

日本語諸方言の音調体系の定式化（6）

岡田 英俊

0 序

前稿（岡田（1993b））に続いて、日本語諸方言の音調体系の定式化を行う。今回は、枠組みの拡張・修正を、中心とする。§1～§4において、これを扱う。

具体的な音調体系としては、まず、§5において、甌島方言主流アクセントを取り上げる。この音調体系については、2つ、分析を示す。そのうちの1つは、今回、新たに定義した関数によって、初めて、容易に定式化できるようになった。技術的な事項ではあるが、この音調体系を取り上げる動機が、ここに存する。

甌島方言主流アクセントとの対照のため、§6と§7において、それぞれ、上甌島江石方言と鹿児島方言の音調を、簡単に扱う。

最後に、プログラミング言語 Scheme により、この3つの音調体系の定式化を示す。

1 音調変動セグメントに関する修正

まず、音調変動セグメントに関して、修正を加える。

音調変動セグメントは、岡田（1988：282）において、導入した。直観的に言えば、「高→低」「中→高」などの音調の変動自体を表現するためのものである。

現在、音調変動セグメントとして使用しているのは、次の5つである。e 以外のものについては、対応する音調の変動の例を、付記する。

- e : 音調パターンの始点・終点を示す
- : 高→低
- + : 低→高
- ' : 高→中, 中→低
- +' : 低→中, 中→高

岡田（1992：116）で述べたとおり、枠組みをプログラミング言語で表現する場合には、「+」「—」などの記号を用いない。次のように、置き換える（eについては、そのままである）。

- : neg
- + : pos
- ' : neg2
- +' : pos2

本稿では、演算記号との混同を避けるため、プログラミング言語によらない場合も、これら

の記号を用いる。

以上が、現時点における、音調変動セグメントに関する事項である。これに修正を加え、音調変動セグメントを、すべて、整数で表現することとする。

整数を、次のように割り当てる。

e	:	0
neg	:	-4
pos	:	4
neg2	:	-2
pos2	:	2

それぞれの数値の由来を述べる。

eは、音調パターンの始点と終点を示す機能しか有していない。実質的な音調の変動を表すことはない。よって、0を割り当てる。その他の4つについては、まず、negに、-4を割り当てる。posは、negと絶対値が等しく、符号が正であると考えて、4を割り当てる。neg2, pos2は、それぞれ、neg, posと符号が等しく、絶対値が2分の1であると考えて、-2, 2を割り当てる。

negに割り当てた-4という数値に、理論上、特別の意味はない。しかし、この数値の着想には、一応の背景がある。音調変動セグメントnegは、代表的には、「高」から「低」への変動である。1から5までの5段階で音調を表現すると、5から1への変動ということになる。すなわち、これが、-4である。

以上が、数値の由来である。

新たな音調変動セグメントが必要になったときは、-4以上4以下の範囲内で、設定する。この方式によって、不足が生ずる場合は、根本的な修正が必要である。取り得る値の範囲を広げる（例えば、-8以上8以下にする）、あるいは、音調変動セグメントの表現を有理数に変更する、などの措置が考えられる。しかし、当分の間、根本的な修正の必要はないと思われる。

注意すべき事項を、述べておく。

日本語諸方言のうち、音調の変動の向きのみが問題になる方言の場合、従来、neg, posで表していた音調変動セグメントは、今回の変更により、それぞれ、-4, 4で表される。そのため、表面上は、当該方言において、新たに、音調の変動の量が考慮されることになったかのように見える。

これは、表面上のことに過ぎない。基本的な捉え方には、何の変更もない。そのことを、あえて明確にしたいならば、当該方言について、例えば、「使用する音調変動セグメントは、-4, 4のいずれかである」という宣言を行えばよい。

また、別の事項であるが、すでに述べたとおり、音調変動セグメントを表現する数値の背景には、調値の5段階表示がある。しかし、これは、数値の着想を得た契機に過ぎない。日本語諸方言について、いわゆる段階観に類する立場を取ることにしたわけではない。音調の変動(の向き、及び、方言によっては、その量)を理論上有意義であるとする立場に、変わりはない(5段階の調値で表現されたデータの扱いが容易になったことは、事実である。これについては、後述する)。

以下、例を挙げる。「高高高高低」という音調パターンを、従来の方式で表現すると、次のようになる。

{(e1, 0, e), (i, 4, neg), (e2, 5, e)}

第4モーラの終点に、音調変動セグメント *neg* が対応する。すなわち、第4モーラの終点で、高から低への音調の変動が生ずる。今回の修正に基づいて、音調変動セグメントを整数で表現すると、次のようになる。

$$\{(e1, 0, 0), (i, 4, -4), (e2, 5, 0)\}$$

この例において、*i* というインデックスは、適当に与えた。一般に、表層の音調のみを示す場合は、インデックスを、適当に与えることになる（音調パターンの始点と終点を表す音調変動のインデックスとしては、習慣上、常に、それぞれ、*e1* と *e2* を用いることとしている）。事実上不要なインデックスを、その都度設定して、表示するのは、煩雑である。そこで、簡素化のため、今後は、各音調変動のインデックスを省いて表示することを、認める。

また、音調パターンの始点を表す音調変動は、いかなる場合でも、(*e1*, 0, 0) (インデックスを省いた表示では、(0, 0)) である。よって、簡素化のため、この音調変動を全く表示しないことを、認める（終点を表す音調変動には、音調パターンの長さを示す機能があるので、省くことができない）。

この2つの簡略化により、今の例は、次のように表示される。

$$\{(4, -4), (5, 0)\}$$

これらの簡略化は、表層の音調を示す際に、事実上不要なものを表示する手間を省くための、便宜上の措置に過ぎない。枠組み自体の変更ではない。

その他の例を挙げる。対照を容易にするため、すべて、5モーラのパターンとする。「降」は、1モーラ内における高→低である。

低高高高低：{(0, -4), (1, 4), (4, -4), (5, 0)}

低低降降低：{(0, -4), (2, 4), (5/2, -4), (5, 0)}

高高中中低：{(2, -2), (4, -2), (5, 0)}

中高高中中：{(0, -2), (1, 2), (2, -2), (5, 0)}

言語によっては、習慣上、音調のデータが、主として、5段階の調値によって、示される。今回の、音調変動セグメントの表現の変更により、定式化に際して、5段階の調値を、直接扱うことができるようになる。

5段階の調値で示されたデータの表現の例を、1つ、挙げておく。次のような、5音節の音調パターンを考える。

22 55 33 33 21

これを、本稿の枠組みで表現すると、次のようになる。

$$\{(0, -3), (1, 3), (2, -2), (4, -1), (9/2, -1), (5, 0)\}$$

直観的に言えば、まず、第1音節の始点（すなわち、音調パターンの始点）に、-3という音調変動セグメントが対応する。また、第1音節の終点、第2音節の終点、第4音節の終点に、それぞれ、3, -2, -1の音調変動セグメントが対応する。そして、第5音節の中間点に、-1の音調変動セグメントが対応する。

日本語諸方言の場合、第1モーラが「低」ならば、音調パターンの始点に、音調変動セグメント-4 (=1-5) を対応させる。それと同様に、この例では、音調パターンの始点に、音調変動セグメント-3 (=2-5) を対応させる。これは、枠組みの方針上、当然の措置である。

なお、音調変動セグメントを整数で表現するようになったことに伴い、関連する関数の修正が、当然、必要となる。岡田(1992)で定義を示した諸関数の一部分は、そのままでは、通用し

ない。しかし、これは、プログラムのみに関する事項に過ぎない。よって、省略する。

2 調値算出関数

次に、5段階の調値を算出する関数を、導入する。関数の名称を、tone-value とする。この関数は、音調パターンを入力として受け取り、調値の配列（後述）を出力する。

以下、この調値算出関数について、解説を加える。関数の定義自体については、省略する。

なお、音調の変動のみに着目することとしている音調体系（例えば、日本語諸方言の音調体系）については、5段階の調値による表現を、想定していない。そのため、この調値算出関数を使用することはない。

また、今のところ、定式化の対象として、日本語以外の言語は、取り上げていない。よって、現時点では、この関数を使用する機会が、全くない。今後、日本語以外の言語を取り上げることがあれば、この関数を使用する機会も、生ずる。

関数 tone-value は、音調変動セグメントが数値で表されていないとしても、別の形で、定義できる。この関数を導入することと、音調変動セグメントを数値化したこととの間に、直接の関係はない。

この関数の出力たる「調値の配列」について、説明を加える。§1の最後の例を、再び用いる。次のとおりである。

22 55 33 33 21

調値の配列では、これが、次のように表現される。

#((2) (5) (3) (3) (2 1))

記号#は、配列を示す(プログラミング言語 Scheme には、1次元の配列のみが、備わっている。そして、通常、それを、vector と呼ぶ。しかし、本稿では、便宜上、「配列」と呼ぶ)。この配列においては、第0要素が、リスト(2)である(最初の要素が、「第0」の要素とされる)。第1要素、第2要素、第3要素が、それぞれ、リスト(5)、(3)、(3)である。最後の要素たる第4要素が、リスト(2 1)である。

第0要素たるリスト(2)は、第1音節の調値が(音節を通じて)2であることを示す。リスト(2 2)では、表現しない。第1要素から第3要素についても、同様である。

第4要素たるリスト(2 1)は、音節内において、調値が2から1へ変動することを示す。したがって、例えば、(1 3 2)のようなリストも、調値の配列の要素として、許される。(1 3 2)は、音節内において、調値が1, 3, 2の順に変動することを示す。

なお、音調パターンは、リストで表現されている。調値の列も、リストで表現することは、可能である。しかし、調値の列については、例えば、「第2要素(すなわち、第3音節の調値)を取り出す」というような形での使用が多い。また、要素の数も、確定している(n 音節ならば、常に、第0要素から第 $(n-1)$ 要素までの n 個の要素によって、表現される。一方、1つの音調パターンに属する音調変動の数には、理論上、制限がない)。よって、配列を用いることとした。

関数 tone-value は、一般には、音調パターンの算出の途上においても、呼び出すことができるものとする。ただし、日本語諸方言については、このような使用を、認めない。直観的に言えば、日本語諸方言については、規則の適用条件においても、「高」「低」などに直接言及することを、認めない。

3 音調変動挿入関数

次に、新しい関数として、音調パターンに音調変動を挿入する関数を、導入する。

音調パターン「高低高低」を「中低高低」に変えるような操作は、直観的には、ごく普通の操作であると感じられる。実際、§5.2では、甌島方言主流アクセントの2つの分析のうち的一方において、このとおりの操作が、必要となる。

この操作は、下記の音調パターン(a)に何らかの規則を適用して、音調パターン(b)を導くということである。

(a) $\{(1, -4), (2, 4), (3, -4), (4, 0)\}$

(b) $\{(0, -2), (1, -2), (2, 4), (3, -4), (4, 0)\}$

(a)のように、「高」で始まるパターンにおいては、パターンの始点に、音調変動が存在しない。一方、(b)のように、「中」で始まるパターンにおいては、パターンの始点に、-2を音調変動セグメントとする音調変動が存在する。

したがって、(a)から(b)を導くには、音調変動を、新たに、1つ、挿入する必要がある（それとともに、点(1)に対応している音調変動セグメントを、-4から-2に変更する必要がある。この操作については、問題がない）。これは、一例に過ぎない。これ以外にも、音調変動を挿入する操作が必要となる場合があることは、十分、予想される。

ところが、現行の枠組みには、音調変動を挿入する関数が、存在しない。そこで、定式化を容易にするため、この操作を実行する関数として、新たに、insert-tc を設ける。この関数は、音調変動 tc を、音調パターン tc-list の中の適切な位置（自動的に判定される）へ挿入し、得られた音調パターンを返す。定義は、次の通りである（岡田(1992)において定義した関数を、断りなく使用する）。

```
(define (insert-tc tc tc-list)
  (if (null? tc-list)
      tc-list
      (if (and (rat>=? (time-comp tc) (time-comp (car tc-list)))
              (rat<? (time-comp tc) (time-comp (cadr tc-list))))
          (cons (car tc-list) (cons tc (cdr tc-list)))
          (cons (car tc-list) (insert-tc tc (cdr tc-list))))))
```

4 その他の修正

この§4では、枠組みに関する細かい修正事項を、まとめて述べる。

まず、関数 tone(岡田(1992:128)参照)の名称を、tone-patternに変更する。tone という名称は、素性の名称である tone と、同一である。このことに、実害はない。しかし、重要な名称に衝突があるのは、好ましくない。よって、変更する。

次に、関数 proc(岡田(1992:136)参照)へ受け渡す情報について、変更を加える。item, dialect

以外の情報は、関数 proc へ受け渡さないこととする。

これにより、関数 proc において、残余パラメーターの設定は、不要になる。また、item が、
make-index-list, contraction1, contraction2,
variation, modification, surface-rule

のいずれかのとき、proc は、関数を返すことになる。

この変更に伴い、関数 proc と関連する部分について、すでに岡田(1992,1993a,1993b)で示したプログラムを、書き直す必要が生ずる。しかし、プログラムのみに関する事項に過ぎないので、省略する。

最後に、合算の操作の適用のしかたを変更する。従来、表層規則の適用直後は、合算を適用しないこととしていた(岡田(1992:117-118)参照)。しかし、この方式では、今後、不都合の生ずる恐れがある。そこで、表層規則の適用直後にも、合算を適用することとする。

5 甑島方言主流アクセント

5.1 データ

具体的な音調体系の分析に移る。上村孝二(1941)のデータに基づき、この§5と次の§6において、それぞれ、甑島方言主流アクセントと上甑島江石方言の音調を、取り上げる。

この2つの音調体系については、Haraguchi(1977:216-222)と、上野善道(1984:364-367)が、それぞれの立場から、分析を行っている。

すでに、§0で述べたとおり、甑島方言主流アクセントに対して示す2つの分析のうち的一方は、音調変動挿入関数の導入 (§3参照)によって、初めて、容易に定式化できるようになった。

なお、「主流アクセント」という語は、上村孝二(1941)の用語を、そのまま用いたものに過ぎない。このような意味で「アクセント」という語を使用することは、本稿の本来の用語法には、反する。

甑島方言主流アクセントの体系を、次に挙げる。語例としては、上野善道(1984:367)の(21)と同じものを用いる。2つの型を、素性Tを用いて、[1 T], [2 T]で表す。それぞれ、上野氏の用いているA, Bに対応する。なお、語音の性質に基づく音調の変異については、省略する。

[1 T]

イシ オナゴ カマボコ ニギーメシ サクラノハナ

[2 T]

ヤマ アタマ アサガオ イナビカイ イロエンピツ

5.2 分析

甑島方言主流アクセントについては、2つ、分析を示す。2つの分析の違いは、「中」の扱いにある。直観的に言えば、第1の分析においては、初めから、「中」を設定する。第2の分析においては、初めは「中」を設けず、規則によって、「高」を「中」に変える。縮約の適用については、いずれの分析も、同じである。以下、2つの分析を、それぞれ、「分析1」「分析2」と称する。

[1 T]の基本的な音調パターンは、分析1では、「低中低…低高低」である。分析2では、「低高低…低高低」である。4モーラのパターンは、5モーラのパターンの第1モーラの除去によって得られる。3モーラのパターンは、4モーラのパターンの第1モーラの除去によって得られる。2モーラのパターンは、3モーラのパターンの第1モーラの除去によって得られる。

[2 T]の基本的な音調パターンは、分析1では、「低中低…低高」である。分析2では、「低高低…低高」である。3モーラのパターンは、4モーラのパターンの第1モーラの除去によって得られる。2モーラのパターンは、3モーラのパターンの第1モーラの除去によって得られる。

直観的に記述すると、縮約が、すべて、「第1モーラの除去」となる。しかし、定式化の際に、そのようにまとめることはできない。ただし、縮約に関する事項は、[1 T]と[2 T]に共通である。このことは、定式化に反映される。

分析2においては、「高」が2つ存在する場合、表層規則によって、第1の「高」を、「中」に変える。例えば、[1 T]の6モーラのパターンにおいては、「低高低低高低」を、「低中低低高低」に変える。[1 T]の4モーラのパターン、及び、[2 T]の3モーラのパターンの場合は、それぞれ、「高低高低」「高低高」を、「中低高低」「中低高」に変えることになる。このとき、音調変動挿入関数 insert-tc (§3参照) を、使用する。

この表層規則は、直観的に言えば、特定の「高」を、無条件で、「中」に変えるものである。すなわち、特定の音調変動の音調変動セグメント成分を、無条件で変更することに相当する（「高」が2つ存在する場合、という条件はある。しかし、「高」が1つしか存在しない場合、その「高」は、言わば、第2の「高」が残ったものである。すなわち、「中」に変わる可能性のない「高」である）。

このような、音調変動セグメントの「無条件の変更」という操作は、本稿の枠組みとしては、異例である。これを乱用すると、枠組みの弱体化を招くことになる。よって、今までの手法に沿っているという点では、分析1の方が、より妥当である。

いずれにしても、分析1と分析2の妥当性に、大きな差はない。また、そもそも、複数の分析の妥当性を論ずることを、それほど意味のあることとは考えない。よって、両分析を併記したままにしておく。

なお、Haraguchi(1977: 216-219)と上野善道(1984: 366-367)における、「中」の扱いは、互いに、類似している(両氏の依拠する枠組み自体は、互いに、全く異なる。ここでは、特定の現象に対する基本的な見方のみを、局所的に取り上げているに過ぎない)。かつ、ここで言う分析1、分析2のいずれとも、異なっている。直観的に言えば、両氏の方式においては、初めは、「中」に当たるものを設けず、あとから、「低」を「中」に変える。

中甌島方言の体系を考慮に入れ、かつ、特に通時的な考察を加えるならば(上野善道(1984: 365-366)参照)、両氏のような方式が、より妥当であるとも考えられる。仮に、本稿の枠組みにおいて、この方針で定式化を行うとすると、「中」に相当する音調変動を、あとから、挿入することになる。音調変動の挿入自体は、本稿の§3において、新しい関数を導入したことにより、容易になった。

しかし、本稿の枠組みにおいては、通時的な考慮を、一切、加えないこととしている。また、それとは別に、枠組み全体の方針上、音調変動の挿入を、このような場合にまで認めることはできない。

音調変動の挿入は、今回の、先頭の「高」を「中」に変える例のように、「実質的には、音調変動セグメントの変更であるにもかかわらず、形式上、音調変動を挿入するという形でしか実施できない」という場合にのみ、認めることとする。したがって、本稿の枠組みにおいて、両氏のような方式をとることはできない。

6 上甌島江石方言

次に、上甌島江石方言を、取り上げる。主流アクセントとは異なり、定式化する上での問題はない。しかし、同じ上村孝二(1941)に記述があり、主流アクセントとの対照という意味もあるので、ついでに扱う。

まず、データを挙げる。主流アクセントと同様の形式とする (§5.1参照)。

[1 T]

イシ オナゴ カマボコ ニギリメシ サクラノハナ

[2 T]

ヤマ アタマ アサガオ イナビカリ イロエンピツ

分析に移る。通常どおり、1つだけ、分析を示す。

[1 T]の基本的な音調パターンを、「低低高低…低高低」とする。5モーラのパターンは、6モーラのパターンの第2モーラの除去によって得られる。4モーラのパターンは、5モーラのパターンの第1モーラの除去によって得られる。3モーラのパターンは、4モーラのパターンの第1モーラの除去によって得られる。2モーラのパターンは、3モーラのパターンの第1モーラの除去によって得られる。

[2 T]の基本的な音調パターンを、「低低高低…低高」とする。4モーラのパターンは、5モーラのパターンの第2モーラの除去によって得られる。3モーラのパターンは、4モーラのパターンの第1モーラの除去によって得られる。2モーラのパターンは、3モーラのパターンの第1モーラの除去によって得られる。

主流アクセントの場合と同じく (§5.2参照)、縮約に関する事項は、[1 T]と[2 T]に共通である。そして、やはり同じく、このことは、定式化に反映される。

「中」は、「高」から導く。「高」が2つ存在する場合、表層規則によって、第2の「高」を、「中」に変える。音調変動挿入関数の使用はない。

主流アクセントの分析2の表層規則 (§5.2参照)とは異なり、この操作は、特定の音調変動に由来する「高」を無条件に「中」に変えるものではない。したがって、この操作が枠組みの弱体化を招くことはない。

7 鹿児島方言

鹿児島方言を、ごく簡単に扱う。鹿児島方言を取り上げる目的は、甌島方言主流アクセント (§5)、及び、上甌島江石方言 (§6)と、対照させることにある。また、中甌島方言の音調体系は、モーラ単位ではあるものの、基本的には、鹿児島方言の音調体系と同じである (上村孝二

(1941: 15), 上野善道(1984: 366)参照)。

データを示す。語例は省略する。

[1 T]

ò ̄ò ̄òò ̄òòò ̄òòòò ̄òòòòò ̄òòòòòò

[2 T]

̄ò ̄òò ̄òòò ̄òòòò ̄òòòòò ̄òòòòòò

[1 T] の基本的な音調パターンは、「低…低高低」である。1 モーラのパターンは、2 モーラのパターンの圧縮によって得られる。

[2 T] の基本的な音調パターンは、「低…低高」である。縮約の適用はない。

8 Scheme による定式化

最後に、Scheme による定式化を行う。

甑島方言主流アクセントに対する2つの分析 (§5.2参照) は、形式上、異なる2つの方言に対する分析であるかのように扱った。すなわち、「方言名」を、第1の分析については、koshikijima-shuryuu とし、第2の分析については、koshikijima-shuryuu-2 とした。

定式化した結果を、前稿と同じく、関数 proc に付け加える部分という形で、本稿の末尾に示す。

参考文献

- 上野善道(1984)「新潟県村上方言のアクセント」『金田一春彦博士古希記念論文集第二巻言語学編』三省堂。347-390.
- 岡田英俊(1988)「生成音韻論による京都方言の音調の分析」『東京大学言語学論集 '88』279-292.
- 岡田英俊(1992)「日本語諸方言の音調体系の定式化(3)」『金沢大学教養部論集人文科学篇』30(1). 113-156.
- 岡田英俊(1993a)「日本語諸方言の音調体系の定式化(4)」『金沢大学教養部論集人文科学篇』30(2). 139-164.
- 岡田英俊(1993b)「日本語諸方言の音調体系の定式化(5)」『金沢大学教養部論集人文科学篇』31(1). 35-48.
- 上村孝二(1941)「甑島方言のアクセント」『音声学協会会報』65/66. 12-15.
- Haraguchi, Shosuke (1977) *The Tone Pattern of Japanese: An Autosegmental Theory of Tonology*. Tokyo: Kaitakusha.

;; 甌島主流方言(分析1)

((koshikijima-shuryuu)

(case item

((tc-list-init)

'(e1 0 (0 . 1) 0)
 (t01 0 (0 . 1) -4)
 (t02 0 (1 . 1) 2)
 (t03 0 (2 . 1) -2)
 (t11 2 (-2 . 1) 4)
 (t12 2 (-1 . 1) -4)
 (t2 2 (-1 . 1) 4)
 (e2 2 (0 . 1) 0)))

((make-index-list)

(lambda (fv-list)

(append

'(t01 t02 t03)

(case (val 'tone fv-list)

((1) '(t11 t12))

((2) '(t2))))))

((total-compression) #f)

((prohibition) '())

((contraction1)

(lambda (fv-list border) (list tautology)))

((contraction2)

(lambda (fv-list border)

(cond ((or (equal? border '(t03 . t11))

(equal? border '(t03 . t2)))

(list not-touch?

delete 't02

delete 't03

delete 't11))

(else (list tautology))))))

((variation) (lambda (fv-list tc-list) tc-list))

((modification) (lambda (fv-list tc-list) tc-list))

((surface-rule) (lambda (fv-list tc-list) tc-list))))

```

;; 甌島主流方言(分析2)
((koshikijima-shuryuu-2)
 (case item
  ((tc-list-init)
   '((e1 0 (0 . 1) 0)
     (t01 0 (0 . 1) -4)
     (t02 0 (1 . 1) 4)
     (t03 0 (2 . 1) -4)
     (t11 2 (-2 . 1) 4)
     (t12 2 (-1 . 1) -4)
     (t2 2 (-1 . 1) 4)
     (e2 2 (0 . 1) 0)))
  ((make-index-list)
   (lambda (fv-list)
    (append
     '(t01 t02 t03)
     (case (val 'tone fv-list)
      ((1) '(t11 t12))
      ((2) '(t2))))))
  ((total-compression) #f)
  ((prohibition) '())
  ((contraction1)
   (lambda (fv-list border) (list tautology)))
  ((contraction2)
   (lambda (fv-list border)
    (cond ((or (equal? border '(t03 . t11))
              (equal? border '(t03 . t2)))
          (list not-touch?
                delete 't02
                delete 't03
                delete 't11))
          (else (list tautology)))))
  ((variation) (lambda (fv-list tc-list) tc-list))
  ((modification) (lambda (fv-list tc-list) tc-list))
  ((surface-rule)
   (lambda (fv-list tc-list)
    (cond ((and (specified? 't03 tc-list)
               (rat>=? (displacement '(e1 . t03) tc-list)

```

```

      '(1 . 1)))
    (set! tc-list
      (map (lambda (tc)
            (if (eq? (index-comp tc) 't03)
                (subst-segment -2 tc)
                tc))
          tc-list))
    (if (specified? 't02 tc-list)
        (set! tc-list
          (map (lambda (tc)
                (if (eq? (index-comp tc) 't02)
                    (subst-segment 2 tc)
                    tc))
              tc-list))
        (set! tc-list
          (insert-tc '(t02 0 (0 . 1) -2) tc-list)))
    tc-list)
  (else tc-list))))))
;; 上甌島江石方言
((kamikoshikijima-eishi)
 (case item
  ((tc-list-init)
   '((e1 0 (0 . 1) 0)
     (t01 0 (0 . 1) -4)
     (t02 0 (2 . 1) 4)
     (t03 0 (3 . 1) -4)
     (t11 2 (-2 . 1) 4)
     (t12 2 (-1 . 1) -4)
     (t2 2 (-1 . 1) 4)
     (e2 2 (0 . 1) 0)))
  ((make-index-list)
   (lambda (fv-list)
     (append
      '(t01 t02 t03)
      (case (val 'tone fv-list)
        ((1) '(t11 t12))
        ((2) '(t2))))))
  ((total-compression) #f)

```

```

((prohibition) '())
((contraction1)
 (lambda (fv-list border) (list tautology)))
((contraction2)
 (lambda (fv-list border)
  (cond ((or (equal? border '(t03 . t11))
            (equal? border '(t03 . t2)))
        (list not-touch?
              delete 't02
              delete 't02
              delete 't03
              delete 't11))
        (else (list tautology))))))
((variation) (lambda (fv-list tc-list) tc-list))
((modification) (lambda (fv-list tc-list) tc-list))
((surface-rule)
 (lambda (fv-list tc-list)
  (if (and (specified? 't03 tc-list)
          (rat>? (displacement '(e1 . t03) tc-list)
                '(0 . 1)))
      (map (lambda (tc)
            (cond ((eq? (index-comp tc) 't11)
                  (subst-segment 2 tc))
                  ((eq? (index-comp tc) 't12)
                  (subst-segment -2 tc))
                  ((eq? (index-comp tc) 't2)
                  (subst-segment 2 tc))
                  (else tc)))
          tc-list)
      tc-list))))))

```

```

;; 鹿児島方言
((kagoshima)
 (case item
  ((tc-list-init)
   '((e1 0 (0 . 1) 0)
     (t0 0 (0 . 1) -4)
     (t11 2 (-2 . 1) 4)
     (t12 2 (-1 . 1) -4)
     (t2 2 (-1 . 1) 4)
     (e2 2 (0 . 1) 0)))
  ((make-index-list)
   (lambda (fv-list)
    (append
     '(t0)
     (case (val 'tone fv-list)
      ((1) '(t11 t12))
      ((2) '(t2))))))
  ((total-compression) #f)
  ((prohibition) '())
  ((contraction1)
   (lambda (fv-list border) (list tautology)))
  ((contraction2)
   (lambda (fv-list border)
    (cond ((equal? border '(t0 . t11))
           (list not-cross?
                 compress 't12))
          (else (list tautology)))))
  ((variation) (lambda (fv-list tc-list) tc-list))
  ((modification) (lambda (fv-list tc-list) tc-list))
  ((surface-rule) (lambda (fv-list tc-list) tc-list)))

```