

TR-I-0285

ATR_v-Talk 音声合成実験システム
ATR_v-Talk Speech Synthesis System

海木延佳 三村克彦 岩橋直人
Nobuyoshi KAIKI Katsuhiko MIMURA Naoto IWAHASHI
 匂坂芳典 道尻佳憲*
Yoshinori SAGISAKA yoshinori MITIJIRI

1992. 10

内容梗概

種々の複合音声単位を選択的に用いる音声規則合成実験システム(ATR_v-Talk システム)を構築した。本報告書はこのシステムを構成するプログラム及び、合成音作成に至るまでに生成される中間ファイルについての解説書である。

ATR 自動翻訳電話研究所
ATR Interpreting Telephony Research Laboratories
*東洋情報システム
*TOYO Information System

©ATR Interpreting Telephony Research Laboratories
©ATR自動翻訳電話研究所

目次

1	はじめに	1
2	プログラムシステム	2
2.1	入力データ解析プログラム	4
2.2	単位選択プログラム	5
2.3	韻律パラメータ生成プログラム	6
2.4	ケプストラムパラメータ修正及び合成プログラム	7
3	システムが用いているファイル構造とデータ構造	8
3.1	ファイル構造と構造体	8
3.2	音素片辞書構造	18
3.3	型コード	21
4	ライブラリ	28
5	動作環境	36
6	おわりに	36
7	文献	37

1 はじめに

種々の複合音声単位を選択的に用いる音声規則合成実験システムを構築した。本合成システムは、ルールによる韻律情報の生成、ケプストラム距離に基づいた4つの単位選択基準を用いた最適な複合音声単位の選択、及びケプストラムパラメータによる素片接続・編集・合成といった処理を統合化したものである。

本稿では、この合成システムを構成するプログラムについて解説する。

2 プログラムシステム構成

本システムは、(1)入力データ解析プログラム、(2)単位選択プログラム、(3)韻律パラメータの生成プログラム、(4)ケプストラムパラメータ修正及び合成プログラムより構成されている。全体のフローを図1に示す。

(1) 入力データ解析プログラム

parse_word	形態素情報の解析
Sentence_in	文データ(CMUフォーマット)の解析
parse_phone	音韻(含ポーズ)情報の生成

(2) 単位選択プログラム

unit_select_main	単位選択
serach_unit_connect	接続点の探索

(3) 韻律パラメータ生成プログラム

make_prosody	韻律パラメータ生成 (音韻継続長・パワー・ピッチ)
--------------	------------------------------

(4) ケプストラムパラメータ生成、素片接続編集及び合成プログラム

modify_prosody	ケプストラムパラメータの修正・素片接続
cepsynf	合成フィルタ

(注)イタリックはバイナリファイル(例 XXX.CP)

その他はキャラクタファイル(例 XXX.phra)

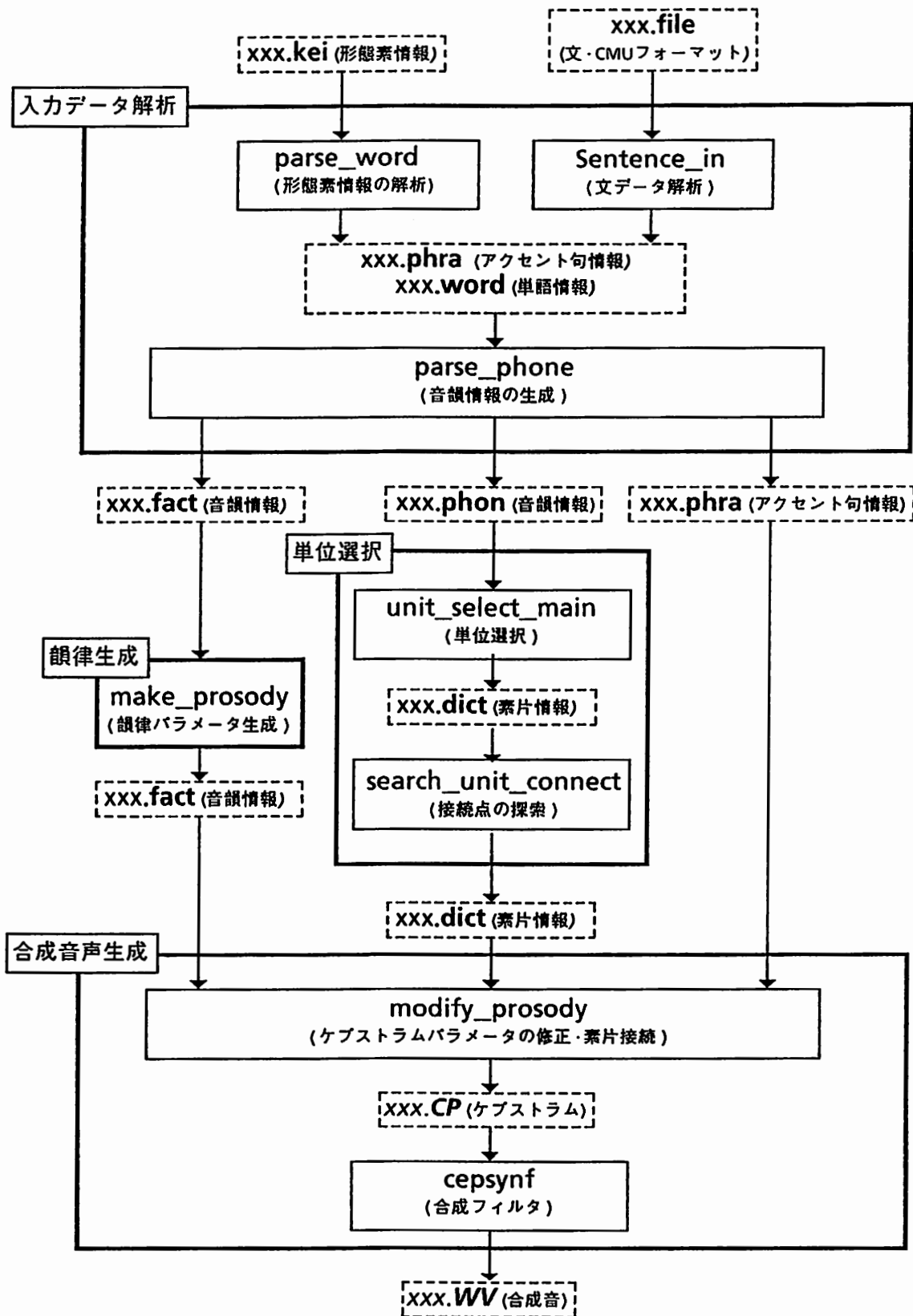


図1 全体のフロー

2.1 入力データ解析プログラム

形態素情報、およびローマ表記による文データ(CMUフォーマット)に基づき韻律パラメータの生成、および音素片の単位選択を行うための情報を生成する。

parse_word (形態素情報の解析)

入力ファイル	xxx.kei
出力ファイル	xxx.phra xxx.word

形態素情報に基づきアクセント句情報、単語情報を生成する。

Sentence_in (文データ [CMUフォーマット] の解析)

入力ファイル	xxx.file
出力ファイル	xxx.phra xxx.word

ローマ表記による文データ [CMUフォーマット] に基づきアクセント結合規則^[1]を用いアクセント核を決定し、アクセント句情報、単語情報を生成する。

parse_phone (音韻情報の生成)

入力ファイル	xxx.phra xxx.word
出力ファイル	xxx.fact xxx.phon xxx.phra

音韻情報、および音素片の単位選択を行うための情報を生成する。また、統計的手法によりポーズ^[2]を挿入する。

2.2 単位選択プログラム

韻律パラメータは入力音韻情報(音韻系列、アクセント型等)からルールに従って生成する。

unit_select_main (単位選択)

入力ファイル xxx.phon
出力ファイル xxx.dict

音韻情報に基づき音素片辞書より音素片を抽出する。抽出は、音韻環境を考慮してケプストラム距離による4つの距離尺度を用い、動的プログラミングによって行われる。^[3]

search_unit_connect (接続点の探索)

入力ファイル xxx.dict
出力ファイル xxx.dict

xxx.dictファイルの修正。素片境界近傍における音韻環境に応じて接続歪の小さい接続点を探索し、新しい素片境界点とする。^[4]

2.3 韻律パラメータ生成プログラム

韻律パラメータは入力音韻情報(音韻系列、アクセント型等)からルールに従って生成する。

make_prosody (韻律の決定)

入力ファイル	xxx.fact
出力ファイル	xxx.fact

統計的手法を用いて文音声の以下の韻律パラメータを決定する。

- (1) 音韻継続時間長^[5]
- (2) パワー^[6]
- (3) 基本周波数^[7]

2.4 単位接続編集及び合成プログラム

素片は既に分析されて改良ケプストラム¹⁸⁾の形で登録されている。

ルールによって音韻継続時間、ピッチ、とパワーを決め、使用する素片も決定した後に、ケプストラムパラメータで素片の接続編集を行い、対数振幅特性近似(LMA)フィルタで合成音を作成する。

modify_prosody (素片情報の展開と韻律情報の修正)

入力ファイル xxx.fact
 xxx.phra
 xxx.dict

出力ファイル xxx.CP

(1) 音素片辞書からケプストラムパラメータで表された素片を抽出し、規則により得られた音韻継続長にあわせて各素片を伸縮して接続し、合成音のケプストラムデータとする。

(2) パワー(ケプストラム0次項)の修正。音素片辞書の素片には素片固有のパワーが記述されているが、素片間にはばらつきがあるので、音韻毎のパワーの正規化によってこのばらつきを修正する。

(3) 点ピッチ情報から、フレーム毎のピッチデータを作成する。

cepsynf (合成フィルタ)

入力ファイル xxx.CP

出力ファイル xxx.WV

対数振幅特性近似(LMA)フィルタ。フィルタ係数としてケプストラムを用いる。¹⁹⁾

3 システムが用いているファイル構造及びデータ構造

3.1 ファイル構造と構造体

合成実験プログラムで用いられる、以下のデータファイルの形式について説明する。また、これらのデータファイルは、以下の構造体と同一になっている。

ファイル名	構造体名	
xxx.kei	9
xxx.file	10
xxx.phra	Factor	11
xxx.word	Phrase	12
xxx.fact	Select_Dict	13
xxx.phon	Phone	14
xxx.dict	Word_Data	15
xxx.CP	16
xxx.WV	17

形態素情報ファイル

ファイル修飾子 : kei

形式 : ASCII

フィールド区切り記号: スペース

内容

フィールド名	型	内容	例
text	*char	形態素のローマ字表記	moshi
acoustic	*char	発声表記(ローマ字)	moshi
cord	*char	品詞情報	感動詞
form	*char	活用型情報	-
type	*char	活用形情報	-
bun_num	int	文内通し番号	1
accent	*char	アクセント核の位置	1
kakari	*char	句末の係受けの深さ	5

文データ (CMU) フォーマットファイル

ファイル修飾子 : file

形式 : ASCII

フィールド区切り記号: スペース

内容

ローマ字表記

ただし、 @: ポーズ記号

例 MOSHI MOSHI @ KAIGI JIMUKYOKU DESU KA @

アクセント句情報

構造体名 : Phrase
 ファイル修飾子 : phra
 形式 : ASCII
 フィールド区切り記号: スペース

内容

フィールド名	型	内容	例
str	char[256]	アクセント句のローマ字表記	moshimoshi
acc	char	アクセント核の位置	0
pau	char	句末のポーズの有無 (0: 無 1: 有)	0
mora	char	アクセント句モーラ数	4
pitch	char	基本周波数 (現在不使用)	0
boundary	char	句境界の種類 (現在不使用)	0
comma	char	句末の読点の有無	1
para	char	句末の並列句の有無	0
kakari	char	句末の係受けの深さ	5
pau_dur	float	句末のポーズの継続時間長 (ms)	0.0
to_phoneme	int	phoneme 構造体へのポインタ	1
to_word	int	word 構造体へのポインタ	1

例 moshimoshi 0 0 4 0 0 1 0 5 0.0 1 1
 kaigijimukyokudesuka 0 0 10 0 0 1 0 5 538.3 9 3

単語情報

構造体名 : Word_Data

ファイル修飾子 : word

形式 : ASCII

フィールド区切り記号: スペース

内容

フィールド名	型	内容	例
text	char[64]	形態素のローマ字表記	kaigi
acoustic	char[64]	形態素の発声表記(ローマ字)	kaigi
hinshi	char	品詞情報	普名詞
inf_form	char	活用型情報	-
inf_type	char	活用形情報	-
word_num	char	アクセント句内通し番号	1
accent	char	アクセント核の位置	1
acc1	char	アクセント句のモーラ数	3
acc2	char	アクセント型	1
acc3	char	アクセント結合規則	5

例	moshi	moshi	感動詞	-	-	1	1	2	1	5
	moshi	moshi	感動詞	-	-	1	1	2	1	5
	kaigi	kaigi	普名詞	-	-	1	1	3	1	5
	jimukyoku	jimukyoku	普名詞	-	-	2	2	4	2	5
	desuka	desuka	その他	-	-	3	0	3	0	5

音韻情報(韻律情報生成用)

構造体名 : Factor
 ファイル修飾子 : fact
 形式 : ASCII
 フィールド区切り記号 : スペース

内容

フィールド名	型	内容	例
pre2_ph	phon_type	先先行音韻シンボル	K
pre_ph	phon_type	先行 音韻シンボル	A
id	phon_type	当該 音韻シンボル	I
next_ph	phon_type	後続 音韻シンボル	G
next2_ph	phon_type	後後続音韻シンボル	I
gem_p	char	先行音韻促音化フラグ (0: 無 1: 有)	0
gem_n	char	後続音韻促音化フラグ (0: 無 1: 有)	0
mo_sen	char	モーラ数 (文内)	14
mo_kok	char	モーラ数 (呼気段落内)	14
mo_acc	char	モーラ数 (アクセント句内)	10
pos_sen	char	位置 (文内)	1
pos_kok	char	位置 (呼気段落内)	1
pos_acc	char	位置 (アクセント句内)	1
hinshi	char	品詞	11
form	char	活用型	9
type	char	活用形	9
accentednesss	char	アクセント核の有無	0
pitch	int	基本周波数 (Hz)	173
est_dur	float	推定音韻継続時間長 (ms)	74.295
est_power	float	推定母音パワー (dB)	7.759
est_cep0	float	推定母音ケプストラム0次項	17.134
mean_cep0	float	平均母音ケプストラム0次項	18.290
gemf	char	促音化フラグ (0: 無 1: 有)	0
longf	char	長母音フラグ (0: 無 1: 有)	0
unvowelf	char	無声化フラグ (0: 無 1: 有)	0
sf	int	合成音声におけるフレーム番号	273

音韻情報(単位選択用)

構造体名 : Phone
 ファイル修飾子 : phon
 形式 : ASCII
 フィールド区切り記号: スペース

内容

フィールド名	型	内容	例
id	phon_type	音韻シンボル	SH
gemf	char	促音化フラグ (0:無 1:有)	0
longf	char	長母音フラグ (0:無 1:有)	0
unvowel f	char	無声化フラグ (0:無 1:有)	0
phone	char[8]	音韻のローマ字表記	sh
pred	float	音韻継続時間長 (現在不使用)	100.0
to_factors	int	factor 構造体へのポインタ	7

例

PAU	0 0 0	pau	100.000	0
M	0 0 0	m	100.000	1
O	0 0 0	o	100.000	2
SH	0 0 0	sh	100.000	3
I	0 0 0	i	100.000	4
M	0 0 0	m	100.000	5
O	0 0 0	o	100.000	6
SH	0 0 0	sh	100.000	7
I	0 0 0	i	100.000	8
K	0 0 0	k	100.000	9
A	0 0 0	a	100.000	10
I	0 0 0	i	100.000	11
G	0 0 0	g	100.000	12
I	0 0 0	i	100.000	13
J	0 0 0	j	100.000	14
I	0 0 0	i	100.000	15
M	0 0 0	m	100.000	16

単位素片情報

構造体名 : Select_Dict

ファイル修飾子 : dict

形式 : ASCII

フィールド区切り記号 : スペース

内容

フィールド名	型	内容	例
sp	int	音素片辞書の開始ポイント	30867
ep	int	音素片辞書の終了ポイント	30871
s_shift	int	開始ポイントでのシフト	1
e_shift	int	終了ポイントでのシフト	-1
ph_class	phone_class	調音タイプ(音素片接続に使用)	3

例

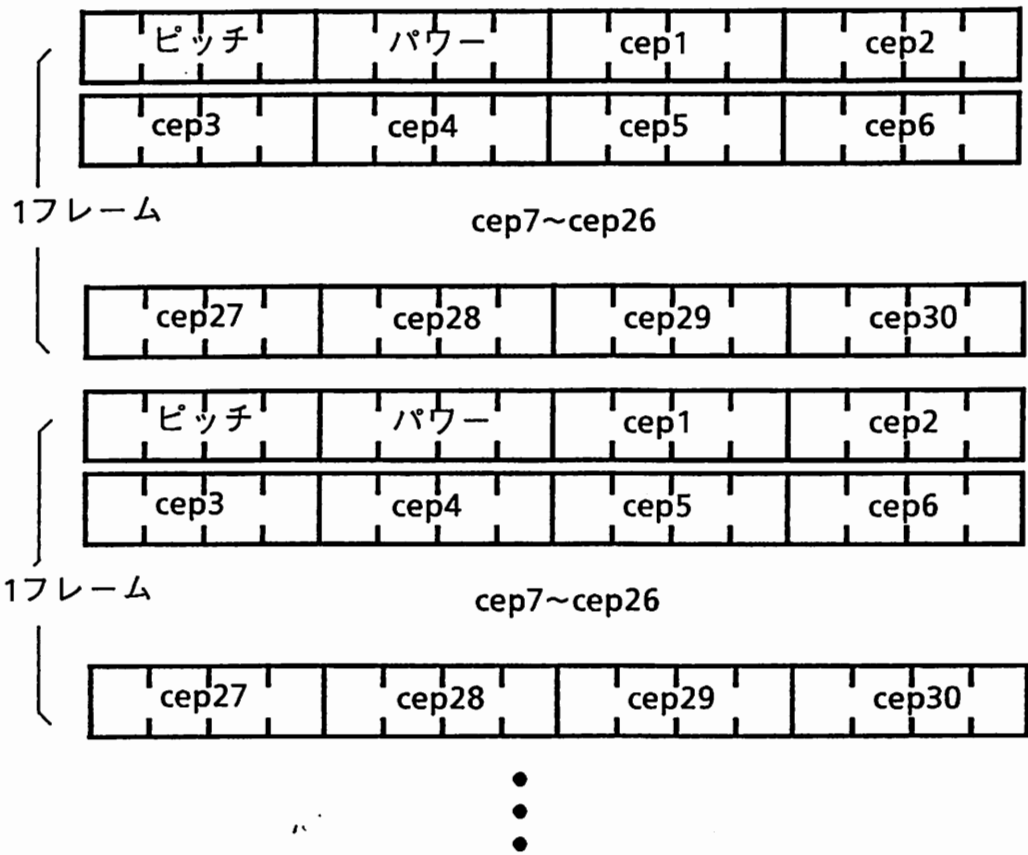
30867	30871	1	-1	3
30863	30865	-4	0	1
24503	24503	-1	1	7
4138	4140	0	0	4
7646	7646	-2	-4	1
23785	23787	-6	2	3
13600	13601	9	3	7
2998	2998	3	5	1
10074	10077	5	-10	1
13454	13455	-11	-12	1
33367	33369	-6	-2	10

ケプストラムデータファイル

ファイル修飾子 : CP

形式: binary (4byte float)

内容

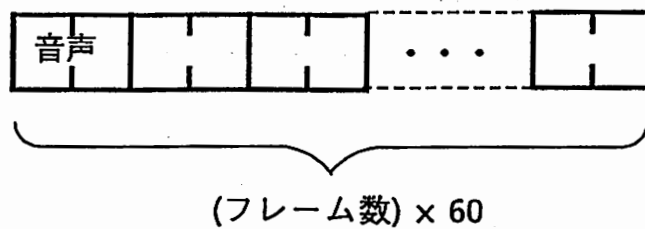


合成音ファイル

ファイル修飾子 : WV

形式: binary (2byte short)

内容



3.2 音素片辞書構造

合成実験用プログラムに共用されている、音素片辞書の構造体について説明する。これらの構造体は、ATR音声データベースのデータ構造に従い定義されている。

音韻記号層	Phone_Dict	19
イベント層	Event_Dict	20
異音化層	Allophone_Dict	20

構造体名 :Phone_Dict

内容 : 音韻記号層に相当するデータを格納する。

フィールド構成

フィールド名	型	内容	例
sp	int	開始時間 (0.1ms 単位)	1030
ep	int	終了時間 (0.1ms 単位)	1210
file_id	int	ファイル番号	1034
id	phon_type	音韻シンボル	H
gemf	char	促音化フラグ (0:無 1:有)	0
longf	char	長母音フラグ (0:無 1:有)	0
to_event	int	Event_Dict へのポインタ	2048

構造体名 : Event_Dict

内容 : イベント層に相当するデータを格納する。

フィールド構成

フィールド名	型	内容	例
sp	int	開始時間 (0.1ms 単位)	1030
ep	int	終了時間 (0.1ms 単位)	1210
file_id	int	ファイル番号	1034
id	event_type	イベントシンボル	eH
to_allophone	int	Allophone_Dict へのポインタ	503

構造体名 : Allophone_Dict

内容 : 異音層に相当するデータを格納する。

フィールド構成

フィールド名	型	内容	例
sp	int	開始時間 (0.1ms 単位)	1030
ep	int	終了時間 (0.1ms 単位)	1050
file_id	int	ファイル番号	1034
next	int	後続異音の有無 (1:有 -1:無)	-1
id	allophone_type	異音シンボル	FR

3.3 型コード

規則合成実験用プログラムに共用されている、以下の新たに宣言されている型、コードについて説明する。

Phone_type	22
Event_type	23
Alophone_type	24
Phone_class	25
品詞	26
活用型	27
活用形	27

型名 : Phone_type

内容 : 音韻の種類の変義。

コード

属性	コード	属性	コード
<i>a</i>	A	<i>r</i>	R
<i>i</i>	I	<i>w</i>	W
<i>u</i>	U	<i>y</i>	Y
<i>e</i>	E	<i>v</i>	V
<i>o</i>	O	<i>ts</i>	TS
N (撥音)	X	<i>sh</i>	SH
<i>p</i>	P	<i>py</i>	PY
<i>t</i>	T	<i>ch</i>	CH
<i>k</i>	K	<i>ky</i>	KY
<i>b</i>	B	<i>by</i>	BY
<i>d</i>	D	<i>gy</i>	GY
<i>g</i>	G	<i>dj</i>	DJ
<i>m</i>	M	<i>my</i>	MY
<i>n</i>	N	<i>ny</i>	NY
<i>s</i>	S	<i>ry</i>	RY
<i>h</i>	H	無音	PAU
<i>f</i>	F	その他	XX
<i>z</i>	Z		
<i>j</i>	J		

型名 : Event_type

内容 : イベントの種類定義。

コード

属性	コード	属性	コード
<i>a</i>	eA	<i>ts</i>	eTS
<i>i</i>	eI	<i>ch</i>	eCH
<i>u</i>	eU	<i>s</i>	eS
<i>e</i>	eE	<i>sh</i>	eSH
<i>o</i>	eO	<i>z</i>	eZ
>	eGT	<i>dj</i>	eDJ
<	eLT	<i>f</i>	eF
*>	e\$LT	<i>h</i>	eH
<i>tr</i>	e\$TR	<i>r</i>	eR
<i>p</i>	eP	<i>w</i>	eW
<i>t</i>	eT	<i>y</i>	eY
<i>k</i>	eK	<i>N</i>	eX
<i>b</i>	eB	<i>j</i>	eJ
<i>d</i>	eD	無音	ePAU
<i>g</i>	eG		
<i>cl</i>	eCL		
<i>\$cl</i>	e\$CL		
<i>mm</i>	eMM		

型名 : Allophone_type

内容 : 異音化の種類の変義。

コード

属性	コード
<i>fr</i>	FR
<i>dv</i>	DV
その他	UNKNOWN

型名 : Phone_class

内容 : 調音タイプ(音韻クラス)の定義。

コード

コード	音韻の種類
Unknown	その他
Vowel	A,I,U,E,O,X
SemiVowel	R,W,Y,RY
Nasal	M,N,MY,NY
VoicedStop	B,D,G,BY,GY,DJ
VoicedAffricate	J
VoicedFricative	Z,V
VoicelessStop	P,T,K,PY,KY
VoicelessAffricate	TS,CH
VoicelessFricative	S,H,F,SH,HY
Edge	PAU

品詞コード

コードテーブル :

属性	コード	属性	コード
記号	0	格助詞	15
形容詞	1	終助詞	16
普通名詞	4	接尾語	17
サ変名詞	5	接頭語	18
代名詞	6	補助動詞	19
数詞	7	固有名詞	30
副詞	8	形容名詞	31
連体詞	9	本動詞	32
接続詞	10	間投詞	33
感動詞	11	準体助詞	34
助動詞	12	並列助詞	35
副助詞	13	係助詞	36
接続助詞	14		

活用型コード

コードテーブル :

属性	コード
変則型活用	0
五段活用	1
上一段活用	2
下一段活用	3
サ変活用	4
カ変活用	5
特殊型活用	6
その他	9

活用形コード

コードテーブル :

属性	コード
未然形	0
連用形	1
終止形	2
連体形	3
仮定形	4
命令形	5
語幹	9

4 ライブラリ

共通なデータファイル及びデータ構造を扱うライブラリについて説明する。

Write_fact	29
Read_fact	29
Write_phon	29
Read_phon	29
Write_dict	30
Read_dict	30
Write_phrs	30
Read_phrs	30
Write_word	31
Read_word	31
make_head	32
readkeitai	32
GetKeitai	32
set_word	32
set_phrase	33
search_data	33
chk_head	33
noun_chk	34
acoustic_chk	34
sr_chk	34
check_phoneme_char	34
speech_chk	35
form_chk	35
type_chk	35

◎ 音韻情報(韻律生成用)を、ファイル (***.fact) に書き出す

```
int    write_fact ( header, fact, cnt )
char   header[]   出力ファイル名のヘッダ
Factor fact[]     音韻情報格納エリアのポインタ
int    cnt        データの総数
```

戻り値: 正常終了 0
 異常終了 -1

◎ 音韻情報(韻律生成用)を、ファイル (***.fact) に読み込む

```
int    read_fact ( header, fact )
char   header[]   入力ファイル名のヘッダ
Factor fact[]     音韻情報格納エリアのポインタ
```

戻り値: 正常終了 読み込んだデータ総数
 異常終了 -1

◎ 音韻情報(単位選択用)を、ファイル (***.phon) に書き出す

```
int    write_phon ( header, phoneme, cnt )
char   header[]   出力ファイル名のヘッダ
Phone  phoneme[]  音韻情報格納エリアのポインタ
int    cnt        データの総数
```

戻り値: 正常終了 0
 異常終了 -1

◎ 音韻情報(単位選択用)を、ファイル (***.phon) に読み込む

```
int    read_phon ( header, phoneme )
char   header[]   入力ファイル名のヘッダ
Phone  phoneme[]  音韻情報格納エリアのポインタ
```

戻り値: 正常終了 読み込んだデータ総数
 異常終了 - 1

◎ 単位選択情報を、ファイル (***.dict) に書き出す

```
int    write_phon ( header, dict, cnt )
char   header[]   出力ファイル名のヘッダ
Select_Dict dict[] 単位選択情報格納エリアのポインタ
int    cnt        データの総数
```

戻り値: 正常終了 0
 異常終了 - 1

◎ 単位選択情報を、ファイル (***.dict) に読み込む

```
int    read_phon ( header, dict )
char   header[]   入力ファイル名のヘッダ
Select_Dict dict[] 単位選択情報格納エリアのポインタ
```

戻り値: 正常終了 読み込んだデータ総数
 異常終了 -1

◎ アクセント句情報を、ファイル (***.phra) に書き出す

```
int    write_phra ( header, phra, cnt )
char   header[]   出力ファイル名のヘッダ
Phrase phra[]     アクセント句情報格納エリアのポインタ
int    cnt        データの総数
```

戻り値: 正常終了 0
 異常終了 - 1

◎ アクセント句情報を、ファイル (***.phra) に読み込む

```
int    read_phra ( header, phra )
char   header[]   入力ファイル名のヘッダ
Phrase phra[]     アクセント句情報格納エリアのポインタ
```

戻り値: 正常終了 読み込んだデータ総数
 異常終了 -1

◎ 形態素情報を、ファイル (***.word) に書き出す

```
int    write_word ( header, word, cnt )
char   header[]   出力ファイル名のヘッダ
Word_Data word[]  形態素情報格納エリアのポインタ
int    cnt        データの総数
```

戻り値: 正常終了 0
 異常終了 -1

◎ 形態素情報を、ファイル (***.word) に読み込む

```
int    read_word ( header, word )
char   header[]  入力ファイル名のヘッダ
Word_Data word[] 形態素情報格納エリアのポインタ
```

戻り値: 正常終了 読み込んだデータ総数
 異常終了 -1

◎ 出力ファイル用のヘッダーを作成する

```
void make_head( header, chk_name )
    char header[] 出力ファイル名ヘッダー格納エリア
    char chk_name[] 入力ファイル名
```

◎ 形態素単位データファイルより単語情報構造体にデータをセットする

```
int readkeitai( data, keitai )
    Word_Data data[] 単語情報格納エリアへのポインタ
    char keitai[] 入力ファイル名
```

戻り値: 正常終了 入力データの数
異常終了 -1

◎ 形態素単位データファイルより文節情報構造体にデータをセットする

```
int GetKeitai( phr, keitai )
    Phrase phr[] 文節情報格納エリアへのポインタ
    char keitai[] 入力ファイル名
```

戻り値: 正常終了 入力データの数
異常終了 -1

◎ 入力された文章から単語情報構造体を作成する

```
int set_word( str, word )
    char str[] 入力データ文字列
    Word_Data word[] 単語情報格納エリアへのポインタ
```

戻り値:: 入力データの数

◎ 単語情報を使用して文節情報構造体にデータをセットする

```
int  set_phrase( cnt,word,data_inf )
        int      cnt      wordのデータ数
        Word_Data word[]   単語情報格納エリアへのポインタ
        Phrase   data_inf[] 文節情報格納エリアへのポインタ
```

戻り値:: 入力データの数

◎ 単語情報に、アクセント核の位置、品詞情報、発声表記などのデータをセットする

```
int  search_data( word,dtp,cnt,no )
        Word_Data word[]   単語情報格納エリアへのポインタ
        char      *dtp     比較単語へのポインタ
        int       cnt      現在のポインタの位置
        int       *no      通し番号
```

戻り値:: 正常終了 チェックした単語の数
 異常終了 -1

◎ その単語が、文節の最初かどうかのチェックを行なう

```
int  chk_head( word,chk )
        Word_Data word[]   単語情報格納エリアへのポインタ
        int       chk     チェックを行なうデータのポインタ
```

戻り値: 文節の頭 1
 文節の頭以外 0

◎ その単語と次の単語がつながると複合名詞となるか、チェックを行なう

```
int noun_chk( word, cnt )
    Word_Data word[]  単語情報格納エリアへのポインタ
    int cnt          チェックを行なうデータのポインタ
```

戻り値: 複合名詞 1
 複合名詞以外 0

◎ 発声表記より、文字表記のセットを行なう

```
void acoustic_chk( word, cnt )
    Word_Data word[]  単語情報格納エリアへのポインタ
    int cnt          データの総数
```

◎ "/"の数を数える

```
int sr_chk( str )
    char str[]      チェックを行なう文字列
```

戻り値: "/"の数

◎ 入力されたキャラクターに対応するidナンバーを返す

```
int check_phoneme_char( phon_id )
    char phon_id[]   チェックを行なう音素キャラクター
```

戻り値: idナンバー

◎ 入力された品詞名に対応するコードナンバーを返す

```
int  speech_chk( str )  
      char      str[]   チェックを行なう品詞名
```

戻り値: コードナンバー

◎ 入力された活用型に対応するコードナンバーを返す

```
int  form_chk( str )  
      char      str[]   チェックを行なう活用型
```

戻り値: コードナンバー

◎ 入力された活用形に対応するコードナンバーを返す

```
int  type_chk( str )  
      char      str[]   チェックを行なう活用形
```

戻り値: コードナンバー

5 動作環境

現在本システムは、同一ソースコードで以下の環境で動作している。

1. DECStation 5000, 3100 シリーズ
cc 1.31, 2.0, 2.10
gcc 1.39, 2.0
2. HP 9000 シリーズ
cc 8.05, 8.07
gcc 2.1
2. SUNSPARC シリーズ
cc 4.1
gcc 1.40

6 おわりに

種々の複合音声単位を選択的に用いる規則合成実験システムを構成するプログラムについて解説を行った。プログラムの構成は大きく4つに分け、(1)入力データの解析部、(2)単位選択部、(3)韻律情報生成部、(4)素片接続・編集・合成部として、いる。更に、システムが用いているファイルとデータの構造・型コード、及びプログラム中で使われているライブラリ、動作するコンピュータ環境について説明を加えた。

今後は、この実験システムをもとに一層の合成音の品質の向上が望まれる。

7 参考文献

- [1] 匂坂、佐藤「テキストからの音声合成を目的とした日本語アクセント結合規則」 通研研究実用化報告32、11、1983、p.2253-2265
- [2] 海木、匂坂「局所的句構造に基づくポーズ長の分析」 信学技報SP91-130、1992.3、pp.63-69
- [3] 岩橋、海木、匂坂「音響的尺度に基づく複合音声単位選択法」 信学技報SP91-5、1991.5、pp.33-40
- [4] 安部、武田、匂坂「音韻環境に応じた音声合成素片の接続方法の検討」 信学技報SP89-66、1989.11
- [5] 海木、武田、匂坂「言語情報を利用した母音継続時間長の制御」 信学論(A)、J75-A、3、p.467,473
- [6] 三村、海木、匂坂「統計的手法を用いた音声パワーの分析と動特性の制御」 信学技報SP91-4、1991.5、pp.25-32
- [7] 海木、匂坂「局所的句構造に基づく F_0 制御」 信学技報SP92-6、1992.5、pp.41-46
- [8] 今井、阿部「改良ケプストラム法によるスペクトル包絡の抽出」 信学論(A)、J62-A、4、p.217,1979
- [9] 今井「対数振幅近似(LMA)フィルタ」 信学論(A)、J63-A、12、p.886,1980