères ASCII gardent le même code sur un octet en UTF8, pour la compatibilité.

Texte enrichi

Un texte n'est pas constitué uniquement de caractères !

Il y a aussi une **structure** :

- ▶ placement du texte
- ▶ liste
- ▶ lien
- ▶ image ...

Texte enrichi

Un texte n'est pas constitué uniquement de caractères !

Il y a aussi une **structure** :

- ▶ placement du texte
- ▶ liste
- ▶ lien
- ▶ image ...

On utilise des **langages descriptifs** de document comme le **html** sur internet ou le **L^AT_EX** pour les textes scientifiques. Ces documents sont organisés en arbre (comme XML).