

# 紙テープの呪縛

安岡孝一\*

紙テープはコンピュータの入出力媒体である以前に、電信機の送受信媒体として使用されてきた [24]。では、紙テープという媒体は、コンピュータにおけるテキスト情報処理にどのような影響を及ぼし、その呪いは現在の我々をどのように縛っているのだろうか。

## 電信機における紙テープ

19世紀から20世紀前半における電信機では、紙テープが送受信媒体として重要な位置を占めていた。電信線における信号の時系列変化を表現する媒体として、紙テープが非常に適していたのである。

Samuel Finley Breese Morse の電信機では、出力媒体に紙テープを用いており [1, 2]、そこにはモールス符号が視覚的に記されるようになっていた (図 1)。あるいは Charles Wheatstone は、モールス電信機の送信機に紙テープを使う方法を考案した [3]。真ん中に送り孔のあいた鑽孔テープと呼ばれるものであり、上段は電流 ON を、下段は電流 OFF を表していた (図 2)。Jean Maurice Emile Baudot の印刷電信機は、受信した文字を紙テープに順に印字していくものだった (図 3) し、Donald Murray の電信機は、送信機にかけられた鑽孔テープと同じ鑽孔テープが受信機に複製される仕掛けだった (図 4)。

このような中、Morkrum 社のタイプライタ型電信機 [7, 8] は、鑽孔テープに対し重要な改革をおこなった。鑽孔テープを 5 穴にしたのである (図 5)。これ以前の鑽孔テープは、あくまで電流の ON/OFF を表したものに過ぎなかったが、Morkrum 社の鑽孔テープは、1 文字を 5 穴で表し、そして 1 文字分に対応する鑽孔が 1 列ずつテープ上に並んでいた。5 つの穴に対応する電気信号は、現実には下から順に電信線を流れていくのだが、鑽孔テープを見る限り、5 穴すなわち 1 文字分があたかも一度に送信されているかのような印象を与えた。これに加え、「Carriage Return」と「Line Feed」という 2 種類の制御文字を導入した、という点も、Morkrum 社の鑽孔テープにおける重要な改革である (図 6)。鑽孔テープ上に表現されているのは、あくまで 1 次元的な文字の列であるにもかかわらず、受信機の印字ヘッドを左端に戻す「Carriage Return」と、印字紙を 1 行分送る「Line Feed」とで、2 次元の紙面上でのテキストが表現できるかのような幻影をもたらしたのである。

1919 年に Morkrum 社は、送受信機が一体となった電信機『Teletype』を発表する [9, 10]。タイプライタ型電信機の代名詞となった『Teletype』は、その後 AT&T のテレックスにおける標準機となり [11]、アメリカのみならず全世界を席捲していくのである。

\*京都大学人文科学研究所附属漢字情報研究センター

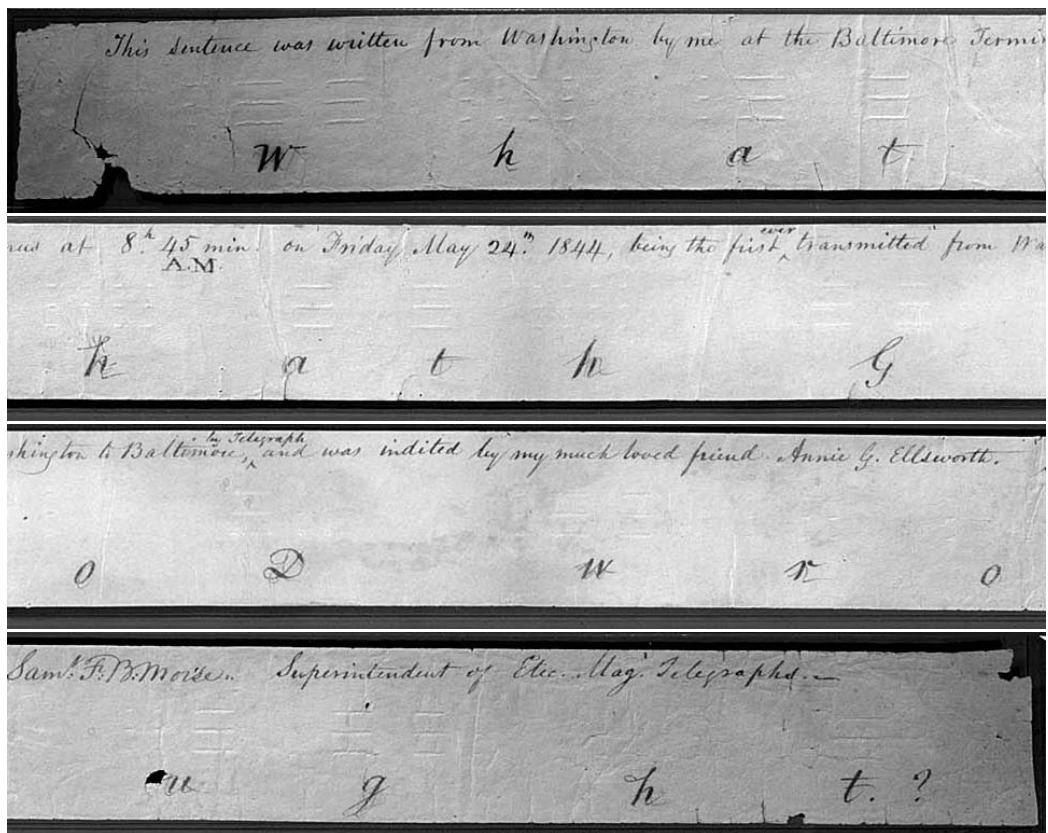


図 1: 1844 年 5 月 24 日付モールの受信テープ (Library of Congress 蔵)

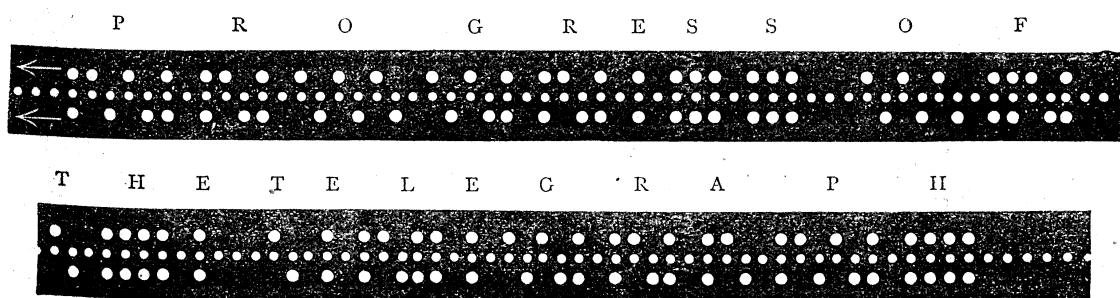


図 2: Wheatstone の鑽孔テープ [4]Vol.XI, No.287, p.511

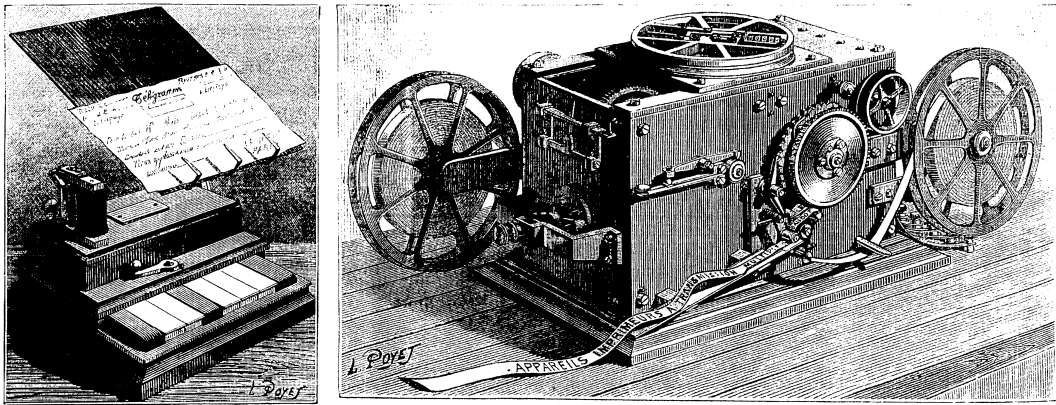


図 3: Baudot の送受信機 [5]No.25, p.2 & No.33, p.402

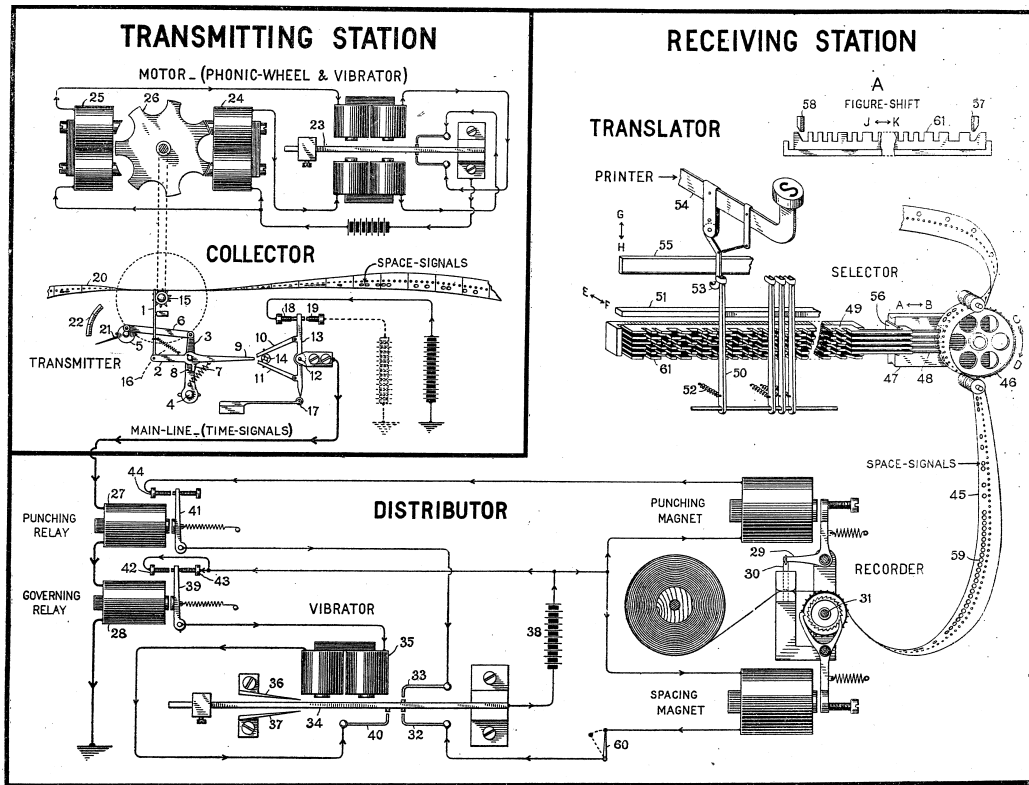


図 4: Murray 電信機のシステム構成 [6]Fig.20

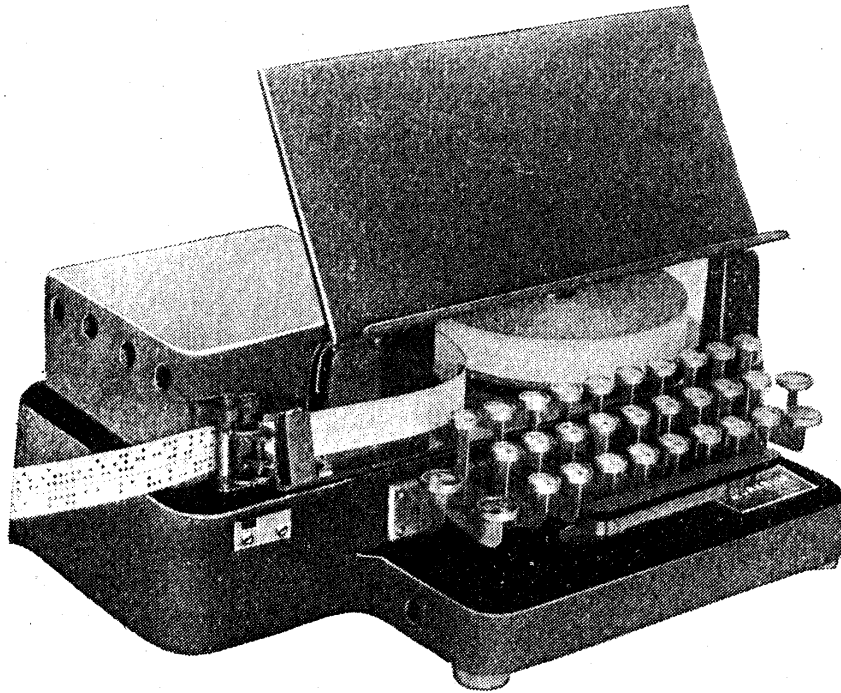


図 5: Morkrum 社の鑽孔タイプライタ [7]p.135

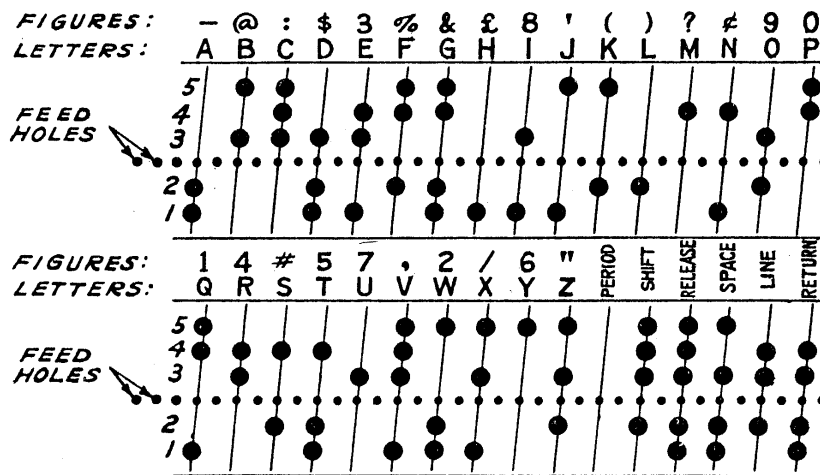


図 6: Morkrum 社の鑽孔テープ [8]No.782, p.575

## コンピュータにおける紙テープ

1936年、Alan Mathison Turingが考案した「機械」は、読み書き可能な紙テープの上を走り、紙テープ上の「シンボル」を書き換えていくことで「計算」をおこなうものであった。Turingは、あらゆる可能な計算をおこなう最小構成の仕組みとして、この「機械」を考案したのであり、換言すると、1次元のメモリとそれを書き換えていく機構さえあれば、計算機械すなわちコンピュータとしては十分である、ということが証明されたのである [12]。1次元のメモリ空間という考え方は、EDVACの設計者らに踏襲され [13]、1949年に発表されたEDSACで完成を見た [14]。また、EDSACの入出力媒体は5穴の鑽孔テープであり、鑽孔テープの作成あるいは鑽孔テープから「文字」への変換には、Creed社のタイプライタ型電信機『Teleprinter』を改造したものが用いられた (図7)。すなわち、プログラム格納型コンピュータはその登場時点で既に、メモリ空間と入出力媒体との両方において、紙テープの呪縛を受けてしまっていたのである。



図 7: EDSAC のプログラムを鑽孔するオペレータ [15]

## ホレリス定数と文字列

1956年、IBMはプログラミング言語FORTRANを発表した [17]。FORTRANでは、プリンタや磁気テープへの出力に際し、FORMAT文の中に短い文章を含め

ておくことができた。これに用いられたのが「Hollerith field」である。「Hollerith field」に記された文章の各文字は、1次元のメモリ空間に連続して配置された。これにより、プリンタや磁気テープへの出力は、連続したメモリを順に出力するだけで可能となったのである [16]。

1960年には、プログラミング言語 ALGOL 60 が発表された [20]。BNF<sup>†</sup>を用いて定式化されたことで有名だが、同時に「string」(文字列)という概念が定式化されている (図8)。もちろん ALGOL 60 の文法は、その実装を規定してはいないが、「string」に対しては、1次元のメモリ空間に連続して配置した文字の列、という実装が自然だった [21]。

つまるところ、1960年代におけるプログラミング言語の普及は、1次元のメモリ空間に連続して配置された文字列、という概念を、世界に広めていく結果となったのである。

## 2.6. STRINGS

### 2.6.1. Syntax

```
⟨proper string⟩ ::= ⟨any sequence of basic symbols not containing  
‘ or ’ ⟩⟨empty⟩  
⟨open string⟩ ::= ⟨proper string⟩|‘⟨open string⟩’|  
⟨open string⟩⟨open string⟩  
⟨string⟩ ::= ‘⟨open string⟩’
```

### 2.6.2. Examples

```
‘5k,’ - ‘[[[‘^=/:’Tt’’  
‘.. This *is *a *’string’
```

### 2.6.3. Semantics

In order to enable the language to handle arbitrary sequences of basic symbols the string quotes ‘ and ’ are introduced. The symbol \* denotes a space. It has no significance outside strings.

図 8: ALGOL 60 における string の定義 [20]p.302

## ASCII とテキスト

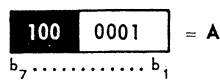
1963年に制定された ASCII (図9) は、アメリカにおける情報交換用の文字コードだったが、鑽孔テープとタイプライタ型電信機を、非常に意識したものとなっていた。NULLのビットパターンがオール0なのは、鑽孔テープの最初の何も打っていない部分を読み飛ばすためである。DELがオール1なのは、鑽孔テープでタイプミスをした場合、間違った部分の鑽孔を全部あけてしまうことでその部

<sup>†</sup>John Warner Backus が考案したプログラム言語記法 [19]。Norm Chomsky の生成文法における「Type 2 grammar」(文脈自由文法)[18]と本質的に同一だが、Backus は導出木の一意な生成という視点からこの記法に到達しており、Chomsky とはアイデアを異にする。

分を削除する、という慣行が、ずっとおこなわれてきたからである。CR (Carriage Return) と LF (Line Feed) は、まさにタイプライタ型電信機の制御文字であり、さらに BELL や HT (Horizontal Tabulation) まで追加されているのが、いかにもタイプライタっぽい。

これに加えて、1967年改正の ASCII(図 10) では、STX (Start of Text) と ETX (End of Text) という制御文字が導入されている。STX と ETX で前後を挟むことにより、一連のデータ中に複数の「テキスト」を含めることができる、というものである。したがって、ASCIIにおける「テキスト」は STX と ETX で挟まれた 1次元のデータを意味する、ということになったのである。

Example:



	000	001	010	011	100	101	110	111
0000	NULL	DC <sub>0</sub> ①	␣	0	@	P		
0001	SOM	DC <sub>1</sub>	!	1	A	Q		
0010	EOA	DC <sub>2</sub>	"	2	B	R		
0011	EOM	DC <sub>3</sub>	#	3	C	S		
0100	EOT	DC <sub>4</sub> (STOP)	\$	4	D	T		
0101	WRU	ERR	%	5	E	U		
0110	RU	SYNC	&	6	F	V		
0111	BELL	LEM	'	7	G	W		
1000	FE <sub>0</sub>	S <sub>0</sub>	(	8	H	X		
1001	HT SK	S <sub>1</sub>	)	9	I	Y		
1010	LF	S <sub>2</sub>	*	:	J	Z		
1011	V TAB	S <sub>3</sub>	+	;	K	[		
1100	FF	S <sub>4</sub>	,	<	L	\		ACK
1101	CR	S <sub>5</sub>	-	=	M	]		②
1110	SO	S <sub>6</sub>	.	>	N	↑		ESC
1111	SI	S <sub>7</sub>	/	?	O	←		DEL

UNASSIGNED

= 4 Bit Subset

図 9: 1963年版 ASCII[23]p.422

		COLUMN →										
		0	1	2	3	4	5	6	7			
ROW ↓	b <sub>7</sub> →	0	0	0	0	1	1	1	1			
	b <sub>6</sub> →	0	0	1	1	0	0	1	1			
	b <sub>5</sub> →	0	1	0	1	0	1	0	1			
	b <sub>4</sub> ↓											
	b <sub>3</sub> ↓											
	b <sub>2</sub> ↓											
	b <sub>1</sub> ↓											
0	0	0	0	0	NUL	DLE	SP	0	@	P	\	p
1	0	0	0	1	SOH	DC1	!	1	A	Q	a	q
2	0	0	1	0	STX	DC2	”	2	B	R	b	r
3	0	0	1	1	ETX	DC3	#	3	C	S	c	s
4	0	1	0	0	EOT	DC4	\$	4	D	T	d	t
5	0	1	0	1	ENQ	NAK	%	5	E	U	e	u
6	0	1	1	0	ACK	SYN	&	6	F	V	f	v
7	0	1	1	1	BEL	ETB	/	7	G	W	g	w
8	1	0	0	0	BS	CAN	(	8	H	X	h	x
9	1	0	0	1	HT	EM	)	9	I	Y	i	y
10	1	0	1	0	LF	SUB	*	:	J	Z	j	z
11	1	0	1	1	VT	ESC	+	;	K	[	k	{
12	1	1	0	0	FF	FS	,	<	L	\	l	!
13	1	1	0	1	CR	GS	-	=	M	]	m	}
14	1	1	1	0	SO	RS	.	>	N	^	n	~
15	1	1	1	1	SI	US	/	?	0	_	o	DEL

図 10: 1967 年版 ASCII[25]p.184

## UNIX とストリーム

1971 年、Ken Thompson と Dennis MacAlistair Ritchie が発表した UNIX [26] は、さまざまな意味での紙テープの呪縛を、一身に背負ったオペレーティングシステムであった。端末は基本的に『Teletype Model 37』[25]であり、入出力ともに ASCII の 1 次元データを想定していた<sup>‡</sup>。STX と ETX は使用しておらず、代わりに EOT (End of Transmission) を入力終了として用いていた。

プログラム中で用いられる文字列は、1 次元のメモリ空間に連続して並べられた ASCII データであり、末尾には NUL が入っていた。1975 年に発表されたプログラミング言語 C [28] においては、文字列は文字の配列であり、1 次元のメモリ空間に

<sup>‡</sup>1963 年版 ASCII に準拠した『Teletype Model 33』[22] も使用可能だったが、大文字は全て小文字とみなされた。



連続して並べられた ASCII データ、という考え方を完全に継承していた。

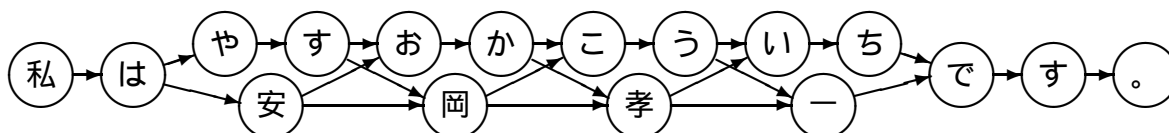
UNIX におけるファイルは、文字の 1 次元の並びを格納するものであり、すなわち、文字列を格納しているものである [27]。ファイルへの入出力は、文字列の各文字を、最初から順に 1 字ずつ入出力するだけのことである。この考え方は、1979 年発表の UNIX 第 7 版で拡張され、ファイルへの入出力も、端末への入出力も、プロセス間の通信も、全て「stream」と呼ばれる 1 次元のバイト列に抽象化された [32]。この結果、UNIX のユーザの間では、テキストを 1 次元のバイト列だとみなす傾向が、どんどん強まっていったのである。

## 紙テープの呪縛

1 次元のメモリ空間上に連続に並べられた「string」や、1 次元の入出力データである「stream」が一般化するにしたいが、「string」や「stream」上での高速な検索手法が提案された [29, 30, 31]。しかし、テキストは本当に 1 次元の「stream」なのだろうか。例として、以下の文章を考えてみよう。

やすおかこういち  
私は安岡孝一です。

この文章に対する検索要求を考えた場合、「安岡孝一です」がマッチングしてほしいのは言うまでもないことだが、「私はやすおか」という検索や「安岡こういち」という検索に対しても、マッチングが起こってほしい。だとすると、この文章における文字の流れは、おおむね以下のようなものになっていると言える。



このような構造を 1 次元の「stream」にムリヤリ埋め込むのは、不可能ではない。XHTML における ruby タグ [33] は、その手法の一つと言えるだろう。しかし、なぜわざわざ 1 次元に押し込まなければならないのだろうか。このような構造はコンピュータ上では、むしろ「linked list」を使って表現する方が、情報科学の思想に合致しているのではないだろうか。

現実のテキストには、ルビ以外にも、割注や脚注、図表やノンブルなど、さまざまな要素が含まれている。これらに対するコンピュータ上での表現は、それぞれ要素の目的に合致したものであるべきだろう。ところが、現代のテキスト情報処理の動向は、要素の目的や処理しやすさなど二の次で、闇雲に 1 次元に押し込めようとする強迫観念が働いているかのようである。紙テープの呪いが、我々を今も縛りつけているのだろう。

## 参考文献

- [1] “Electro-Magnetic Telegraph”, Journal of the Franklin Institute of the State of Pennsylvania and Mechanics’ Register, Vol.XX, No.5 (November 1837), pp.323-325.
- [2] Alfred Vail: The American Electro Magnetic Telegraph, Philadelphia: Leo & Blanchard (1845).
- [3] M. Charles Wheatstone: “Télégraphe Automatique Écrivain”, Comptes Rendus Hebdomadaires des Séances de l’Académie des Sciences, Tome XLVIII, No.4 (Janvier 1859), pp.214-220.
- [4] “The Progress of the Telegraph”, Nature, Vol.XI, No.281 (March 1875), pp.390-392; No.284 (April 1875), pp.450-452; No.285 (April 1875), pp.470-472; No.287 (April 1875), pp.510-512; Vol.XII, No.289 (May 1875), pp.30-32; No.291 (May 1875), pp.69-72; No.293 (June 1875), pp.110-113; No.295 (June 1875), pp.149-151; No.300 (July 1875), pp.254-256.
- [5] Ch. Bontemps: “Le Télégraphie Baudot”, L’Électricien, Tome II, No.19 (Janvier 1882), pp.321-325; No.21 (Février 1882), pp.417-422; Tome III, No.25 (Avril 1882), pp.1-5; No.26 (Mai 1882), pp.49-54; No.27 (Mai 1882), pp.97-102; No.30 (Juillet 1882), pp.241-246; No.31 (Juillet 1882), pp.289-295; No.33 (Août 1882), pp.401-406.
- [6] Donald Murray: “Setting Type by Telegraph”, Journal of the Institution of Electrical Engineers, Vol.34, No.172, pp.555-597 (February 1905).
- [7] “The Morkrum Telegraph Printers”, Telegraph and Telephone Age, No.764 (March 16, 1915), pp.134-136.
- [8] “The Morkrum Printing Telegraph System”, Telegraph and Telephone Age, No.782 (December 16, 1915), pp.575-576; No.783 (January 1, 1916), pp.17-18; No.784 (January 16, 1916), pp.35-36; No.785 (February 1, 1916), pp.65-66; No.786 (February 16, 1916), pp.89-90; No.787 (March 1, 1916), pp.111-112.
- [9] “Teletype”, Telegraph and Telephone Age, No.870 (August 16, 1919), p.i.
- [10] “The Morkrum Printing Telegraph System”, Telegraph and Telephone Age, No.875 (November 1, 1919), pp.528-533; No.876 (November 16, 1919), pp.557-564; No.877 (December 1, 1919), pp.586-589; No.878 (December 16, 1919), pp.620-622; No.879 (January 1, 1920), pp.9-12; No.880 (January 16, 1920), pp.32-35; No.881 (February 1, 1920), pp.69-72; No.882 (February 16, 1920), pp.100-105.

- [11] G. A. Locke: “Nation-Wide Teletypewriter Service”, Bell Laboratories Record, Vol.10, No.5 (January 1932), pp.145-149.
- [12] A. M. Turing: “On Computable Numbers, with an Application to the Entscheidungsproblem”, Proceedings of the London Mathematical Society, Ser.2, Vol.42, Nos.2143-2145, pp.230-265 (November 12, 1936); Vol.43, Nos.2197-2198, pp.544-546 (May 20, 1937).
- [13] John von Neumann: First Draft of a Report of the EDVAC, Moore School of Electrical Engineering, University of Pennsylvania (June 30, 1945).
- [14] M. V. Wilkes and W. Renwick: “The EDSAC”, Report of a Conference on High Speed Automatic Calculating Machines (June 1949), pp.9-11.
- [15] Maurice V. Wilkes, David J. Wheeler and Stanley Gill: The Preparation of Programs for an Electronic Digital Computer, Addison-Wesley Press, Cambridge (1951).
- [16] 704 Electronic Data-Processing Machine, International Business Machines, New York (1954).
- [17] J. W. Backus, R. J. Beeber, S. Best, R. Goldberg, H. L. Herrick, R. A. Hughes, L. B. Mitchell, R. A. Nelson, R. Nutt, D. Sayre, P. B. Sheridan, H. Stern, I. Ziller: The FORTRAN Automatic Coding System for the IBM 704 EDPM, Applied Science Division and Programming Research Dept., International Business Machines, New York (October 15, 1956).
- [18] Noam Chomsky: “On Certain Formal Properties of Grammars”, Information and Control, Vol.2, No.2 (June 1959), pp.137-167.
- [19] J. W. Backus: “The Syntax and Semantics of the Proposed International Algebraic Language of the Zurich ACM-GAMM Conference”, Proceedings of the International Conference on Information Processing (June 1959), pp.125-132.
- [20] Peter Naur, J. W. Backus, F. L. Bauer, J. Green, C. Katz, J. McCarthy, A. J. Perlis, H. Rutishauser, K. Samelson, B. Vauquois, J. H. Wegstein, A. van Wijngaarden, M. Woodger: “Report on the Algorithmic Language ALGOL 60”, Communications of the ACM, Vol.3, No.5 (May 1960), pp.299-314.
- [21] J. H. Wegstein and W. W. Youden: “A String Language for Symbol Manipulation Based on ALGOL 60”, Communications of the ACM, Vol.5, No.1 (January 1962), pp.54-61.

- [22] “This Is a Teletype Printer?”, *Administrative Management*, Vol.24, No.5 (May 1963), p.67.
- [23] S. Gorn, R. W. Bemer, J. Green: “American Standard Code for Information Interchange”, *Communications of the ACM*, Vol.6, No.8 (August 1963), pp.422-426.
- [24] 恐怖のルート 87, ウルトラマン, 第 20 話 (1966 年 11 月 27 日).
- [25] Fred W. Smith: “Revised U.S.A. Standard Code for Information Interchange”, *Western Union Technical Review*, Vol.21, No.4 (November 1967), pp.184-190.
- [26] K. Thompson, D. M. Ritchie: *UNIX Programmer’s Manual* (November 3, 1971).
- [27] Dennis M. Ritchie and Ken Thompson: “The UNIX Time-Sharing System”, *Communications of the ACM*, Vol.17, No.7 (July 1974), pp.365-375.
- [28] Dennis M. Ritchie: “C Reference Manual”, *Documents for Use with the UNIX Time-Sharing System* (May 1975).
- [29] Alfred V. Aho and Margaret J. Corasick: “Efficient String Matching: An Aid to Bibliographic Search”, *Communications of the ACM*, Vol.18, No.6 (June 1975), pp.333-340.
- [30] Donald E. Knuth, James H. Morris, Jr., and Vaughan R. Pratt: “Fast Pattern Matching in Strings”, *SIAM Journal on Computing*, Vol.6, No.2 (June 1977), pp.323-350.
- [31] Robert S. Boyer and J. Strother Moore: “A Fast String Searching Algorithm”, *Communications of the ACM*, Vol.20, No.10 (October 1977), pp.762-772.
- [32] *Unix Programmer’s Manual, Seventh Edition*, Bell Telephone Laboratories, Murray Hill (January 1979).
- [33] Marcin Sawicki, Michel Suignard, Masayasu Ishikawa, Martin Dürst, *Textin: Ruby Annotation, W3C Recommendation* (May 31, 2001).