Contents

Mule 概要 **1**
Mule とは 1
Mule でサポートする文字セット 2
文字セットのタイプ 4
Mule でサポートするコード系 5

EmacsLisp での多国語の扱い **6**
buffer, string での多国語テキスト . . . 6
文字オブジェクト 8
コード系オブジェクト 10
コード変換 11
文字カテゴリ 13

多国語関係以外の拡張 **14**
文字修飾 14
正規表現コンパイラ 15

その他の話題 **16**

♢ その他の話題

- Unicode
  今のところまだ Unicode をサポートする予定はない。

- DOS サポート
  阪大の東田氏による demacs の取り込み

- GNU Emacs Ver.19 の動向
  もうすぐβテストが開始されるが、今のところ Mule の Ver.19 対応の予定はない。 Ver.19 では LATIN-1 のみがサポートされる予定。

- 正式リリースの時期
  来春早々、タイ、ベトナム、ヘブライ、アラビア文字のサポート後。

◊◊ 正規表現コンパイラ

- 任意の正規表現をバックトラックなし探索用のコードにコンパイル

  複雑な正規表現の探索では１００倍以上の高速化

- 逆方向探索用のコードも生成

  オリジナルでは１文字ごとに前に戻りながら順方向マッチで調べていた。

- 文字列のチャンクを正規表現で指定できる

  例えば日本語用の文節の定義

```
"\\cK+\\cH*\\|\\cC+\\cH*\\|\\cH+\\|\\sw+"
```

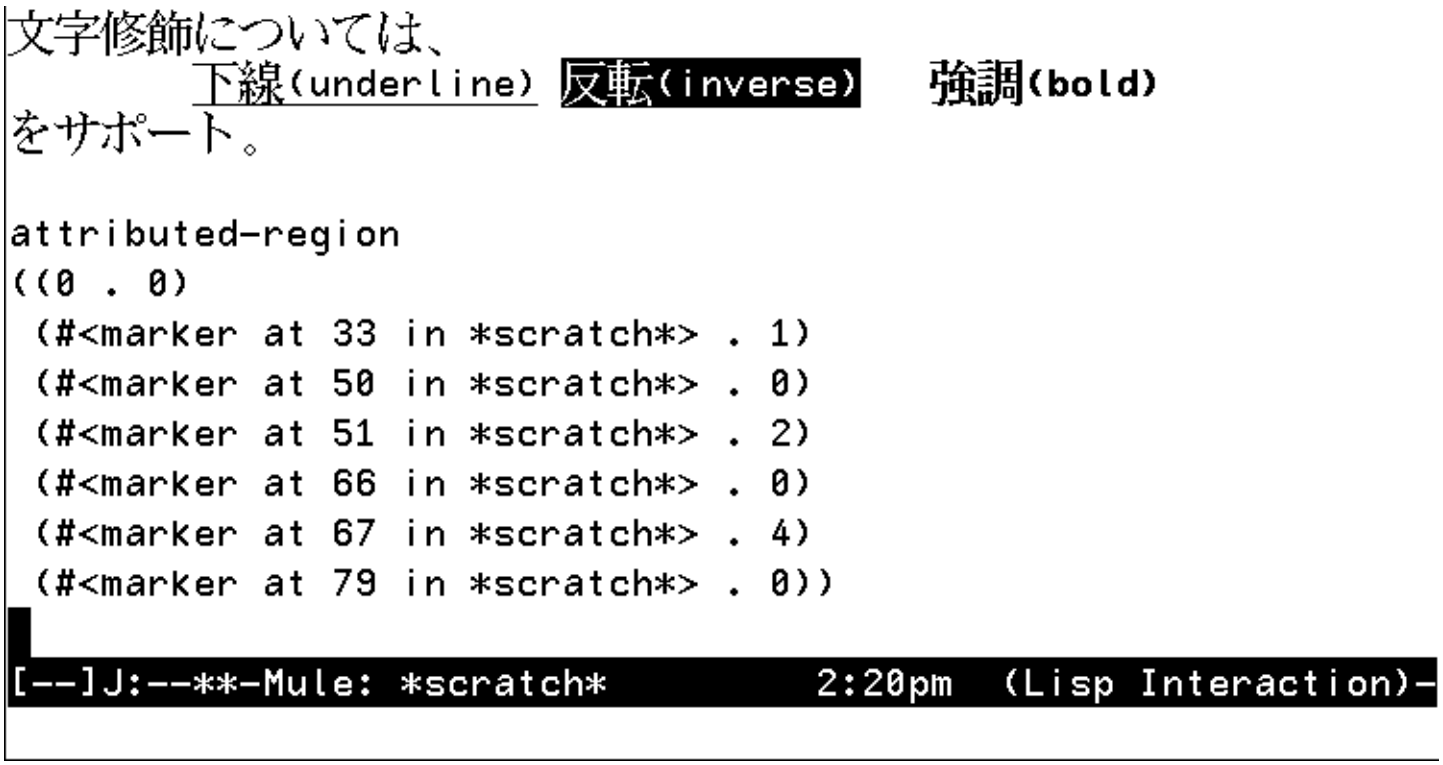
- コンパイルされた正規表現を文字列として保持

◇ 多国語関係以外の拡張

◇◇ 文字修飾

- 下線、反転、強調をサポート。
- 修飾情報はバッファごとにマーカーで保持

  attributed-region



♢♢ 文字カテゴリ

GNU Emacs のサポートする「文字シンタックス」に類似。

- 文字や文字セット単位で指定可能
- 複数のカテゴリを同時に持てる
- 正規表現では \\c? で指定。

```
(char-category ?漢)
"Cj"
(char-category ?あ)
"Hj"
(char-category ?汉)
"Cc"
(string-match "\\cH" "find ひらがな")
5
```

```
[--]J:--**-Mule: *scratch*         2:54pm  (Lisp Interaction)-
```

```
(code-convert-string "ABC漢字" *internal* *junet*)
"ABC^[$B4A;z^[(B"

(code-convert-string "ABC汉语" *internal* *junet*)
"ABC^[$A::So^[(B"

(code-convert-string "ABC漢字" *internal* *euc-japan*)
"ABC\xB4\xC1\xBB\xFA"

(code-convert-string
 (code-convert-string "ABC漢字" *internal* *euc-japan*)
 *euc-korea* *internal*)
"ABC늑샙"

(describe-coding-system *euc-japan*)
".
"
```

```
[--]-:--**-Mule: *scratch*          10:25pm  (Lisp Interac
Coding-system[E]: Coding-system of Japanese EUC (Extende\
d Unix Code).
Type: 2
Initial designations: [0 146 137 148]
  G0 -- ASCII
  G1 -- JIS X0208 Japanese
  G2 -- JIS X0201 Japanese Katakana
  G3 -- JIS X0212 Japanese Supplement
Flags: 96
  LongForm, ASCII@EOL, ASCII@CNTL, 8bit.

[--]-:-----Mule: *Help*             10:25pm  (Fundamental)
Mark set
```

♢♢ コード変換

コード変換が必要な局面

- ファイル I/O –
  `file-coding-system (buffer local)`
  `file-coding-system-for-read`
- プロセス I/O –
  `(process-coding-system PROC)`
- 表示 –
  端末表示 - `display-coding-system`
  X - ハードコーディングで変換
- キー入力 – `keyboard-coding-system`
- カット＆ペースト – `*ctext*`

◇◇ コード系オブジェクト

コード系オブジェクトは vector で表現。

TYPE:
　NIL- 無変換, T- 自動変換,
　0- 内部コード, 1-SJIS, 2-ISO2022, 3-Big5, ...

MNEMONIC: モードラインに表示する文字

LC-G0,1,2,3:
　各グラフィックレジスタの初期値

SHORT,ASCII-EOL,SEVEN,LOCK,...:
　ISO2022 用の指示

標準的なコード系

internal	内部コード
junet	いわゆる JIS、 JUNET で使用
euc-japan	いわゆる EUC
euc-korea	韓国語版 EUC
euc-china	中国語版 EUC
sjis	Shift-JIS
big5	Big5
autoconv	自動変換

```
(string-to-char "A")
65

?A
65

(string-to-char "漢")
1225921

?漢
1225921

(list (aref "漢" 0) (aref "漢" 1) (aref "漢" 2))
(146 180 193)

(char-to-string (make-character 146 180 193))
"漢"

(progn (setq x "漢字") (sref x 0))
1225921

(progn (sset x 3 ?文) x)
"漢文"

(progn (sset x 2 ?変) x)
"漢文"

(char-to-string (1+ ?あ))
"い"
```

◊◊ 文字オブジェクト

c : integer で代用

Type 1-1: $0 \leq c \leq$ 0x7F

$c =$ C

Type 1-2: 0x01A0 $\leq c \leq$ 0x0FFF

$c =$ ((LC & 0x7F)<<8) | C1

Type 1-3: 0x20A0 $\leq c \leq$ 0x3FFF

$c =$ ((LC' & 0x7F)<<8) | C1

Type 2-3: 0x10A0A0 $\leq c \leq$ 0x19FFFF

$c =$ ((LC & 0x7F)<<16) | (C1<<8) | C2

Type 2-4: 0x40A0A0 $\leq c \leq$ 0x5FFFFF

$c =$
((LC'' & 0x7F)<<16) | (C1<<8) | C2

Type 3-4: 0x60A0A0 $\leq c \leq$ 0x7FFFFF

$c =$ ((C1 - 0x40)<<16) | (C2<<8) | C3

```
(setq text "HELLO 你好 안녕하세요 Grüß Gott")
"HELLO 你好 안녕하세요 Grüß Gott"

(length text)
40

(string-width text)
31

(substring text (string-match "\\cc" text))
"你好 안녕하세요 Grüß Gott"

(string-match "\\cj" text)
nil

(save-excursion
  (set-buffer "temp")
  (insert "これは日本語の文字列です。\n")
  (insert "これは中国語の文字列です。\n")
  (forward-line -1)
  (replace-string "は中国" "も日本"))
"Done"
```

```
[--]-:--**-Mule: *scratch*         9:24pm  (Lisp Interact
```

これは日本語の文字列です。
これも日本語の文字列です。

```
[--]-:--**-Mule: temp              9:24pm  (Fundamental)-
```

♢ EmacsLisp での多国語の扱い

♢♢ buffer, string での多国語テキスト

- ASCII

  C – ASCII コードそのまま

- その他の文字 – 各文字を可変長シーケンスで

  LC C1 C2 ...

  LC は leading character (0x81 ≤ LC ≤ 0x9E)

  C1, C2... は文字コードの MSB を立てたもの

- テキスト表現の例：

A	b	ß	Ψ	あ	中	文
C	C	LC C1	LC C1	LC C1 C2	LC C1 C2	LC C1 C2

- leading char の区分

LC	type	sequence
なし	Type 1-1	C
0x81 ~ 0x8F	Type 1-2	LC C1
0x90 ~ 0x99	Type 2-3	LC C1 C2
0x9A ~ 0x9B	Type 1-3	LC LC' C1
0x9C ~ 0x9D	Type 2-4	LC LC'' C1 C2
0x9E	Type 3-4	LC C1 C2 C3

♢♢ Mule でサポートするコード系

- ISO2022 – どの文字セットにも対応
  - JUNET type
    7 ビット環境、 GL のみ、エスケープシーケンス使用
  - EUC type
    8 ビット環境、 GR に 2 バイトコード指定
  - CTEXT type
    8 ビット環境、 GR に 1 バイトコード指定
  - KOREAN-MAIL type
    7 ビット環境、 GR に 2 バイトコード指定、
    ロッキングシフト使用。
  - その他
- Shift-JIS – ASCII, JIS X0208, JIS X0201
- Big5 – Big5-ETen/HKU

♢♢ 文字セットのタイプ

Type 1-1: ASCII

Type 1-2: ASCII 以外の 1 バイト文字セット
JIS X0201, ISO 8859-*, ...

Type 1-3: プライベート 1 バイト文字セット
TSCII(タイ), KOI(キリル),
IPA(国際表音文字), ...

Type 2-3: 2 バイト文字セット
GB, JIS, KS

Type 2-4: プライベート 2 バイト文字セット
CNS

Type 3-4: プライベート 3 バイト文字セット

You may need many fonts to read all.
Please correct this incomplete list and add more!

Danish (Dansk)	Hej, Goddag
English	Hello
Esperanto	Saluton
Estonian	Tere, Tervist
FORTRAN	PROGRAM
Finnish (Suomi)	Hei
French (Français)	Bonjour, Salut
German (Deutsch Nord)	Guten Tag
German (Deutsch Süd)	Grüß Gott
Greek (Ηλλένικα)	Γειά σας
Hebrew	Shalom
Italiano	Ciao, Buonjorno
Maltese	Ciao
Nederlands, Vlaams	Hallo, Dag
Polish	Gen Dobré
Russian (Русский)	Здравствуйте!
Spanish	Hola
Swedish (Svenska)	Hej, Goddag
Norwegian (Norsk)	Hei, God dag
Turkish (Türkçe)	Merhaba
Yoguslavian	Zdravo

Japanese (日本語)	こんにちは
Chinese (中文,普通话,汉语)	你好
Cantonese (粵语,广东话)	早晨, 你好
Hangul (한글)	안녕하세요

[--]J:-----Mule: demo 9:31pm (Fundamental)-

♢♢ Mule でサポートする文字セット

- 英語 ... ASCII, JIS X0201 の左半分
- 日本語 ... JIS X0208, JIS X0201 の右半分
- 中国語 ... GB2312, CNS 11643
- 韓国語 ... KS C5601

- ヨーロッパ系 ... ISO 8859-1,2,3,4,5,7,9
- プライベート ... ISO2022 の規約に沿ったプライベートセット
- その他 ... Big5, FSCII, VISCII...

♢ Mule 概要

♢♢ Mule とは

Multilingual eanhacement
to
GNU Emacs

- Nemacs の後継ソフト

  Nemacs は Ver.3.3.2 が最終版

- GNU Emacs 18.58 ベース
- メイリングリスト上でβテスト中 (Ver.0.9.6)
- 現在、日本語、中国語、韓国語、多くのヨーロッパ系の言語をサポート
- 近い将来、さらに、東南アジア系言語、アラビア語、ヘブライ語などもサポート
- 文字セット、文字コード系は基本的に ISO2022

Mule - 多国語化 GNU Emacs

半田剣一
電子技術総合研究所

１９９２年１２月２日