

TR-I-0073 (appendix)

スペクトログラム・リーディングに基づく  
音韻セグメンテーション知識(付録)

Phoneme segmentation knowledge  
based on spectrogram reading(appendix)

畑崎香一郎 小森康弘  
Kaichiro Hatazaki, Yasuhiro Komori

1989.3

### 概要

スペクトログラム・リーディングの手法および知識を用いて、連続音声中の音韻セグメンテーションを行うエキスパートシステムを構築した。本編は報告書TR-I-0073「スペクトログラム・リーディングに基づく音韻セグメンテーション知識」の付録である。

ATR 自動翻訳電話研究所

ATR Interpreting Telephony Research Laboratories

© (株)ATR 自動翻訳電話研究所 1989

© 1989 by ATR Interpreting Telephony Research Laboratories

付録2

Source

Z.M06:>Komori>sre>globals>defglobals.art.17

For: Yasuhiro Komori

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 12/01/88 12:02:00

Queued at: 3/31/89 10:17:53

```
;;;-- Mode: ART; Package: art-user; Base: 10. --
```

```
(defglobal
  ?*no-power-upper* = -45 ; Upper limit so that one can
                        ; say here is no-power.

  ?*vowel-mid-power-lower* = -45
  ?*voicing-power-min* = -45 ;[hatazaki 135] <- -50 ; -45 ; MAU_1_0146 last /su/, -45.2
  ?*voicing-power-min-before-fricative* = -45
  ?*voicing-power-lower* = -45 ;[hatazaki 102] ; For MAU_B_0146 right of /sh/, -30.99
  ?*high-freq-power-lower* = -45
  ?*closure-power-upper* = -50 ; -45
  ?*imcomplete-closure-power-upper* = -35
  ?*silence-power-upper* = -70 ;[hatazaki 100] ; Changed <- -45 [hatazaki 7/21/88 19:36:47]
  ?*fricative-middle-power-upper* = -35
  ?*cutoff-th* = -35
  ?*fricative-cutoff-lower* = 900)
```

```
(defglobal
  ?*default-min-change* = 0.5 ; 1.0

  ?*vowel-F2-end-min-change* = 1.0
  ?*vowel-F2-end-min-change-smaller* = 0.2
  ?*vowel-F2-start-min-change* = 1.0
  ; ?*high-freq-power-start-min-change* = 0.5 ; 0.8 ; (MAU_B_0181 /h/ : 0.99)

  ?*closure-start-min-change* = 1.0
  ?*closure-end-min-change* = 0.5 ; 1.0
```

```
;;[hatazaki 145] Renamed to ?*vowel-start-min-change-after-unvoiced-fricative*
;; defined in seg-unvoiced-fricative-globals.art
;;
;; ?*vowel-start-min-change* = 1.5 , ;<- 1.0 ; 2.0
;; ;(cf. MAU_B_0029 /s/)
;; ; [hatazaki 6/22/88 19:48:06]
;)
```

```
(defglobal
  ?*closure-min-duration* = 30 ; cf. MAU_B_0122 (Not closure)
```

```
(defglobal
  ?*normal-smoothing-size* = (seq$ '(11 9))
  ?*spectrum-global-peak-moving-average-size* = 25)
```

```
(defglobal
  ?*high-freq-top* = 6000
  ?*high-freq-bottom* = 4000
  ?*middle-freq-top* = 4000
  ?*middle-freq-bottom* = 1000
  ?*low-freq-top* = 1000
  ?*low-freq-bottom* = 0)
```

```
(defglobal
  ?*F3-top* = 3000
  ?*F3-bottom* = 2000
  ?*F2-top* = 2000
  ?*F2-bottom* = 1000
  ?*F1-top* = 1000
  ?*F1-bottom* = 0)
```

```
(defglobal
  ?*CFvalid* = -0.2
  ?*CFstrongly-valid* = 0.6)
```

LM06: >Komori>sre>seg-unvoiced-stop>seg-unvoiced-stop-globals. art. 21

For: Yasuhiro Komori

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 11/14/88 15:09:30

Queued at: 3/31/89 10:27:48

```
;;; -- Mode: ART; Base: 10.; Package: ART-USER --

(defglobal
  ?*normalize-factor* = (seq$ '(5 0))
  ?*head-burst-min-change* = 1.5
  ?*burst-strength-lower-upper* = -70
  ?*high-freq-burst-power-upper* = -30
  ?*high-freq-vowel-power-upper* = -55
  ?*low-freq-vowel-power-upper* = -45
  ?*low-freq-power-start-min-change* = -50)

(defglobal ; By hatazaki.
  ?*smoothing-for-finding-burst* = (seq$ '(5 0))
  ?*smoothing-for-finding-burst-end* = (seq$ '(0 0)) ;[hatazaki 127]
  ?*spectrum-peak-smoothing-for-getting-burst* = (seq$ '(0 7))

  ?*voicing-power-after-stop-lower* = -40 ; [hatazaki 135] <- -50.
  ?*vowel-1000-4000-power-before-stop-lower* = -40
  ?*speech-region-1000-4000-lower* = -70
  ;;?*closure-power-upper* = -50 ; Defined in defglobals.art

  ?*burst-freq-min* = 700
  ?*burst-freq-max* = 4000

  ?*vowel-start-min-change-after-unvoiced-stop* = 2.0
  ?*burst-1000-4000-start-min-change* = 1.5 ;[hatazaki 131]
  ?*burst-start-min-change* = 3.0 ;;2.0 ;1.0
  ?*burst-end-min-change* = 3.0 ;; <-- 1.0
  ;;; <-- 0.0
  ;;; <-- 1.0 referred MAU_B_0026 /k/
  ; [hatazaki 109]
  ?*fricative-start-min-change-after-unvoiced-stop* = 1.0
)
```

LM06: >Komori>sre>seg-unvoiced-fricative>seg-unvoiced-fricative-globals.a  
rt.7

For: Yasuhiro Komori

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 12/08/88 21:46:53

Queued at: 3/31/89 10:42:29

;;; -\*- Mode: ART; Base: 10; Package: ART-USER -\*-

```
(defglobal
  ?*frication-start-min-change* = 0.5          ; - This includes for /h/.
                                           ; (MAU_B_0181 /h/ 0.99)
```

```
?*vowel-start-min-change-after-unvoiced-fricative* = 1.5
```

```
?*frication-start-min-change-for-small-smoothing* = 2.0
)
```

```
(defglobal
```

```
?*voicing-power-lower-after-unvoiced-fricative* = -40
```

```
?*voicing-power-smaller-lower-after-unvoiced-fricative* = -55
```

```
)
```

```
(defglobal
```

```
?*smoothing-for-getting-power-change-of-uvfricative-at-word-initial* = (seq$ '(5 0))
```

```
)
```



LM06: >Komori>sre>SEG-H>seg-h-globals. art. 9

For: Yasuhiro Komori

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/02/89 15:52:24

Queued at: 3/31/89 10:18:24

```
;;;-*- Mode: ART; Package: art-user; Base: 10. -*-  
(defglobal  
  ?*h-fricative-start-change-2000-6000* = 0.8  
  
  ?*h-left-is-silence-power* = -65  
  ?*h-right-is-silence-power* = -65  
  ?*spectrum-peak-smoothing-for-phoneme-h* = (seq$ '(0 7))  
  
  ?*vowel-start-min-change-after-h* = 0.5  
  ?*vowel-end-min-change-before-h* = 0.5  
  
  ?*h-start-min-change* = 0.5  
  ?*h-end-min-change* = 0.5  
  
  ?*vowel-formant-start-min-change-after-h* = 0.2  
  ?*vowel-formant-end-min-change-before-h* = 0.2  
)  
;;; (middle h-region-candidate ?from&?? ?to&??) is goal  
(defrelation h-region-candidate  
  (?from ?to)  
  implicit  
  (syntax output  
    (MIDDLE H region candidate is ?from to ?to)))
```

LM06: >Komori>sre>seg-voiced-fricative>seg-voiced-fricative-globals.art.3

For: Yasuhiro Komori

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 12/01/88 20:20:37

Queued at: 3/31/89 10:30:01

```
;;;-*- Mode: ART; Package: art-user; Base: 10. -*-
```

```
(defglobal
```

```
?*vowel-start-min-change-after-voiced-fricative* = 0.2 ; (cf. MAU_B_0069 /zu/)
```

```
;;[hatazaki 168] 0.5 <-- 0.2, to ignore an extra power decrease in /z/ region.
```

```
;; (cf. MAU_B_0086 has an extra power decrease (0.21).)
```

```
;; (cf. MAU_1_0174 has a desired power decrease followed by an extra power decrease (0.3).
```

```
;;[] 0.2 ; (cf. MAU_1_0174 0.3)
```

```
;;[] 0.6 ; (cf. MAU_B_0105 1.0)
```

```
;;
```

```
?*vowel-end-change-lower-before-voiced-fricative* = 0.5
```

```
; To ignore 0.21 (cf. MAU_B_0086)
```

```
;
```

```
; 0.2 ; (cf. MAU_1_0174 0.3)
```

```
; 0.6 ; (cf. MAU_B_0105 1.0)
```

```
?*z-after-silence-power-change-lower* = 1.0
```

```
?*dj-smoothing-for-finding-burst* = (seq$ '(5 0))
```

```
?*dj-burst-end-min-change* = 2.0
```

```
?*dj-burst-start-min-change* = 2.0
```

```
)
```

LM06: >Komori>sre>seg-voiced-stop>seg-voiced-stop-globals. art. 14

For: Yasuhiro Komori

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/18/89 17:31:52

Queued at: 3/31/89 10:30:28

```
;;;-- Mode: ART; Package: art-user; Base: 10. -*-
```

```
(defglobal
  ?*vowel-start-min-change-after-voiced-stop* = 1.0
  ;; ?*vowel-start-min-change-after-voiced-stop* = 0.2
  ?*vowel-end-min-change-before-voiced-stop* = 0.2

  ?*voiced-stop-start-min-change* = 0.2
  ?*voiced-stop-end-min-change* = 0.2
)

;;; (voiced-stop-region-candidate ?from&?? ?to&??) is goal
(defrelation voiced-stop-region-candidate
  (?from ?to)
  implicit
  (syntax output
    (Voiced-stop region candidate is ?from to ?to)))

;;; (voiced-stop-peak-candidate ?pos&?? ?value&??) is goal
(defrelation voiced-stop-peak-candidate
  (?pos ?value)
  implicit
  (syntax output
    (Voiced-stop peak candidate is at ?pos with ?value)))
```

```
-----

(defglobal
  ?*normalize-factor* = (seq$ '(5 0))
  ?*head-voiced-stop-burst-min-change* = 1.5
  ?*voiced-stop-burst-strength-lower-upper* = -70
  ?*high-freq-voiced-stop-burst-power-upper* = -30
  ?*high-freq-vowel-power-upper* = -55
  ?*low-freq-vowel-power-upper* = -45
  ?*low-freq-power-start-min-change* = -50)

(defglobal ; By hatazaki.
  ?*smoothing-for-finding-voiced-stop-burst* = (seq$ '(5 0))
  ?*smoothing-for-finding-voiced-stop-burst-end* = (seq$ '(0 0)) ;[hatazaki 127]
  ?*spectrum-peak-smoothing-for-getting-voiced-stop-burst* = (seq$ '(0 7))

  ?*voicing-power-after-stop-lower* = -40 ; [hatazaki 135] <- -50.
  ?*vowel-1000-4000-power-before-stop-lower* = -40
  ?*speech-region-1000-4000-lower* = -70
  ;; ?*closure-power-upper* = -50 ; Defined in defglobals.art

  ?*voiced-stop-burst-freq-min* = 1000
  ?*voiced-stop-burst-freq-max* = 6000

  ?*voiced-stop-burst-1000-6000-start-min-change* = 1.0 ;[hatazaki 131]
  ?*voiced-stop-burst-start-min-change* = 2.0 ;; 2.0 ; 1.0
  ?*voiced-stop-burst-end-min-change* = 2.0 ;; <-- 1.0
  ;; <-- 0.0
  ;; <-- 1.0 referred MAU_B_0026 /k/
  ; [hatazaki 109]

  ?*fricative-start-min-change-after-voiced-stop* = 1.0
)
```

LM06:>Komori>sre>seg-nasal>seg-nasal-globals.art.12

For: Yasuhiro Komori

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 11/16/88 09:45:43

Queued at: 3/31/89 10:26:49

```

;;-*- Mode: ART; Package: art-user; Base: 10. -*-

(defglobal
  ?*vowel-start-min-change-after-nasal* = 0.2
  ?*vowel-end-min-change-before-nasal* = 0.2

  ?*nasal-start-min-change* = 0.2
  ?*nasal-end-min-change* = 0.2
)

;;; (nasal-region-candidate ?from&?? ?to&??) is goal
(defrelation nasal-region-candidate
  (?from ?to)
  implicit
  (syntax output
    (Nasal region candidate is ?from to ?to)))

;;; (nasal-peak-candidate ?peak-loc&?? ?peak-value&??) is goal
(defrelation nasal-peak-candidate
  (?peak-pos ?peak-value)
  implicit
  (syntax output
    (Nasal peak candidate is at ?peak-pos msec and value is ?peak-value)))

;;;
;;; EXAMPLE of defrelation
;;;
;;; (spectral-change (?pos ?value)
;;;                 =(- (min ?nasal-end ?vowel-start) 10)    is goal
;;;                 =(+ (max ?nasal-end ?vowel-start) 10)
;;;                 0 6000 (5 5) 7.5 0.0 0.2 0.5)
;;;
;;; (defrelation spectral-change
;;;   (?time-val ?from ?to ?bottom ?top ?smoothing-size
;;;     ?interval ?apart ?change_allowance ?power_min)
;;;   implicit
;;;   (syntax output
;;;     (Spectrum changes at ?time-val between ?from to ?to msec and ?bottom to ?top Hz
;;;       with smoothing ?smoothing-size interval ?interval apart ?apart
;;;       change_allowance ?change_allowance power_min ?power_min)))
;;;
;;;

```



LM06: >Komori>sre>seg-liquid>seg-liquid-globals.art.5

For: Yasuhiro Komori

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/19/89 10:00:24

Queued at: 3/31/89 10:41:47

```
;;-*- Mode: ART; Package: art-user; Base: 10. -*-
```

```
(defglobal
  ?*vowel-start-min-change-after-liquid* = 0.5
  ?*vowel-end-min-change-before-liquid* = 0.5

  ?*liquid-start-min-change* = 0.5
  ?*liquid-end-min-change* = 0.5
)
```

```
;;; (middle liquid-region-candidate ?from&?? ?to&??) is goal
(defrelation mid-liquid-region-candidate
  (?from ?to)
  implicit
  (syntax output
    (MIDDLE LIQUID region candidate is ?from to ?to)))
```

LM17: >hatazaki>sre>seg-unvoiced-stop>seg-stop.art.103

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/24/89 15:44:36

Queued at: 3/28/89 10:57:13

```

;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-

(defrelation UNVOICED-STOP-CANDIDATE (?from ?to)
  implicit
  (syntax output
   (Unvoiced stop candidate at ?from - ?to)))

(defrule sg-uvstop-0
  "Finds a candidate of unvoiced-stop segment."
  (declare (salience ?*find-candidate*))
  (segmenting unvoiced-stop)
  (UNVOICED-STOP-CANDIDATE ?s1-from ?s1-to)
  (CF (unvoiced-stop closure-duration =(- ?s1-to ?s1-from)) ?CFduration&:mightbe-valid)
  (power-strength ?s1-from ?s1-to 0 500 ?0-500-P)
  (CF (unvoiced-stop closure-0-500-power ?0-500-P) ?CF-0-500-P&:mightbe-valid)
  (power-strength ?s1-from ?s1-to 500 1000 ?500-1000-P)
  (CF (unvoiced-stop closure-0-500-500-1000-power-ratio =(- ?0-500-P ?500-1000-P))
   ?CF-0-500-500-1000-PR&:mightbe-valid)
  (power-strength ?s1-from ?s1-to 0 6000 ?0-6000-P)
  (CF (unvoiced-stop closure-0-6000-power ?0-6000-P) ?CF-0-6000-P&:mightbe-valid)
  =>
  (bind ?segment (gensym "SEG-"))
  (assert (schema ?segment (instance-of phoneme-segment)))
  (hypothesize
   (assert (category ?segment unvoiced-stop))
   (assert (prop ?segment (candidate-loc ?s1-from ?s1-to))
    (segment-status ?segment segmentation)
    ;;
    ;; [hatazaki 143] CF of (category ?segment unvoiced-stop),
    ;; -->
    ;; =(CFcomb (CFweight ?*weak-evidence* ?CFduration)
    ;;          (CFweight ?*evidence*
    ;;                    (CFand ?CF-0-500-P
    ;;                          ?CF-0-500-500-1000-PR
    ;;                          ?CF-0-6000-P))))
    ;;
    ;; [hatazaki 143] CF of (category ?segment unvoiced-stop),
    ;; =(CFweight ?*evidence*
    ;;          (CFand ?CFduration
    ;;                ?CF-0-500-P
    ;;                ?CF-0-500-500-1000-PR
    ;;                ?CF-0-6000-P))
    ;; -->
    ;; =(CFcomb (CFweight ?*weak-evidence* ?CFduration)
    ;;          (CFweight ?*weak-evidence* ?CF-0-500-P)
    ;;          (CFweight ?*weak-evidence* ?CF-0-500-500-1000-PR)
    ;;          (CFweight ?*weak-evidence* ?CF-0-6000-P))
    ;;
    (CF (category ?segment unvoiced-stop)
      =(CFand (CFcomb (CFunknown)
                    (CFweight ?*weak-evidence* ?CFduration)
                    (CFweight ?*weak-evidence* ?CF-0-500-P)
                    (CFweight ?*weak-evidence* ?CF-0-500-500-1000-PR)
                    (CFweight ?*weak-evidence* ?CF-0-6000-P))
          ?CF-0-500-P
          ?CF-0-500-500-1000-PR
          ?CF-0-6000-P))))))

;;; New definition. [hatazaki 176]
;(defrule sg-uvstop-1a
  "Finds an unvoiced stop candidate at word initial (with burst)."
  ; (declare (salience ?*find-candidate*))
  ; (segmenting unvoiced-stop)
  ; (UNVOICED-STOP-CANDIDATE ?s1-from ?s1-to)
  ; (not (UNVOICED-STOP-CANDIDATE ?s2&:(?s2 < ?s1-from) ?))
  ; ;;
  ; ;; [hatazaki 176] Searches a burst from more left, since ?s1-to may be at the end of /k/.
  ; ;; after =(- ?s1-to 100) 130
  ; ;; <-- after =(- ?s1-to 50) 80
  ; ;; [hatazaki 132] Removed the rule added by modification below.
  ; ;; (power-start ?speech-start&~NONE
  ; ;; before =(+ ?s1-to 20) 70 1000 4000 ?*speech-region-1000-4000-lower*)
  ; ;; [hatazaki 132] Searches a burst only in speech region, since there may be power change
  ; ;; in the preceding silent region.
  ; ;;
  ; ;; [hatazaki 114].

```

```

; ;;
; (burst-exists ?burst-start ?burst-end ?burst-freq
;   after =(- ?s1-to 100) 130 ?*burst-freq-min* ?*burst-freq-max*
;   ?CF-burst&: mightbe-valid)
; (not (burst-exists ? ? ?
;   after =(- ?s1-to 100) 130 ?*burst-freq-min* ?*burst-freq-max*
;   ?another-burst&:(CFgreaterp ?another-burst ?CF-burst)))
; (power-strength =(- ?burst-start 30) ?burst-start 0 500 ?0-500-P)
; (CF (unvoiced-stop closure-0-500-power ?0-500-P) ?CF-0-500-P&: mightbe-valid)
; (power-strength =(- ?burst-start 30) ?burst-start 500 1000 ?500-1000-P)
; (CF (unvoiced-stop closure-0-500-500-1000-power-ratio =(- ?0-500-P ?500-1000-P))
;   ?CF-0-500-500-1000-PR&: mightbe-valid)
; (power-strength =(- ?burst-start 30) ?burst-start 0 6000 ?0-6000-P)
; (CF (unvoiced-stop closure-0-6000-power ?0-6000-P) ?CF-0-6000-P&: mightbe-valid)
; ;;
; ;;[hatazaki 153] Commented out the following.
; ;; ;;
; ;; ;;[hatazaki 149] after =(+ ?burst-start 10) 200 <-- after =(+ ?burst-start 10) 100,
; ;; ;; for /ch/ (MAU_B_0014).
; ;; ;;
; ;; (power-start ?vowel-start&~NONE
; ;;   after =(+ ?burst-start 10) 200 0 500 ?*voicing-power-lower*)
; =>
; (bind ?segment (gensym "SEG-"))
; (assert (schema ?segment (instance-of phoneme-segment)))
; (hypothesize
;   (assert (category ?segment unvoiced-stop))
;   (assert (segment-status ?segment segmentation)
;     ;;
;     ;; [hatazaki 143] CF of (category ?segment unvoiced-stop),
;     ;; -->
;     ;; =(CFcomb (CFweight ?*strong-evidence* ?CF-burst)
;     ;;   (CFweight ?*weak-evidence*
;     ;;     (CFand ?CF-0-500-P ?CF-0-500-500-1000-PR ?CF-0-6000-P)))
;     ;;
;     ;; [hatazaki 143] CF of (category ?segment unvoiced-stop),
;     ;; =(CFweight ?*evidence*
;     ;;   (CFand ?CF-burst ?CF-0-500-P ?CF-0-500-500-1000-PR ?CF-0-6000-P)))
;     ;; -->
;     ;; =(CFcomb (CFweight ?*strong-evidence* ?CF-burst)
;     ;;   (CFweight ?*weak-evidence* ?CF-0-500-P)
;     ;;   (CFweight ?*weak-evidence* ?CF-0-500-500-1000-PR)
;     ;;   (CFweight ?*weak-evidence* ?CF-0-6000-P)))
;     ;;
;     (CF (category ?segment unvoiced-stop)
;       =(CFand (CFcomb (CFunknown)
;         (CFweight ?*strong-evidence* ?CF-burst)
;         (CFweight ?*weak-evidence* ?CF-0-500-P)
;         (CFweight ?*weak-evidence* ?CF-0-500-500-1000-PR)
;         (CFweight ?*weak-evidence* ?CF-0-6000-P))
;         ?CF-burst ?CF-0-500-P ?CF-0-500-500-1000-PR ?CF-0-6000-P))
;       (has-burst ?segment yes)
;       (CF (has-burst ?segment yes) ?CF-burst)
;       (left-context ?segment word-initial)
;       (CF (left-context ?segment word-initial) 0.4)
;       ;;(prop ?segment (location word-initial))
;       (prop ?segment (candidate-loc UNKNOWN ?burst-start))
;       (prop ?segment (burst ?burst-start ?burst-end ?burst-freq))))))
; ;; Old definition.
; (defrule sg-uvstop-1a
;   "Finds an unvoiced stop candidate at word initial (with burst)."
;   (declare (salience ?*find-candidate*))
;   (segmenting unvoiced-stop)
;   (UNVOICED-STOP-CANDIDATE ?s1-from ?s1-to)
;   (not (UNVOICED-STOP-CANDIDATE ?s2&:(?s2 < ?s1-from) ?))
;   ;;
;   ;; [hatazaki 132] Removed the rule added by modification below.
;   ;; (power-start ?speech-start&~NONE
;   ;;   before =(+ ?s1-to 20) 70 1000 4000 ?*speech-region-1000-4000-lower*)
;   ;; [hatazaki 132] Searches a burst only in speech region, since there may be power change
;   ;; in the preceding silent region.
;   ;;
;   ;; [hatazaki 114].
;   ;;
;   ;;
;   ;;

```

```

(burst-exists ?burst-start ?burst-end ?burst-freq
  after =(- ?s1-to 50) 80 ?*burst-freq-min* ?*burst-freq-max*
  ?CF-burst&: mightbe-valid)
(not (burst-exists ? ? ?
  after =(- ?s1-to 50) 80 ?*burst-freq-min* ?*burst-freq-max*
  ?another-burst&:(CFgreaterp ?another-burst ?CF-burst)))
(power-strength =(- ?burst-start 30) ?burst-start 0 500 ?0-500-P)
(CF (unvoiced-stop closure-0-500-power ?0-500-P) ?CF-0-500-P&: mightbe-valid)
(power-strength =(- ?burst-start 30) ?burst-start 500 1000 ?500-1000-P)
(CF (unvoiced-stop closure-0-500-500-1000-power-ratio =(- ?0-500-P ?500-1000-P)
  ?CF-0-500-500-1000-PR&: mightbe-valid)
  ?CF-0-500-500-1000-PR)
(power-strength =(- ?burst-start 30) ?burst-start 0 6000 ?0-6000-P)
(CF (unvoiced-stop closure-0-6000-power ?0-6000-P) ?CF-0-6000-P&: mightbe-valid)
;;
;;[hatazaki 153] Commented out the following.
;;
;;
;;[hatazaki 149] after =(+ ?burst-start 10) 200 <-- after =(+ ?burst-start 10) 100,
;;
;; for /ch/ (MAU_B_0014).
;;
;;
;; (power-start ?vowel-start&~NONE
;;   after =(+ ?burst-start 10) 200 0 500 ?*voicing-power-lower*)
=>
(bind ?segment (gensym "SEG-"))
(assert (schema ?segment (instance-of phoneme-segment)))
(hypothesize
  (assert (category ?segment unvoiced-stop))
  (assert (segment-status ?segment segmentation)
    ;;
    ;; [hatazaki 143] CF of (category ?segment unvoiced-stop),
    ;; -->
    ;; =(CFcomb (CFweight ?*strong-evidence* ?CF-burst)
    ;;   (CFweight ?*weak-evidence*
    ;;     (CFand ?CF-0-500-P ?CF-0-500-500-1000-PR ?CF-0-6000-P)))
    ;;
    ;; [hatazaki 143] CF of (category ?segment unvoiced-stop),
    ;; =(CFweight ?*evidence*
    ;;   (CFand ?CF-burst ?CF-0-500-P ?CF-0-500-500-1000-PR ?CF-0-6000-P))
    ;; -->
    ;; =(CFcomb (CFweight ?*strong-evidence* ?CF-burst)
    ;;   (CFweight ?*weak-evidence* ?CF-0-500-P)
    ;;   (CFweight ?*weak-evidence* ?CF-0-500-500-1000-PR)
    ;;   (CFweight ?*weak-evidence* ?CF-0-6000-P))
    ;;
    (CF (category ?segment unvoiced-stop)
      =(CFand (CFcomb (CFunknown)
        (CFweight ?*strong-evidence* ?CF-burst)
        (CFweight ?*weak-evidence* ?CF-0-500-P)
        (CFweight ?*weak-evidence* ?CF-0-500-500-1000-PR)
        (CFweight ?*weak-evidence* ?CF-0-6000-P))
        ?CF-burst ?CF-0-500-P ?CF-0-500-500-1000-PR ?CF-0-6000-P))
      (has-burst ?segment yes)
      (CF (has-burst ?segment yes) ?CF-burst)
      (left-context ?segment word-initial)
      (CF (left-context ?segment word-initial) 0.4)
      ;;(prop ?segment (location word-initial))
      (prop ?segment (candidate-loc UNKNOWN ?burst-start))
      (prop ?segment (burst ?burst-start ?burst-end ?burst-freq))))))
(defrule sg-uvstop-2
  "Finds SOKUON."
  (declare (salience ?*find-candidate*))
  (segmenting unvoiced-stop)
  (UNVOICED-STOP-CANDIDATE ?s1-from ?s1-to)
  (CF (unvoiced-stop sokuon-closure-duration =(- ?s1-to ?s1-from) ?CFduration&: mightbe-valid)
    (exists (UNVOICED-STOP-CANDIDATE ?s2&:(?s2 < ?s1-from) ?))
    (exists (UNVOICED-STOP-CANDIDATE ?s3&:(?s3 > ?s1-to))))
  (power-strength ?s1-from ?s1-to 0 500 ?0-500-P)
  (CF (unvoiced-stop closure-0-500-power ?0-500-P) ?CF-0-500-P&: mightbe-valid)
  (power-strength ?s1-from ?s1-to 500 1000 ?500-1000-P)
  (CF (unvoiced-stop closure-0-500-500-1000-power-ratio =(- ?0-500-P ?500-1000-P)
    ?CF-0-500-500-1000-PR&: mightbe-valid)
    ?CF-0-500-500-1000-PR)
  (power-strength ?s1-from ?s1-to 0 6000 ?0-6000-P)
  (CF (unvoiced-stop closure-0-6000-power ?0-6000-P) ?CF-0-6000-P&: mightbe-valid)
  =>
  (bind ?segment (gensym "SEG-"))
  (assert (schema ?segment

```

```

(instance-of phoneme-segment)))
(hypothesize
(assert (category ?segment unvoiced-stop))
(assert (prop ?segment (type sokuon))
(prop ?segment (candidate-loc ?sl-from ?sl-to))
(segment-status ?segment segmentation)
;;
;; [hatazaki 143] CF of (category ?segment unvoiced-stop),
;; -->
;; =(CFcomb (CFweight ?*weak-evidence* ?CFduration)
;;          (CFweight ?*evidence*
;;                    (CFand ?CF-0-500-P
;;                          ?CF-0-500-500-1000-PR
;;                          ?CF-0-6000-P))))
;;
;; [hatazaki 143] CF of (category ?segment unvoiced-stop)
;; =(CFweight ?*evidence*
;;          (CFand
;;            ?CFduration
;;            ?CF-0-500-P
;;            ?CF-0-500-500-1000-PR
;;            ?CF-0-6000-P))
;; -->
;; =(CFcomb (CFweight ?*weak-evidence* ?CFduration)
;;          (CFweight ?*weak-evidence* ?CF-0-500-P)
;;          (CFweight ?*weak-evidence* ?CF-0-500-500-1000-PR)
;;          (CFweight ?*weak-evidence* ?CF-0-6000-P))
;;
(CF (category ?segment unvoiced-stop)
=(CFand (CFcomb (CFunknown)
                (CFweight ?*weak-evidence* ?CFduration)
                (CFweight ?*weak-evidence* ?CF-0-500-P)
                (CFweight ?*weak-evidence* ?CF-0-500-500-1000-PR)
                (CFweight ?*weak-evidence* ?CF-0-6000-P))
        ?CF-0-500-P
        ?CF-0-500-500-1000-PR
        ?CF-0-6000-P))))

```

```

;;;-----

```

```

(defrule sg-uvstop-bottom-up-1
  ""
  ;;
  ;;[hatazaki 126] Get REGIONS as intersected regions of
  ;; (caddr (#LUSER::call-speechfunc (list "POWER-ISLAND-IN-TIME"
  ;;                                     0 2000
  ;;                                     0 6000
  ;;                                     -999 -50
  ;;                                     11)))
  ;; and
  ;; (caddr (#LUSER::call-speechfunc (list "POWER-ISLAND-IN-TIME"
  ;;                                     0 2000
  ;;                                     0 500
  ;;                                     -999 -50
  ;;                                     11)))
  ;;
  ;;[hatazaki 118] Changed the threshold for detecting candidates to -50dB from -60dB.
  ;;
  (declare (salience ?*top-down*))
  (goal (UNVOICED-STOP-CANDIDATE ?from&?? ?to&??))
  (not (explicit (UNVOICED-STOP-CANDIDATE ? ?)))
  (BB ?BB)
  =>
  (at ?BB
    (for REGION in (intersected-regions
      (list
        (caddr (#LUSER::call-speechfunc (list "POWER-ISLAND-IN-TIME"
          0 2000
          0 6000
          -999 -50
          11)))
        (caddr (#LUSER::call-speechfunc (list "POWER-ISLAND-IN-TIME"
          0 2000
          0 500
          -999 -50
          11))))))
    do (assert (UNVOICED-STOP-CANDIDATE =(car REGION) =(cadr REGION))))))
  ;;
  ;;-----

```



```

(defrule sg-uvstop-3
  "burst exists or not?"
  (declare (salience ?*start-segmentation*))
  (join
    (segment-status ?segment segmentation)
    (category ?segment unvoiced-stop))
  (not (left-context ?segment word-initial))
  (not (applied ?segment sg-uvstop-3))
  (prop ?segment (candidate-loc ?cand-from ?cand-to))
  (case
    ;;
    ;; [hatazaki 122] Search a burst from ?cand-from, since a burst may be
    ;; included in the closure region. (cf. MAU_B_0065).
    ;; Old code ...
    ;; (burst-exists ?bu-from ?bu-to ?bu-freq after =(- ?cand-to 20) 60
    ;;           ?burst-CF& (?burst-CF >= ?*mightbe-valid*))
    ;; [hatazaki 120] Changed the range for searching a burst, from 50msec to 60msec.
    ;; [hatazaki 114]
    ((burst-exists ?bu-from ?bu-to ?bu-freq
      after ?cand-from =(+ 40 (- ?cand-to ?cand-from ))
      ?*burst-freq-min* ?*burst-freq-max*
      ?burst-CF&:mightbe-valid)
     (not (burst-exists ? ? ?
      after ?cand-from =(+ 40 (- ?cand-to ?cand-from ))
      ?*burst-freq-min* ?*burst-freq-max*
      ?another-CF&:(CFgreaterp ?another-CF ?burst-CF)))
    =>
    (assert (has-burst ?segment yes)
      (CF (has-burst ?segment yes)
        =(CFweight ?*strong-evidence* ?burst-CF)))
    (assert (prop ?segment (burst ?bu-from ?bu-to ?bu-freq)))
    (assert (applied ?segment sg-uvstop-3)))
  (otherwise
    =>
    (assert (has-burst ?segment no)
      (CF (has-burst ?segment no)
        =(CFweight ?*strong-evidence* 0.8)))
    (assert (applied ?segment sg-uvstop-3))))

(defrule sg-uvstop-4
  "Has low frequency power at the aspiration?"
  ;;
  ;;[hatazaki 172] Merge has-burst-power-in-low-frequency into
  ;; has-aspiration-power-in-low-frequency, Removed rule sg-uvstop-6.
  ;; Removed (has-burst ?segment no) in this rule.
  ;;[hatazaki 120]
  ;;
  (declare (salience ?*start-segmentation*))
  (join
    (segment-status ?segment segmentation)
    (category ?segment unvoiced-stop))
  ;;(has-burst ?segment no)
  (not (applied ?segment sg-uvstop-4))
  =>
  (assert (applied ?segment sg-uvstop-4))
  (conflicting-alternatives
    (hypothesize
      (assert (prop ?segment (has-aspiration-power-in-low-frequency yes))))
    (hypothesize
      (assert (prop ?segment (has-aspiration-power-in-low-frequency no))))))

(defrule sg-uvstop-8
  "Has burst power in low frequency?"
  (declare (salience ?*start-segmentation*))
  (join
    (segment-status ?segment segmentation)
    (category ?segment unvoiced-stop))
  ;;[hatazaki 177] Commented out.
  ;; This rule is applied even if the stop does not have a burst.
  ;;(has-burst ?segment yes)
  (not (applied ?segment sg-uvstop-8))
  =>
  (conflicting-alternatives
    (hypothesize

```

```

    (assert (prop ?segment (has-burst-power-in-low-frequency yes))))
  (hypothesize
    (assert (prop ?segment (has-burst-power-in-low-frequency no))))
  (assert (applied ?segment sg-uvstop-8)))

;;;(defrule sg-uvstop-6
;;;  "Has strong burst power in low frequency?"
;;;  (declare (salience ?*start-segmentation*))
;;;  (join
;;;    (segment-status ?segment segmentation)
;;;    (category ?segment unvoiced-stop))
;;;  (has-burst ?segment yes)
;;;  (not (applied ?segment sg-uvstop-6))
;;;  =>
;;;  (assert (applied ?segment sg-uvstop-6))
;;;  (conflicting-alternatives
;;;    (hypothesize
;;;      (assert (prop ?segment (has-burst-power-in-low-frequency yes))))
;;;    (hypothesize
;;;      (assert (prop ?segment (has-burst-power-in-low-frequency no))))))

(defrule sg-uvstop-5a
  "Has double burst?"
  ;;
  ;; [hatazaki 122] Hypothesizes the stop has a double burst or not.
  ;;
  (declare (salience ?*start-segmentation*))
  (join
    (segment-status ?segment segmentation)
    (category ?segment unvoiced-stop))
  (join
    (has-burst ?segment yes)
    (prop ?segment (burst ?burst-start ?burst-end ?burst-freq)))
  (not (applied ?segment sg-uvstop-5a))
  (burst-exists ?2nd-burst-from ?2nd-burst-to ?2nd-burst-freq
    after =(+ ?burst-start 10) 50 =(- ?burst-freq 250) =(+ ?burst-freq 250)
    ?CF2nd-burst&: mightbe-valid)
  (CF (unvoiced-stop double-burst-distance =(- ?2nd-burst-from ?burst-start)
    ?CFdistance&: mightbe-valid)
  =>
  (hypothesize
    (assert (prop ?segment (has-double-burst yes)))
    (assert (CF (prop ?segment (has-double-burst yes))
      =(CFweight ?*strong-evidence*
        (CFand ?CF2nd-burst ?CFdistance)))
      (prop ?segment
        (2nd-burst ?2nd-burst-from ?2nd-burst-to ?2nd-burst-freq))
      (applied ?segment sg-uvstop-5a))))

(defrule sg-uvstop-5b
  "Does not have double burst?"
  ;;
  ;; [hatazaki 122] Hypothesizes the stop has a double burst or not.
  ;;
  (declare (salience ?*start-segmentation*))
  (join
    (segment-status ?segment segmentation)
    (category ?segment unvoiced-stop))
  (join
    (has-burst ?segment yes)
    (prop ?segment (burst ?burst-start ?burst-end ?burst-freq)))
  (not (applied ?segment sg-uvstop-5b))
  (burst-exists ?2nd-burst-from ?2nd-burst-to ?2nd-burst-freq
    after =(+ ?burst-start 10) 50 =(- ?burst-freq 250) =(+ ?burst-freq 250)
    ?CF2nd-burst&: isnot-valid)
  (CF (unvoiced-stop double-burst-distance =(- ?2nd-burst-from ?burst-start)
    ?CFdistance)
  =>
  (hypothesize
    (assert (prop ?segment (has-double-burst no)))
    ;;
    ;; [hatazaki 172] When ?CFdistance is negative, CF of (prop ?segment (has-double-burst no))
    ;; must have a large positive CF value, even when ?CF2nd-burst is positive.
    ;; (cf. MAU_1_0282)
    ;;
    =(CFweight ?*strong-evidence*
      (CFor (- ?CF2nd-burst) (- ?CFdistance))))

```

```

;; <-- =(CFweight ?*strong-evidence*
;;      (- ?CF2nd-burst))
;;[hatazaki 142] =(min 0.9 (- 0.8 ?CF2nd-burst)) -->
;;      =(CFweight ?*strong-evidence* (- ?CF2nd-burst))
(assert (CF (prop ?segment (has-double-burst no))
            =(CFweight ?*strong-evidence*
              (CFor (- ?CF2nd-burst) (- ?CFdistance))))
        (applied ?segment sg-uvstop-5b))))

(defrule sg-uvstop-5c
  "Has double burst?"
  ;;
  ;; [hatazaki 122] Hypothesizes the stop has a double burst or not.
  ;;
  (declare (salience ?*start-segmentation*))
  (join
    (segment-status ?segment segmentation)
    (category ?segment unvoiced-stop))
  (join
    (has-burst ?segment yes)
    (prop ?segment (burst ?burst-start ?burst-end ?burst-freq)))
  (not (applied ?segment sg-uvstop-5c))
  (not (burst-exists ? ? ?
                    after =(+ ?burst-start 10) 50 =(- ?burst-freq 250) =(+ ?burst-freq 250)
                    ?))
  =>
  (hypothesize
    (assert (prop ?segment (has-double-burst no)))
    (assert (CF (prop ?segment (has-double-burst no))
                =(CFweight ?*strong-evidence* 0.9))
            (applied ?segment sg-uvstop-5c))))

(defcontradiction sg-uvstop-5d
  ""
  (prop ?segment (has-double-burst no))
  (prop ?segment (has-double-burst yes)))

(defrule sg-uvstop-7
  "Has two power-decrease at the preceding vowel end?"
  ;;
  ;;[hatazaki 155] Newly defined to hypothesize (has-extra-voicing-in-closure yes) and
  ;; (has-extra-voicing-in-closure no).
  ;; (prop ?segment (has-extra-voicing-in-closure yes)) is for the case
  ;; when there are two power decrease at the preceding vowel end, one is the end of
  ;; vowel, another is the end of extra voicing.
  ;;
  (declare (salience ?*start-segmentation*))
  (join
    (segment-status ?segment segmentation)
    (category ?segment unvoiced-stop))
  (not (applied ?segment sg-uvstop-7))
  =>
  (assert (applied ?segment sg-uvstop-7))
  (conflicting-alternatives
    (hypothesize
      (assert (prop ?segment (has-extra-voicing-in-closure yes))))
    (hypothesize
      (assert (prop ?segment (has-extra-voicing-in-closure no))))))

;;-----

```

```

(defrule sg-uvstop-find-closure-1
  ""
  (declare (salience ?*start-segmentation*))
  (join
    (segment-status ?segment segmentation)
    (category ?segment unvoiced-stop)
    ?x <- (CF (category ?segment unvoiced-stop) ?CFcategory))
  (not (applied ?segment sg-uvstop-find-closure-1))
  (not (left-context ?segment word-initial))
  (has-burst ?segment no)
  (prop ?segment (candidate-loc ?from ?to))
  (closure-exists ?cl-from ?cl-to before ?to =(- ?to ?from) ?CFclosure&: mightbe-valid)
  =>
  (assert (applied ?segment sg-uvstop-find-closure-1))
  (assert (prop ?segment (closure ?cl-from ?cl-to ?CFclosure)))
  (retract ?x)
  ;;
  ;;[hatazaki 159] =(CFand ?CFcategory (CFweight ?*evidence* ?CFclosure))
  ;; <-- =(CFcomb ?CFcategory (CFweight ?*evidence* ?CFclosure)).
  ;;[hatazaki 142] Changed a new CF of (category ?segment unvoiced-stop):
  ;; =(CFand ?CFcategory ?CFclosure) -->
  ;; =(CFcomb ?CFcategory (CFweight ?*evidence* ?CFclosure))
  (assert (CF (category ?segment unvoiced-stop)
    =(CFweight ?*evidence* ?CFclosure))))

(defrule sg-uvstop-find-closure-2
  ""
  (declare (salience ?*start-segmentation*))
  (join
    (segment-status ?segment segmentation)
    (category ?segment unvoiced-stop))
  ?x <- (CF (category ?segment unvoiced-stop) ?CFcategory)
  (not (applied ?segment sg-uvstop-find-closure-2))
  (not (left-context ?segment word-initial))
  (join
    (has-burst ?segment yes)
    (CF (has-burst ?segment yes) ?CFburst&: mightbe-valid)
    (prop ?segment (burst ?here ? ?)))
  (closure-exists ?cl-from ?here before ?here 30 ?CFclosure&: mightbe-valid)
  =>
  (assert (applied ?segment sg-uvstop-find-closure-2))
  (assert (prop ?segment (closure ?cl-from ?here ?CFclosure)))
  (retract ?x)
  ;;
  ;;[hatazaki 159] =(CFand ?CFcategory (CFweight ?*evidence* ?CFclosure))
  ;; <-- =(CFcomb ?CFcategory (CFweight ?*evidence* ?CFclosure)).
  ;;[hatazaki 142] Changed a new CF of (category ?segment unvoiced-stop):
  ;; =(CFand ?CFcategory ?CFclosure) -->
  ;; =(CFcomb ?CFcategory (CFweight ?*evidence* ?CFclosure))
  (assert (CF (category ?segment unvoiced-stop)
    =(CFweight ?*evidence* ?CFclosure))))

(defrule sg-uvstop-find-closure-3
  ""
  ;;
  ;; [hatazaki 130] Added this rule for unvoiced stop at the word beginning.
  ;;
  (declare (salience ?*start-segmentation*))
  (join
    (segment-status ?segment segmentation)
    (category ?segment unvoiced-stop))
  ?x <- (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid)
  (not (applied ?segment sg-uvstop-find-closure-3))
  (left-context ?segment word-initial)
  (join
    (has-burst ?segment yes)
    (CF (has-burst ?segment yes) ?CFburst&: mightbe-valid)
    (prop ?segment (burst ?here ? ?)))
  =>
  (assert (applied ?segment sg-uvstop-find-closure-3))
  (assert (prop ?segment (closure UNKNOWN ?here ?CFburst))))

(defrule get-closure-1
  "(BACKWARD CHAINING RULE)"

```

```

(declare (salience ?*top-down*))
(goal (closure-exists ?from&?? ?to&?? before ?here&~?? ?range&~?? ?CF&??))
;;
;;[hatazaki 128] Searches ?cl-end to the right of ?here.
;;[hatazaki 127] Uses ?*normal-smoothing-size* and
;; ?*closure-end-min-change* instead of *smoothing-for-finding-burst
;; and ?*burst-start-min-change*, to searches ?cl-end, since this rule
;; is
;; for the case when there is no burst.
;;
;;[hatazaki 127] Searches ?cl-start ?cl-end with 1000-4000 Power
;; change instead of 0-6000 Powre change, and uses
;; ?*burst-start-min-change* instead of ?*closure-end-min-change* for
;; searching ?cl-end.
;;
;;[hatazaki 125] It is better to search closure boundaries by using power-change,
;; since the burst power is not enough large. (cf. MAU_B_0010 /k/)
;; OLD code ...
;; (power-end ?cl-start&~NONE
;;   before ?here =(+ ?range 500) 0 6000 ?*closure-power-upper*)
;; (power-start ?cl-end&~NONE
;;   after ?cl-start =(- ?here ?cl-start -10) 0 6000 ?*closure-power-upper*)
;;
;; - Closure-end cannot be founded by power-increase with ?*normal-smoothing-size*,
;; since its power change is so short. Then, the closure boundaries shoud be searched by
;; power-start and power-end.
;; OLD code ...
;; (power-decrease (?cl-start&~NONE ?cl-start-change)
;;   before ?here =(+ ?range 500) 0 6000
;;   ?*normal-smoothing-size* ?*closure-start-min-change*)
;; (power-increase (?cl-end&~NONE ?cl-end-change)
;;   after ?cl-start =(- ?here ?cl-start) 0 6000
;;   ?*normal-smoothing-size* ?*closure-end-min-change*)
;;
;; Old code ...
;; (power-start ?cl-end&~NONE
;;   before ?here ?range 0 6000 ?*closure-power-upper*)
;; (power-end ?cl-start&~NONE
;;   before ?cl-end 400 0 6000 ?*closure-power-upper*)
;;
(power-decrease (?cl-start&~NONE ?cl-start-change)
  before ?here =(+ ?range 500) 1000 4000
  ?*normal-smoothing-size* ?*closure-start-min-change*)
(power-increase (?cl-end&~NONE ?cl-end-change)
  after ?cl-start =(- ?here ?cl-start -50) 1000 4000
  ?*normal-smoothing-size* ?*closure-end-min-change*)
(test (<= (- ?here ?cl-end) ?range))
(CF (unvoiced-stop closure-duration =(- ?cl-end ?cl-start)) ?CF1)
(CF (unvoiced-stop sokuon-closure-duration =(- ?cl-end ?cl-start)) ?CF2)
(test (mightbe-valid (CFor ?CF1 ?CF2)))
(power-strength ?cl-start ?cl-end 0 6000 ?0-6000-power)
(CF (unvoiced-stop closure-0-6000-power ?0-6000-power) ?CFpower&:mightbe-valid)
(BB ?BB)
=>
(at ?BB
  (assert (closure-exists ?cl-start ?cl-end before ?here ?range
    =(CFand (CFor ?CF1 ?CF2) ?CFpower))))

(defrule get-closure-2
  "When ?to is provided."
  ;;
  ;; [hatazaki 127]
  ;;
  (declare (salience ?*top-down*))
  (goal (closure-exists ?from&?? ?to&~?? before ?here&~?? ?range&~?? ?CF&??))
  (power-decrease (?cl-start&~NONE ?cl-start-change)
    before ?here =(+ ?range 500) 1000 4000
    ?*normal-smoothing-size* ?*closure-start-min-change*)
  (CF (unvoiced-stop closure-duration =(- ?to ?cl-start)) ?CF1)
  (CF (unvoiced-stop sokuon-closure-duration =(- ?to ?cl-start)) ?CF2)
  (test (mightbe-valid (CFor ?CF1 ?CF2)))
  (power-strength ?cl-start ?to 0 6000 ?0-6000-power)
  (CF (unvoiced-stop closure-0-6000-power ?0-6000-power) ?CFpower&:mightbe-valid)
  (BB ?BB)
  =>
  (at ?BB

```

```

(assert (closure-exists ?cl-start ?to before ?here ?range
=(CFand (CFor ?CF1 ?CF2) ?CFpower))))

(defrule get-closure-after-1
  ""
  ;;[hatazaki 148] Searches ?cl-end first, then ?cl-start backward from ?cl-end,
  ;; since there may be power decrease before ?cl-start.
  ;; <--
  ;; (power-decrease (?cl-start&~NONE ?cl-start-change)
  ;;   after ?here ?range 1000 4000
  ;;   ?*normal-smoothing-size* ?*closure-start-min-change*)
  ;; (power-increase (?cl-end&~NONE ?cl-end-change)
  ;;   after ?cl-start 500 1000 4000
  ;;   ?*normal-smoothing-size* ?*closure-end-min-change*)
  ;;
  ;;[hatazaki 143] This is for unvoiced-fricative followed by a closure,
  ;; and used by rule sg-uvfric-bfr-stop-or-affr-1.
  ;;
  (declare (salience ?*top-down*))
  (goal (closure-exists ?from&?? ?to&?? after ?here&~?? ?range&~?? ?CF&??))
  (power-increase (?cl-end&~NONE ?cl-end-change)
    after ?here =(+ ?range 500) 1000 4000
    ?*normal-smoothing-size* ?*closure-end-min-change*)
  (power-decrease (?cl-start&~NONE ?cl-start-change)
    before ?cl-end =(- ?cl-end ?here -50) 1000 4000
    ?*normal-smoothing-size* ?*closure-start-min-change*)
  (test (<= (- ?cl-start ?here) ?range))
  (CF (unvoiced-stop closure-duration =(- ?cl-end ?cl-start)) ?CF1)
  (CF (unvoiced-stop sokuon-closure-duration =(- ?cl-end ?cl-start)) ?CF2)
  (test (mightbe-valid (CFor ?CF1 ?CF2)))
  (power-strength ?cl-start ?cl-end 0 6000 ?0-6000-power)
  (CF (unvoiced-stop closure-0-6000-power ?0-6000-power) ?CFpower&:mightbe-valid)
  (BB ?BB)
  =>
  (at ?BB
    (assert (closure-exists ?cl-start ?cl-end after ?here ?range
      =(CFand (CFor ?CF1 ?CF2) ?CFpower))))

  ;;-----

```

```

(defrelation search-burst-after (?here ?range ?freq-min ?freq-max))

(defrule get-burst-right-of-unvoiced-stop-0
  "Gets a burst following the unvoiced stop.
  (BACKWARD CHAINING RULE)"
  ;; [hatazaki 114] Assert more than one fact.
  ;; [hatazaki 107]
  ;;
  (declare (salience ?*top-down*))
  ;;(declare (salience ?*right-segmentation*))
  (goal (burst-exists ?from&?? ?to&?? ?freq&??
    after ?here&~?? ?range&~?? ?freq-min&~?? ?freq-max&~??
    ?CF&??))
  (explicit (not (burst-exists ? ? ? after ?here ?range ?freq-min ?freq-max ?)))
  (BB ?BB)
  =>
  (at ?BB
    (assert (search-burst-after ?here ?range ?freq-min ?freq-max))))

(defrule get-burst-right-of-unvoiced-stop-1
  ""
  ;;
  ;; [hatazaki 143] Searches a burst between ?freq-min and ?freq-max, which are specified in
  ;; (burst-exists ?from&?? ?to&?? ?freq&??
  ;; after ?here&~?? ?range&~?? ?freq-min&~?? ?freq-max&~??
  ;; ?CF&??)
  ;;
  ;; [hatazaki 131] Uses ?*burst-1000-4000-start-min-change* instead of
  ;; ?*burst-start-min-change* for finding ?burst-1000-4000-start.
  ;;
  ;; [hatazaki 127] Uses ?*smoothing-for-finding-burst-end* (is no
  ;; smoothing) for finding burst end.
  ;;
  ;; [hatazaki 119] Checks the burst duration.
  ;;
  ;; [hatazaki 116] Modified get-burst-right-of-unvoiced-stop-1 to check
  ;; power change at the spectral peaks for both burst start and end
  ;; time. Power change at the burst start time should be checked if this
  ;; is
  ;; enough large, to say it is a burst. Power change at the burst end
  ;; may not be so large (cf. MAU_B_0169), then it is OK even if the power
  ;; change at the burst end is nearly 0.
  ;;
  (declare (salience ?*top-down*))
  (search-burst-after ?here ?range ?freq-min ?freq-max)
  (power-increase (?burst-1000-4000-start&~NONE ?burst-1000-4000-start-change)
    after ?here ?range ?freq-min ?freq-max
    ?*smoothing-for-finding-burst*
    ?*burst-1000-4000-start-min-change*)
  (CF (unvoiced-stop burst-1000-4000-start-change ?burst-1000-4000-start-change)
    ?CF-1000-4000-start&:mightbe-valid)
  (spectrum-peak (?peak-freq&~NONE&:(?peak-freq >= ?freq-min)&:(?peak-freq <= ?freq-max)
    ?peak-amp ?peak-Q)
    at ?burst-1000-4000-start=(+ ?burst-1000-4000-start 10)
    ?*spectrum-peak-smoothing-for-getting-burst*)
  ;;
  ;; [hatazaki 173] Searches ?burst-start around ?burst-1000-4000-start.
  ;; after ?here =(- ?burst-1000-4000-start ?here -10)
  ;; <-- after ?here ?range
  ;;
  (power-increase (?burst-start&~NONE ?burst-start-change)
    after ?here =(- ?burst-1000-4000-start ?here -10)
    =(- ?peak-freq 100) =(+ ?peak-freq 100)
    ?*smoothing-for-finding-burst*
    ?*burst-start-min-change*)
  (CF (unvoiced-stop burst-start-change ?burst-start-change) ?CF-start&:mightbe-valid)
  ;;
  ;; [hatazaki 150] ?burst-end may not be found.
  ;; <--
  ;; (power-decrease (?burst-end&~NONE ?burst-end-change)
  ;; after ?burst-start 50
  ;; =(- ?peak-freq 100) =(+ ?peak-freq 100)
  ;; ?*smoothing-for-finding-burst-end*
  ;; ?*burst-end-min-change*)

```

```

;; (CF (unvoiced-stop burst-end-change ?burst-end-change)
;;      ?CF-end&:(?CF-end >= ?*mightbe-valid*))
;; (CF (unvoiced-stop burst-duration =(- ?burst-end ?burst-start))
;;      ?CFduration&:(?CFduration >= ?*mightbe-valid*))
;;
(case
  ((power-decrease (?burst-end&-NONE ?burst-end-change)
    after ?burst-start 30
    =(- ?peak-freq 100) =(+ ?peak-freq 100)
    ?*smoothing-for-finding-burst-end*
    ?*burst-end-min-change*)
    (CF (unvoiced-stop burst-end-change ?burst-end-change)
      ?CF-end&:mightbe-valid)
    (CF (unvoiced-stop burst-duration =(- ?burst-end ?burst-start))
      ?CFduration&:mightbe-valid))
  (otherwise
    =>
    (bind ?burst-end 'NONE)
    (bind ?CF-end ?*CFunknown*)
    (bind ?CFduration ?*CFunknown*)))
; (power-strength ? ?burst-start ?burst-end =(- ?peak-freq 100) =(+ ?peak-freq 100)
;      ?burst-power)
(BB ?BB)
=>
;;
;;[hatazaki 150] =(CFcomb (CFweight ?*strong-evidence* ?CF-start)
;;                      (CFweight ?*weak-evidence* ?CF-end )
;;                      (CFweight ?*weak-evidence* ?CFduration))
;;
;;      <--
;;      =(CFand ?CF-start ?CF-end ?CFduration).
;;[hatazaki 135] Changed (CFand (CFcomb ?CF-start ?CF-end) ?CFduration) to
;;                      (CFand ?CF-start ?CF-end ?CFduration).
;;
;;
(at ?BB
  (assert
    (burst-exists ?burst-start ?burst-end ?peak-freq
      after ?here ?range ?freq-min ?freq-max
      =(CFvalue (CFand (CFcomb (CFunknown)
        (CFweight ?*strong-evidence* ?CF-start)
        (CFweight ?*weak-evidence* ?CF-end )
        (CFweight ?*weak-evidence* ?CFduration))
        ?CF-start))))))

```



LM17: >hatazaki>sre>seg-unvoiced-stop>seg-stop-left. art. 33

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 12/01/88 20:20:31

Queued at: 3/28/89 10:58:37

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-uvstop-left-1
  "Hypothesizes left phoneme."
  ;;
  ;; [hatazaki 130] Changed (prop ?segment (location word-initial)) to
  ;; (left-context ?segment word-initial), and
  ;; Added (CF (category ?segment unvoiced-stop)
  ;; ?CFcategory& (?CFcategory >= ?*mightbe-valid*)).
  ;; [hatazaki 127] Added (prop ?segment (closure ?cl-from ?cl-to ?CFclosure)).
  ;;
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
    (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
    (not (left-context ?segment word-initial))
    (not (start-time ?segment ?))
    (not (applied ?segment sg-uvstop-left-1))
  =>
  (assert (applied ?segment sg-uvstop-left-1))
  (conflicting-alternatives
    (hypothesize
      (assert (left-context ?segment vowel))
      (assert (CF (left-context ?segment vowel) =(CFunknown))))
    (hypothesize
      (assert (left-context ?segment devoiced-vowel))
      (assert (CF (left-context ?segment devoiced-vowel) =(CFunknown))))
    (hypothesize
      (assert (left-context ?segment nasal))
      (assert (CF (left-context ?segment nasal) =(CFunknown))))
  ))
)
```

```
-----
```

```

(defrule sg-uvstop-left-word-initial-done-1
  "For unvoiced stop at word initial."
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory))
  (join
    (left-context ?segment word-initial)
    (CF (left-context ?segment word-initial) ?CFcontext))
  ;(prop ?segment (location word-initial))
  (not (applied ?segment sg-uvstop-left-word-initial-done-1))
  (prop ?segment (burst ?burst-start ?burst-end ?bu-freq))
  =>
  (assert (applied ?segment sg-uvstop-left-word-initial-done-1))
  (hypothesize
    (assert (start-time ?segment ?burst-start))
    ;;
    ;;[hatazaki 169] =(CFvalue ?CFcontext)
    ;; <-- =(CFcomb (CFunknown)
    ;;           (CFweight ?*strong-evidence* ?CFcontext))
    ;;[hatazaki 142] Changed new CF of (start-time ?segment ?burst-start),
    ;; ?CF-category --> =(CFcomb (CFweight ?*strong-evidence* ?CF-category)).
    ;;
    (assert (CF (start-time ?segment ?burst-start)
      =(CFvalue ?CFcontext))))))

;;-----
(defrule sg-uvstop-aft-vowel-0
  "There should not be a closure before this segment."
  ;; - This rule should be applied before the other segmentation rules below are applied.
  (declare (salience (+ ?*left-segmentation* 1)))
  (segment-status ?segment segmentation)
  (category ?segment unvoiced-stop)
  (join
    (left-context ?segment vowel)
    ?old-CF <- (CF (left-context ?segment vowel) ?CFcontext))
    (not (applied ?segment seg-uvstop-aft-vowel-0))
    (prop ?segment (candidate-loc ?cand-from ?))
  ;;
  ;;[hatazaki 128] Checks (?to < ?cand-from) since get-closure finds
  ;;closure which end time it right of ?cand-from.
  (closure-exists ?from ?to&:(?to < ?cand-from) before ?cand-from 20 ?closure-CF)
  =>
  (retract ?old-CF)
  (assert (CF (left-context ?segment vowel)
    =(CFcomb (CFand ?CFcontext
      (- (CFvalue ?closure-CF)))
      (CFweight ?*evidence* (- (CFvalue ?closure-CF))))))
  (assert (applied ?segment seg-uvstop-aft-vowel-0)))

(defrule sg-uvstop-aft-vowel-1a
  "find time the preceding vowel ends."
  ;; [hatazaki 107]
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment vowel)
    ?x <- (CF (left-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvstop-aft-vowel-1a))
  (prop ?segment (closure ?cl-from&~UNKNOWN ?cl-to ?CFclosure))
  (prop ?segment (has-extra-voicing-in-closure no))
  ;;
  ;; (power-end ?vowel-region&~NONE
  ;;           before =(+ ?cl-from 20) 100 1000 4000
  ;;           ?*vowel-1000-4000-power-before-stop-lower*)
  ;; (power-decrease (?vowel-end&~NONE ?change)
  ;;                 after =(- ?vowel-region 30) 60 1000 4000
  ;;                 ?*normal-smoothing-size* ?*vowel-F2-end-min-change*)
  ;;[hatazaki 152] Searches vowel-end by power-end first, and then power-decrease.

```

```

;; <--
;; (power-decrease (?vowel-end&~NONE ?change)
;;   before ?cl-to =(- ?cl-to ?cl-from -50) 1000 4000
;;   ?*normal-smoothing-size* ?*vowel-F2-end-min-change*)
;;[hatazaki 125] Searches ?vowel-end from ?cl-to, since ?vowel-end
;; may be right of ?cl-from.
;; OLD code ...
;;   ... before =(+ ?cl-from 20) 70 ...
;;
(power-decrease (?vowel-end&~NONE ?change)
  before ?cl-to =(- ?cl-to ?cl-from -50) 1000 4000
  ?*normal-smoothing-size* ?*vowel-F2-end-min-change*)
(power-strength =(- ?vowel-end 30) =(- ?vowel-end 10) 0 500 ?0-500-power)
(CF (vowel 0-500-power ?0-500-power) ?CF-vowel-0-500)
(power-strength =(- ?vowel-end 30) =(- ?vowel-end 10) 1000 6000 ?1000-6000-power)
(CF (vowel 1000-6000-power ?1000-6000-power) ?CF-vowel-1000-6000)
=>
(assert (applied ?segment sg-uvstop-aft-vowel-1a))
(assert (prop ?segment (vowel-end-time ?segment ?vowel-end)))
;;
;; [hatazaki 142] New CF of (left-context ?segment vowel),
;;   =(CFcomb ?CF (CFweight ?*evidence*
;;     (CFand ?CF-vowel-0-500 ?CF-vowel-1000-6000)))
;; -->
;;   =(CFcomb ?CF
;;     (CFweight ?*evidence* ?CF-vowel-0-500)
;;     (CFweight ?*evidence* ?CF-vowel-1000-6000))
;;
(retract ?x)
(assert (CF (left-context ?segment vowel)
  =(CFand (CFcomb ?CFcontext
    (CFweight ?*evidence* ?CF-vowel-0-500)
    (CFweight ?*evidence* ?CF-vowel-1000-6000))
  ?CF-vowel-0-500 ?CF-vowel-1000-6000))))

(defrule sg-uvstop-aft-vowel-1b
  "find time the preceding vowel ends, for the case when there are two power-decrease."
  ;; [hatazaki 107]
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
  (join
    (left-context ?segment vowel)
    ?x <- (CF (left-context ?segment vowel) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvstop-aft-vowel-1b))
  (prop ?segment (closure ?cl-from&~UNKNOWN ?cl-to ?CFclosure))
  (prop ?segment (has-extra-voicing-in-closure yes))
  (power-decrease (?voicing-end&~NONE ?voicing-end-change)
    before ?cl-to =(- ?cl-to ?cl-from -50) 1000 4000
    ?*normal-smoothing-size* ?*vowel-F2-end-min-change*)
  (power-decrease (?vowel-end&~NONE ?change)
    before ?voicing-end 100 1000 4000
    ?*normal-smoothing-size* ?*vowel-F2-end-min-change*)
  (power-increase (?y&:(eq ?y 'NONE) ?z)
    after ?vowel-end =(- ?voicing-end ?vowel-end) 1000 4000
    ?*normal-smoothing-size* ?*vowel-F2-start-min-change*)
  (power-strength =(- ?vowel-end 30) =(- ?vowel-end 10) 0 500 ?0-500-power)
  (CF (vowel 0-500-power ?0-500-power) ?CF-vowel-0-500)
  (power-strength =(- ?vowel-end 30) =(- ?vowel-end 10) 1000 6000 ?1000-6000-power)
  (CF (vowel 1000-6000-power ?1000-6000-power) ?CF-vowel-1000-6000)
  =>
  (assert (applied ?segment sg-uvstop-aft-vowel-1b))
  (assert (prop ?segment (vowel-end-time ?segment ?vowel-end)))
  (retract ?x)
  (assert (CF (left-context ?segment vowel)
    =(CFand (CFcomb ?CFcontext
      (CFweight ?*evidence* ?CF-vowel-0-500)
      (CFweight ?*evidence* ?CF-vowel-1000-6000))
    ?CF-vowel-0-500 ?CF-vowel-1000-6000))))

(defrule sg-uvstop-aft-vowel-done-1
  ""
  ;; [hatazaki 107]
  (declare (salience ?*left-segmentation-end*))

```

```
(segment-status ?segment segmentation)
(category ?segment unvoiced-stop)
(left-context ?segment vowel)
(not (applied ?segment sg-uvstop-aft-vowel-done-1))
(CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid)
(CF (left-context ?segment vowel) ?CFcontext&:mightbe-valid)
(prop ?segment (vowel-end-time ?segment ?vowel-end))
=>
(assert (applied ?segment sg-uvstop-aft-vowel-done-1))
(hypothesize
  (assert (start-time ?segment ?vowel-end))
  (assert (CF (start-time ?segment ?vowel-end)
              =(CFvalue ?CFcontext))))))
```

-----

```

(defrule sg-uvstop-aft-devoiced-vowel-1
  ""
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment devoiced-vowel)
    (?x <- (CF (left-context ?segment devoiced-vowel) ?CFcontext&: mightbe-valid))
    (not (applied ?segment sg-uvstop-aft-devoiced-vowel-1))
    ;;(prop ?segment (candidate-loc ?from ?))
    (prop ?segment (closure ?from ?to ?CFclosure))
    ;;
    ;;[hatazaki 125] Searches ?closure-start from ?to, since ?vowel-end
    ;; may be right of ?from.
    ;; OLD code ...
    ;; ... before =(+ ?from 20) 100 ...
    ;;
    (power-decrease (?closure-start&~NONE ?change)
      before ?to =(- ?to ?from -80) 1000 6000
      ?*normal-smoothing-size* ?*vowel-F2-end-min-change*)
    (power-strength =(- ?closure-start 30) =(- ?closure-start 10) 0 500 ?0-500-power)
    (CF (vowel 0-500-power ?0-500-power) ?CF-vowel-0-500)
    =>
    (assert (applied ?segment sg-uvstop-aft-devoiced-vowel-1))
    (assert (prop ?segment (closure-start-time ?closure-start)))
    (retract ?x)
    (assert (CF (left-context ?segment devoiced-vowel)
      =(CFand (CFcomb ?CFcontext
        (CFweight ?*strong-evidence*
          (- (CFvalue ?CF-vowel-0-500))))
        (- (CFvalue ?CF-vowel-0-500)))))))

(defrule sg-uvstop-aft-devoiced-vowel-done-1
  ""
  (declare (saliency ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment devoiced-vowel)
    (CF (left-context ?segment devoiced-vowel) ?CFcontext&: mightbe-valid))
    (not (applied ?segment sg-uvstop-aft-devoiced-vowel-done-1))
    (prop ?segment (closure-start-time ?closure-start))
    =>
    (assert (applied ?segment sg-uvstop-aft-devoiced-vowel-done-1))
    (hypothesize
      (assert (start-time ?segment ?closure-start))
      (assert (CF (start-time ?segment ?closure-start)
        =(CFcomb (CFweight ?*no-evidence* ?CFcategory)
          (CFweight ?*definite-evidence* ?CFcontext))))))

;;-----

```

```

;;(defrule sg-uvstop-aft-syllabic-nasal-1
;;  "Preceded by a syllabic nasal."
;;  (declare (salience ?*left-segmentation*))
;;  (segment-status ?segment segmentation)
;;  (join
;;    (category ?segment unvoiced-stop)
;;    (CF (category ?segment unvoiced-stop) ?CFcategory&:(?CFcategory >= ?*mightbe-valid*)))
;;  (join
;;    (left-context ?segment nasal)
;;    ?x <- (CF (left-context ?segment nasal) ?CF&:(?CF >= 0)))
;;  (not (applied ?segment sg-uvstop-aft-syllabic-nasal-1))
;;  ;;(prop ?segment (candidate-loc ?from ?))
;;  (prop ?segment (closure ?from ? ?CFclosure))
;;  (power-decrease (?nasal-end&~NONE ?change)
;;    before =(+ ?from 20) 100 1000 4000
;;    ?*normal-smoothing-size* ?*vowel-F2-end-min-change*)
;;  (power-strength =(- ?nasal-end 30) =(- ?nasal-end 10) 0 500 ?0-500-P)
;;  (CF (syllabic-nasal 0-500-power ?0-500-P) ?CF-0-500-p)
;;  (power-strength =(- ?nasal-end 30) =(- ?nasal-end 10) 1000 6000 ?1000-6000-P)
;;  (CF (syllabic-nasal 1000-6000-power ?1000-6000-P) ?CF-1000-6000-P)
;;  =>
;;  (assert (applied ?segment sg-uvstop-aft-syllabic-nasal-1))
;;  (assert (prop ?segment (nasal-end ?nasal-end)))
;;  (retract ?x)
;;  ;;
;;  ;; [hatazaki 142] New CF of (left-context ?segment nasal),
;;  ;;   =(CFcomb ?CF (CFweight ?*evidence*
;;  ;;   (CFand ?CF-0-500-p ?CF-1000-6000-P)))
;;  ;; -->
;;  ;;   =(CFcomb ?CF
;;  ;;   (CFweight ?*evidence* ?CF-0-500-p)
;;  ;;   (CFweight ?*evidence* ?CF-1000-6000-P))
;;  ;;
;;  (assert (CF (left-context ?segment nasal)
;;    =(CFcomb ?CF
;;      (CFweight ?*evidence* ?CF-0-500-p)
;;      (CFweight ?*evidence* ?CF-1000-6000-P))))))
;;(defrule sg-uvstop-aft-syllabic-nasal-done-1
;;  ""
;;  (declare (salience ?*left-segmentation-end*))
;;  (segment-status ?segment segmentation)
;;  (join
;;    (category ?segment unvoiced-stop)
;;    (CF (category ?segment unvoiced-stop) ?CFcategory&:(?CFcategory >= ?*mightbe-valid*)))
;;  (join
;;    (left-context ?segment nasal)
;;    (CF (left-context ?segment nasal) ?CFcontext&:(?CFcontext >= ?*mightbe-valid*)))
;;  (not (applied ?segment sg-uvstop-aft-syllabic-nasal-done-1))
;;  (prop ?segment (nasal-end ?nasal-end))
;;  =>
;;  (assert (applied ?segment sg-uvstop-aft-syllabic-nasal-done-1))
;;  (hypothesize
;;    (assert (start-time ?segment ?nasal-end))
;;    (assert (CF (start-time ?segment ?nasal-end)
;;      =(CFcomb (CFweight ?*no-evidence* ?CFcategory)
;;        (CFweight ?*definite-evidence* ?CFcontext))))))

```





.LM17: >hatazaki>sre>seg-unvoiced-stop>seg-stop-right. art. 78

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/01/89 17:03:10

Queued at: 3/28/89 10:59:33

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-uvstop-right-0
  "Hypothesizes right phoneme."
  ;;
  ;; [hatazaki 130] Removed (prop ?segment (location word-initial)), and
  ;; Added (CF (category ?segment unvoiced-stop
  ;;           ?CFcategory& (?CFcategory >= ?*mightbe-valid*))).
  ;; [hatazaki 128] Changed (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
  ;; to (or (prop ?segment (location word-initial))
  ;;        (prop ?segment (closure ?cl-from ?cl-to ?CFclosure)))
  ;; [hatazaki 127] Added (prop ?segment (closure ?cl-from ?cl-to ?CFclosure)).
  ;;
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
   (category ?segment unvoiced-stop)
   (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
  (not (end-time ?segment ?))
  (not (applied ?segment sg-uvstop-right-0))
  =>
  (assert (applied ?segment sg-uvstop-right-0))
  (conflicting-alternatives
   (hypothesize
    (assert (right-context ?segment vowel))
    (assert (CF (right-context ?segment vowel) =(CFunknown))))))
  (hypothesize
   (assert (right-context ?segment stop))
   (assert (CF (right-context ?segment stop) =(CFunknown))))))
  (hypothesize
   (assert (right-context ?segment fricative))
   (assert (CF (right-context ?segment fricative) =(CFunknown))))))
```

```
;;;-----
```

```

;;;[hatazaki 177] sg-uvstop-bfr-vowel-2a,b are merged to
;;; sg-uvstop-bfr-vowel-3a, b, c.
;;;(defrule sg-uvstop-bfr-vowel-2a
;;;  "No burst and no aspiration-power-in-low-frequency."
;;;  ;; [hatazaki 107]
;;;  ;;
;;;  ;; (i.e.) MAU_B_0012
;;;  ;;
;;;  (declare (salience ?*left-segmentation*))
;;;  (segment-status ?segment segmentation)
;;;  (join
;;;    (category ?segment unvoiced-stop)
;;;    (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
;;;  (join
;;;    (right-context ?segment vowel)
;;;    ?x <- (CF (right-context ?segment vowel) ?CFcontext&:mightbe-valid))
;;;  (has-burst ?segment no)
;;;  (prop ?segment (has-aspiration-power-in-low-frequency no))
;;;  (not (applied ?segment sg-uvstop-bfr-vowel-2a))
;;;  (prop ?segment (candidate-loc ? ?cand-to))
;;;  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
;;;  ;;
;;;  ;;[hatazaki 172] Use power-increase instead of power-start, to find ?vowel-region,
;;;  ;; since voicing power may be small.
;;;  ;; <--
;;;  ;; (power-start ?vowel-region&~NONE
;;;  ;;   after =(- ?cl-to 20) 200 0 500 ?*voicing-power-after-stop-lower*)
;;;  (power-increase (?vowel-region&~NONE ?vowel-region-change)
;;;    after =(- ?cl-to 20) 200 0 500
;;;    ?*normal-smoothing-size*
;;;    ?*default-min-change*)
;;;  ;;
;;;  ;;[hatazaki 173] Checks vowel power at =(+ ?vowel-region 20) through =(+ ?vowel-region 40)
;;;  ;; rather than =(+ ?vowel-region 10) =(+ ?vowel-region 20),
;;;  ;; since ?vowel-region is a little bit left of the real vowel region when obtained by
;;;  ;; power-increase.
;;;  ;;[hatazaki 151] Checks vowel using powerat =(+ ?vowel-region 10) =(+ ?vowel-region 20),
;;;  ;; <-- =(+ ?vowel-region 20) =(+ ?vowel-region 40)
;;;  ;;
;;;  (power-strength =(+ ?vowel-region 20) =(+ ?vowel-region 40) 0 500 ?voicing-power)
;;;  (CF (vowel 0-500-power ?voicing-power) ?CFvowel)
;;;  ;;
;;;  ;;[hatazaki 174] Searches ?vowel-start before =(+ ?vowel-region 40) instead of
;;;  ;; =(+ ?vowel-region 20), since ?voicing-power is got for the region from
;;;  ;; =(+ ?vowel-region 20) =(+ ?vowel-region 40).
;;;  ;;[hatazaki 172] Searches ?vowel-start using (- ?voicing-power 10) instead of
;;;  ;; (max -45 (- ?voicing-power 20)), since ?voicing-power may be smaller than -45dB.
;;;  ;;[hatazaki 137] Uses power-start rather than power-increase, to find ?vowel-start.
;;;  ;; 01d code ...
;;;  ;; (power-increase (?vowel-start&~NONE ?change)
;;;  ;;   before =(+ ?vowel-region 20) 150 0 500
;;;  ;;   ?*normal-smoothing-size*
;;;  ;;   ?*vowel-start-min-change-after-unvoiced-stop*)
;;;  (power-start ?vowel-start&~NONE
;;;    before =(+ ?vowel-region 40) 100 0 500 =(- ?voicing-power 10))
;;;  ;;
;;;  ;; [hatazaki 130] Added checking of voice-onset-time.
;;;  ;;
;;;  (CF (unvoiced-stop voice-onset-time =(- ?vowel-start ?cl-to)) ?CFvot)
;;;  =>
;;;  (assert (applied ?segment sg-uvstop-bfr-vowel-2a))
;;;  (retract ?x)
;;;  (assert (prop ?segment (following-vowel-start ?vowel-start)))
;;;  (assert (CF (right-context ?segment vowel)
;;;    =(CFand (CFcomb ?CFcontext
;;;      (CFweight ?*evidence* ?CFvowel)
;;;      (CFweight ?*weak-evidence* ?CFvot))
;;;    ?CFvowel ?CFvot))))
;;;
;;;(defrule sg-uvstop-bfr-vowel-2b
;;;  "No burst, but having aspiration-power-in-low-frequency."
;;;  ;; (cf. MAU_B_0042 /ch/)
;;;  ;;
;;;  ;; [hatazaki 107]

```

```

;;; (declare (salience ?*left-segmentation*))
;;; (segment-status ?segment segmentation)
;;; (join
;;;   (category ?segment unvoiced-stop)
;;;   (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
;;; (join
;;;   (right-context ?segment vowel)
;;;   ?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
;;; (has-burst ?segment no)
;;; (prop ?segment (has-aspiration-power-in-low-frequency yes))
;;; (not (applied ?segment sg-uvstop-bfr-vowel-2b))
;;; ;;(prop ?segment (candidate-loc ? ?cand-to))
;;; (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
;;; ;;
;;; ;;[hatazaki 177] Makes the searching range longer;
;;; ;; after =(- ?cl-to 20) 150 0 500 <-- after =(- ?cl-to 20) 70 0 500
;;; ;; ;[hatazaki 172] Use power-increase instead of power-start, to find ?vowel-region,
;;; ;; since voicing power may be small.
;;; ;; <--
;;; ;; (power-start ?aspiration&~NONE
;;; ;;   after =(- ?cl-to 20) 70 0 500 ?*voicing-power-after-stop-lower*)
;;; ;; (power-start ?vowel-region&~NONE
;;; ;;   after ?aspiration 70 0 500 ?*voicing-power-after-stop-lower*)
;;; ;;
;;; (power-increase (?aspiration&~NONE ?aspiration-change)
;;;   after =(- ?cl-to 20) 150 0 500
;;;   ?*normal-smoothing-size* ?*default-min-change*)
;;; (power-increase (?vowel-region&~NONE ?vowel-region-change)
;;;   after ?aspiration 150 0 500
;;;   ?*normal-smoothing-size* ?*default-min-change*)
;;; (power-strength =(+ ?vowel-region 20) =(+ ?vowel-region 40) 0 500 ?voicing-power)
;;; (CF (vowel 0-500-power ?voicing-power) ?CFvowel)
;;; ;;
;;; ;;[hatazaki 172] Searches ?vowel-start using (- ?voicing-power 10) instead of
;;; ;; (max -45 (- ?voicing-power 20)), since ?voicing-power may be smaller than -45dB.
;;; ;; ;[hatazaki 137] Uses power-start rather than power-increase, to find ?vowel-start.
;;; ;; Old code ...
;;; ;; (power-increase (?vowel-start&~NONE ?change)
;;; ;;   before =(+ ?vowel-region 20) 150 0 500
;;; ;;   ?*normal-smoothing-size*
;;; ;;   ?*vowel-start-min-change-after-unvoiced-stop*)
;;; (power-start ?vowel-start&~NONE
;;;   before =(+ ?vowel-region 40) 100 0 500 =(- ?voicing-power 10))
;;; ;;
;;; ;; [hatazaki 130] Added checking of voice-onset-time.
;;; ;;
;;; (CF (unvoiced-stop voice-onset-time =(- ?vowel-start ?cl-to)) ?CFvot)
;;; =>
;;; (assert (applied ?segment sg-uvstop-bfr-vowel-2b))
;;; (retract ?x)
;;; (assert (prop ?segment (following-vowel-start ?vowel-start)))
;;; (assert (CF (right-context ?segment vowel)
;;;   =(CFand (CFcomb ?CFcontext
;;;     (CFweight ?*evidence* ?CFvowel)
;;;     (CFweight ?*weak-evidence* ?CFvot))
;;;   ?CFvowel ?CFvot))))
;;;
;;; ;[hatazaki 178]
;;; - The unvoiced stop may or may not have a burst, both of these cases
;;; are treated by the following rules. Rules sg-uvstop-bfr-vowel-2a,
;;; sg-uvstop-bfr-vowel-2b are removed.
;;;
;;; ;[hatazaki 174, 12/09/88 10:48:21]
;;; - For right segmentation of unvoiced stops, changed the strategy to
;;; find the following vowel start time when the stop has a burst. New
;;; strategy hypothesizes three cases, (1) there is not burst power nor
;;; aspiration power in low frequency, (2) there is burst power, but not
;;; aspiration power, (3) there is both burst power and aspiration power.
;;; Modified sg-uvstop-bfr-vowel-3a, sg-uvstop-bfr-vowel-3b,
;;; sg-uvstop-bfr-vowel-3c.
;;; Removed sg-uvstop-bfr-vowel-3d.
;;; Also defined sg-uvstop-8 to hypothesize the existence of the burst
;;; power.
;;;

```

```

(defrule sg-uvstop-bfr-vowel-3a
  "has a burst, but not burst power and aspiration power in low freq."
  ;; [hatazaki 107]
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
  (join
    (right-context ?segment vowel)
    ?x <- (CF (right-context ?segment vowel) ?CFcontext&:mightbe-valid))
  ;;
  ;;[hatazaki 177] Comment out the following.
  ;; (join
  ;;   (has-burst ?segment yes)
  ;;   (CF (has-burst ?segment yes) ?CFburst&:mightbe-valid)
  ;;   (prop ?segment (burst ?bu-from ?bu-to ?bu-freq)))
  ;;
  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
  (prop ?segment (has-burst-power-in-low-frequency no))
  (prop ?segment (has-aspiration-power-in-low-frequency no))
  (not (applied ?segment sg-uvstop-bfr-vowel-3a))
  ;;
  ;;[hatazaki 177] Searches from =(- ?cl-to 20) <-- =(- ?bu-from 20).
  ;;[hatazaki 172] Use power-increase instead of power-start, to find ?vowel-region,
  ;; since voicing power may be small.
  ;; <--
  ;; (power-start ?vowel-region&~NONE
  ;;   after =(- ?bu-from 20) 200 0 500 ?*voicing-power-after-stop-lower*)
  ;;
  (power-increase (?vowel-region&~NONE ?vowel-region-change)
    after =(- ?cl-to 20) 150 0 500
    ?*normal-smoothing-size* ?*default-min-change*)
  (power-strength =(+ ?vowel-region 20) =(+ ?vowel-region 40) 0 500 ?voicing-power)
  (CF (vowel 0-500-power ?voicing-power) ?CFvowel)
  (power-start ?vowel-start&~NONE
    before =(+ ?vowel-region 40) 100 0 500 =(- ?voicing-power 10))
  ;;
  ;;[hatazaki 177] =(- ?vowel-start ?cl-to) <-- =(- ?vowel-start ?bu-from)
  ;;[hatazaki 150] VOT=(- ?vowel-start ?bu-from) <-- (- ?vowel-start ?bu-to).
  ;;[hatazaki 130] Added checking of voice-onset-time.
  ;;
  (CF (unvoiced-stop voice-onset-time =(- ?vowel-start ?cl-to)) ?CFvot)
  =>
  (assert (applied ?segment sg-uvstop-bfr-vowel-3a))
  (retract ?x)
  (assert (prop ?segment (following-vowel-start ?vowel-start)))
  ;;[hatazaki 178] (CFweight ?*strong-evidence* ?CFvowel)
  ;;   <-- (CFweight ?*evidence* ?CFvowel)
  ;;[hatazaki 142] New CF of (right-context ?segment vowel),
  ;;   =(CFcomb ?CFcontext (CFand ?CFvowel ?CFvot)) -->
  ;;   =(CFcomb ?CFcontext
  ;;     (CFweight ?*evidence* ?CFvowel)
  ;;     (CFweight ?*evidence* ?CFvot))
  ;;
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFcontext
      (CFweight ?*strong-evidence* ?CFvowel)
      (CFweight ?*weak-evidence* ?CFvot))
    ?CFvowel ?CFvot))))))

(defrule sg-uvstop-bfr-vowel-3b
  "Has burst and burst power in low frequency, but not aspiration power."
  ;; [hatazaki 107]
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
  (join
    (right-context ?segment vowel)
    ?x <- (CF (right-context ?segment vowel) ?CFcontext&:mightbe-valid))
  ;;
  ;; (join
  ;;   (has-burst ?segment yes)
  ;;   (CF (has-burst ?segment yes) ?CFburst&:mightbe-valid)
  ;;   (prop ?segment (burst ?bu-from ?bu-to ?bu-freq)))
  ;;
  (prop ?segment (burst ?bu-from ?bu-to ?bu-freq))
  ;;
  (prop ?segment (has-burst-power-in-low-frequency no))
  (prop ?segment (has-aspiration-power-in-low-frequency no))
  (not (applied ?segment sg-uvstop-bfr-vowel-3b))
  ;;
  ;;[hatazaki 177] Searches from =(- ?cl-to 20) <-- =(- ?bu-from 20).
  ;;[hatazaki 172] Use power-increase instead of power-start, to find ?vowel-region,
  ;; since voicing power may be small.
  ;; <--
  ;; (power-start ?vowel-region&~NONE
  ;;   after =(- ?bu-from 20) 200 0 500 ?*voicing-power-after-stop-lower*)
  ;;
  (power-increase (?vowel-region&~NONE ?vowel-region-change)
    after =(- ?cl-to 20) 150 0 500
    ?*normal-smoothing-size* ?*default-min-change*)
  (power-strength =(+ ?vowel-region 20) =(+ ?vowel-region 40) 0 500 ?voicing-power)
  (CF (vowel 0-500-power ?voicing-power) ?CFvowel)
  (power-start ?vowel-start&~NONE
    before =(+ ?vowel-region 40) 100 0 500 =(- ?voicing-power 10))
  ;;
  ;;[hatazaki 177] =(- ?vowel-start ?cl-to) <-- =(- ?vowel-start ?bu-from)
  ;;[hatazaki 150] VOT=(- ?vowel-start ?bu-from) <-- (- ?vowel-start ?bu-to).
  ;;[hatazaki 130] Added checking of voice-onset-time.
  ;;
  (CF (unvoiced-stop voice-onset-time =(- ?vowel-start ?cl-to)) ?CFvot)
  =>
  (assert (applied ?segment sg-uvstop-bfr-vowel-3b))
  (retract ?x)
  (assert (prop ?segment (following-vowel-start ?vowel-start)))
  ;;[hatazaki 178] (CFweight ?*strong-evidence* ?CFvowel)
  ;;   <-- (CFweight ?*evidence* ?CFvowel)
  ;;[hatazaki 142] New CF of (right-context ?segment vowel),
  ;;   =(CFcomb ?CFcontext (CFand ?CFvowel ?CFvot)) -->
  ;;   =(CFcomb ?CFcontext
  ;;     (CFweight ?*evidence* ?CFvowel)
  ;;     (CFweight ?*evidence* ?CFvot))
  ;;
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFcontext
      (CFweight ?*strong-evidence* ?CFvowel)
      (CFweight ?*weak-evidence* ?CFvot))
    ?CFvowel ?CFvot))))))

```

```

(prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
(prop ?segment (has-burst-power-in-low-frequency yes))
(prop ?segment (has-aspiration-power-in-low-frequency no))
(not (applied ?segment sg-uvstop-bfr-vowel-3b))
;;
;;[hatazaki 177] Makes the searching range longer;
;; after =(- ?cl-to 20) 150 0 500 <-- after =(- ?cl-to 20) 70 0 500
;;[hatazaki 172] Use power-increase instead of power-start, to find ?vowel-region,
;; since voicing power may be small.
;; <--
;; (power-start ?burst-region&~NONE
;;   after =(- ?bu-from 20) 50 0 500 ?*voicing-power-after-stop-lower*)
;; (power-start ?vowel-region&~NONE
;;   after ?burst-region 200 0 500 ?*voicing-power-after-stop-lower*)
;;
(power-increase (?burst&~NONE ?burst-change)
  after =(- ?cl-to 20) 150 0 500
  ?*normal-smoothing-size* ?*default-min-change*)
(power-increase (?vowel-region&~NONE ?vowel-region-change)
  after ?burst 150 0 500
  ?*normal-smoothing-size* ?*default-min-change*)
(power-strength =(+ ?vowel-region 20) =(+ ?vowel-region 40) 0 500 ?voicing-power)
(CF (vowel 0-500-power ?voicing-power) ?CFvowel)
(power-start ?vowel-start&~NONE
  before =(+ ?vowel-region 40) 100 0 500 =(- ?voicing-power 10))
;;
;;[hatazaki 150] VOT=(- ?vowel-start ?bu-from) <-- (- ?vowel-start ?bu-to).
;; [hatazaki 130] Added checking of voice-onset-time.
;;
(CF (unvoiced-stop voice-onset-time =(- ?vowel-start ?cl-to)) ?CFvot)
=>
(assert (applied ?segment sg-uvstop-bfr-vowel-3b))
(retract ?x)
(assert (prop ?segment (following-vowel-start ?vowel-start)))
;;
;; [hatazaki 142] New CF of (right-context ?segment vowel),
;;   =(CFcomb ?CFcontext (CFand ?CFvowel ?CFvot)) -->
;;   =(CFcomb ?CFcontext
;;     (CFweight ?*evidence* ?CFvowel)
;;     (CFweight ?*evidence* ?CFvot))
;;
(assert (CF (right-context ?segment vowel)
  =(CFand (CFcomb ?CFcontext
    (CFweight ?*strong-evidence* ?CFvowel)
    (CFweight ?*weak-evidence* ?CFvot))
  ?CFvowel ?CFvot))))

(defrule sg-uvstop-bfr-vowel-3c
  "Has burst, burst power and aspiration power in low frequency."
  ;; [hatazaki 107]
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment vowel)
    ?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
  ;; (join
  ;;   (has-burst ?segment yes)
  ;;   (CF (has-burst ?segment yes) ?CFburst&: mightbe-valid)
  ;;   (prop ?segment (burst ?bu-from ?bu-to ?bu-freq)))
  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
  (prop ?segment (has-burst-power-in-low-frequency yes))
  (prop ?segment (has-aspiration-power-in-low-frequency yes))
  (not (applied ?segment sg-uvstop-bfr-vowel-3c))
  ;;
  ;;[hatazaki 172] Use power-increase instead of power-start, to find ?vowel-region,
  ;; since voicing power may be small.
  ;; <--
  ;; (power-start ?burst-region&~NONE
  ;;   after =(- ?bu-from 20) 50 0 500 ?*voicing-power-after-stop-lower*)
  ;; (power-start ?vowel-region&~NONE
  ;;   after ?burst-region 200 0 500 ?*voicing-power-after-stop-lower*)
  ;;
  (power-increase (?burst&~NONE ?burst-change)

```

```

        after =(- ?c1-to 20) 7150 0 500
        ?*normal-smoothing-size* ?*default-min-change*
(power-increase (?aspiration&~NONE ?aspiration-change)
  after ?burst 150 0 500
  ?*normal-smoothing-size* ?*default-min-change*
(power-increase (?vowel-region&~NONE ?vowel-region-change)
  after ?aspiration 150 0 500
  ?*normal-smoothing-size* ?*default-min-change*
(power-strength =(+ ?vowel-region 20) =(+ ?vowel-region 40) 0 500 ?voicing-power)
(CF (vowel 0-500-power ?voicing-power) ?CFvowel)
(power-start ?vowel-start&~NONE
  before =(+ ?vowel-region 40) 100 0 500 =(- ?voicing-power 10))
;;
;;[hatazaki 150] VOT=(- ?vowel-start ?bu-from) <-- (- ?vowel-start ?bu-to).
;; [hatazaki 130] Added checking of voice-onset-time.
;;
(CF (unvoiced-stop voice-onset-time =(- ?vowel-start ?c1-to)) ?CFvot)
=>
(assert (applied ?segment sg-uvstop-bfr-vowel-3c))
(retract ?x)
(assert (prop ?segment (following-vowel-start ?vowel-start)))
;;
;; [hatazaki 142] New CF of (right-context ?segment vowel),
;;   =(CFcomb ?CFcontext (CFand ?CFvowel ?CFvot)) -->
;;   =(CFcomb ?CFcontext
;;     (CFweight ?*evidence* ?CFvowel)
;;     (CFweight ?*evidence* ?CFvot))
;;
(assert (CF (right-context ?segment vowel)
  =(CFand (CFcomb ?CFcontext
    (CFweight ?*strong-evidence* ?CFvowel)
    (CFweight ?*weak-evidence* ?CFvot))
  ?CFvowel ?CFvot))))
;;
-----
;;;
;;;
;;;(defrule sg-uvstop-bfr-vowel-3a
;;;  "has a burst, but not a double-burst."
;;;  ;; [hatazaki 107]
;;;  (declare (salience ?*left-segmentation*))
;;;  (segment-status ?segment segmentation)
;;;  (join
;;;    (category ?segment unvoiced-stop)
;;;    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
;;;  (join
;;;    (right-context ?segment vowel)
;;;    ?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
;;;  (join
;;;    (has-burst ?segment yes)
;;;    (CF (has-burst ?segment yes) ?CFburst&: mightbe-valid)
;;;    (prop ?segment (burst ?bu-from ?bu-to ?bu-freq)))
;;;  (join
;;;    (prop ?segment (has-double-burst no))
;;;    (CF (prop ?segment (has-double-burst no)) ?CFno-double&: mightbe-valid))
;;;  (prop ?segment (has-aspiration-power-in-low-frequency no))
;;;  (not (applied ?segment sg-uvstop-bfr-vowel-3a))
;;;  ;;
;;;  ;;[hatazaki 172] Use power-increase instead of power-start, to find ?vowel-region,
;;;  ;; since voicing power may be small.
;;;  ;; <--
;;;  ;; (power-start ?vowel-region&~NONE
;;;  ;;   after =(- ?bu-from 20) 200 0 500 ?*voicing-power-after-stop-lower*)
;;;  ;;
;;;  (power-increase (?vowel-region&~NONE ?vowel-region-change)
;;;    after =(- ?bu-from 20) 150 0 500
;;;    ?*normal-smoothing-size* ?*default-min-change*
;;;  (power-strength =(+ ?vowel-region 20) =(+ ?vowel-region 40) 0 500 ?voicing-power)
;;;  (CF (vowel 0-500-power ?voicing-power) ?CFvowel)
;;;  (power-start ?vowel-start&~NONE
;;;    before =(+ ?vowel-region 20) 100 0 500 =(- ?voicing-power 10))
;;;  ;;
;;;  ;;[hatazaki 150] VOT=(- ?vowel-start ?bu-from) <-- (- ?vowel-start ?bu-to).
;;;  ;; [hatazaki 130] Added checking of voice-onset-time.
;;;  ;;
;;;  (CF (unvoiced-stop voice-onset-time =(- ?vowel-start ?bu-from)) ?CFvot)
;;;  =>

```

```

;;; (assert (applied ?segment sg-uvstop-bfr-vowel-3a))
;;; (retract ?x)
;;; (assert (prop ?segment (following-vowel-start ?vowel-start)))
;;; ;;
;;; ;; [hatazaki 142] New CF of (right-context ?segment vowel),
;;; ;;   =(CFcomb ?CFcontext (CFand ?CFvowel ?CFvot)) -->
;;; ;;   =(CFcomb ?CFcontext
;;; ;;     (CFweight ?*evidence* ?CFvowel)
;;; ;;     (CFweight ?*evidence* ?CFvot))
;;; ;;
;;; (assert (CF (right-context ?segment vowel)
;;;   =(CFand (CFcomb ?CFcontext
;;;     (CFweight ?*evidence* ?CFvowel)
;;;     (CFweight ?*weak-evidence* ?CFvot))
;;;   ?CFvowel ?CFvot ?CFno-double))))
;;;
;;;(defrule sg-uvstop-bfr-vowel-3b
;;;  "Has burst and has-burst-power-in-low-frequency"
;;;  ;; [hatazaki 107]
;;;  (declare (salience ?*left-segmentation*))
;;;  (segment-status ?segment segmentation)
;;;  (join
;;;    (category ?segment unvoiced-stop)
;;;    (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
;;;  (join
;;;    (right-context ?segment vowel)
;;;    ?x <- (CF (right-context ?segment vowel) ?CFcontext&:mightbe-valid))
;;;  (join
;;;    (has-burst ?segment yes)
;;;    (prop ?segment (burst ?bu-from ?bu-to ?bu-freq)))
;;;  (join
;;;    (prop ?segment (has-double-burst no))
;;;    (CF (prop ?segment (has-double-burst no)) ?CFno-double&:mightbe-valid))
;;;  (prop ?segment (has-aspiration-power-in-low-frequency yes))
;;;  (not (applied ?segment sg-uvstop-bfr-vowel-3b))
;;;  ;;
;;;  ;;[hatazaki 172] Use power-increase instead of power-start, to find ?vowel-region,
;;;  ;; since voicing power may be small.
;;;  ;; <--
;;;  ;; (power-start ?burst-region&~NONE
;;;  ;;   after =(- ?bu-from 20) 50 0 500 ?*voicing-power-after-stop-lower*)
;;;  ;; (power-start ?vowel-region&~NONE
;;;  ;;   after ?burst-region 200 0 500 ?*voicing-power-after-stop-lower*)
;;;  ;;
;;;  (power-increase (?aspiration&~NONE ?aspiration-change)
;;;    after =(- ?bu-from 20) 70 0 500
;;;    ?*normal-smoothing-size* ?*default-min-change*)
;;;  (power-increase (?vowel-region&~NONE ?vowel-region-change)
;;;    after ?aspiration 150 0 500
;;;    ?*normal-smoothing-size* ?*default-min-change*)
;;;  (power-strength =(+ ?vowel-region 20) =(+ ?vowel-region 40) 0 500 ?voicing-power)
;;;  (CF (vowel 0-500-power ?voicing-power) ?CFvowel)
;;;  (power-start ?vowel-start&~NONE
;;;    before =(+ ?vowel-region 20) 100 0 500 =(- ?voicing-power 10))
;;;  ;;
;;;  ;;[hatazaki 150] VOT=(- ?vowel-start ?bu-from) <-- (- ?vowel-start ?bu-to).
;;;  ;; [hatazaki 130] Added checking of voice-onset-time.
;;;  ;;
;;;  (CF (unvoiced-stop voice-onset-time =(- ?vowel-start ?bu-from)) ?CFvot)
;;;  =>
;;;  (assert (applied ?segment sg-uvstop-bfr-vowel-3b))
;;;  (retract ?x)
;;;  (assert (prop ?segment (following-vowel-start ?vowel-start)))
;;;  ;;
;;;  ;; [hatazaki 142] New CF of (right-context ?segment vowel),
;;;  ;;   =(CFcomb ?CFcontext (CFand ?CFvowel ?CFvot)) -->
;;;  ;;   =(CFcomb ?CFcontext
;;;  ;;     (CFweight ?*evidence* ?CFvowel)
;;;  ;;     (CFweight ?*evidence* ?CFvot))
;;;  ;;
;;;  (assert (CF (right-context ?segment vowel)
;;;    =(CFand (CFcomb ?CFcontext
;;;      (CFweight ?*evidence* ?CFvowel)
;;;      (CFweight ?*weak-evidence* ?CFvot))
;;;    ?CFvowel ?CFvot ?CFno-double))))

```



```

;;;(defrule sg-uvstop-bfr-vowel-3c
;;;  "Double burst exists"
;;;  ;; [hatazaki 107]
;;;  (declare (salience ?*left-segmentation*))
;;;  (segment-status ?segment segmentation)
;;;  (join
;;;    (category ?segment unvoiced-stop)
;;;    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
;;;  (join
;;;    (right-context ?segment vowel)
;;;    ?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
;;;  (join
;;;    (has-burst ?segment yes)
;;;    (prop ?segment (burst ?bu-from ?bu-to ?bu-freq)))
;;;  (join
;;;    (prop ?segment (has-double-burst yes))
;;;    (CF (prop ?segment (has-double-burst yes)) ?CFd-burst&: mightbe-valid)
;;;    (prop ?segment (2nd-burst ?2nd-burst-from ?2nd-burst-to ?2nd-burst-freq)))
;;;  (prop ?segment (has-aspiration-power-in-low-frequency no))
;;;  (not (applied ?segment sg-uvstop-bfr-vowel-3c))
;;;  ;;
;;;  ;;[hatazaki 172] Use power-increase instead of power-start, to find ?vowel-region,
;;;  ;; since voicing power may be small.
;;;  ;; <--
;;;  ;; (power-start ?vowel-region&~NONE
;;;  ;;   after =(- ?2nd-burst-from 20) 200 0 500 ?*voicing-power-after-stop-lower*)
;;;  ;;[hatazaki 150] after =(- ?2nd-burst-from 20) <-- =(- ?2nd-burst-to 20).
;;;  ;;
;;;  (power-increase (?vowel-region&~NONE ?vowel-region-change)
;;;    after =(- ?2nd-burst-from 20) 100 0 500
;;;    ?*normal-smoothing-size* ?*default-min-change*)
;;;  (power-strength =(+ ?vowel-region 20) =(+ ?vowel-region 40) 0 500 ?voicing-power)
;;;  (CF (vowel 0-500-power ?voicing-power) ?CFvowel)
;;;  (power-start ?vowel-start&~NONE
;;;    before =(+ ?vowel-region 20) 100 0 500 =(- ?voicing-power 10))
;;;  ;;
;;;  ;;[hatazaki 150] VOT=(- ?vowel-start ?2nd-burst-from) <-- (- ?vowel-start ?2nd-burst-to).
;;;  ;;[hatazaki 130] Added checking of voice-onset-time.
;;;  ;;
;;;  (CF (unvoiced-stop voice-onset-time =(- ?vowel-start ?2nd-burst-from)) ?CFvot)
;;;  =>
;;;  (assert (applied ?segment sg-uvstop-bfr-vowel-3c))
;;;  (retract ?x)
;;;  (assert (prop ?segment (second-burst ?2nd-burst-from ?2nd-burst-to ?2nd-burst-freq)))
;;;  (assert (prop ?segment (following-vowel-start ?vowel-start)))
;;;  ;;
;;;  ;; [hatazaki 142] New CF of (right-context ?segment vowel),
;;;  ;;   =(CFcomb ?CFcontext (CFand ?CFvowel ?CFvot)) -->
;;;  ;;   =(CFcomb ?CFcontext
;;;  ;;     (CFweight ?*evidence* ?CFvowel)
;;;  ;;     (CFweight ?*evidence* ?CFvot))
;;;  ;;
;;;  (assert (CF (right-context ?segment vowel)
;;;    =(CFand (CFcomb ?CFcontext
;;;      (CFweight ?*evidence* ?CFvowel)
;;;      (CFweight ?*weak-evidence* ?CFvot))
;;;    ?CFvowel ?CFvot ?CFd-burst))))
;;;(defrule sg-uvstop-bfr-vowel-3d
;;;  "Double burst exists and has-burst-power-in-low-frequency."
;;;  ;; [hatazaki 107]
;;;  (declare (salience ?*left-segmentation*))
;;;  (segment-status ?segment segmentation)
;;;  (join
;;;    (category ?segment unvoiced-stop)
;;;    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
;;;  (join
;;;    (right-context ?segment vowel)
;;;    ?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
;;;  (join
;;;    (has-burst ?segment yes)
;;;    (prop ?segment (burst ?bu-from ?bu-to ?bu-freq)))
;;;  (join
;;;    (prop ?segment (has-double-burst yes))
;;;    (CF (prop ?segment (has-double-burst yes)) ?CFd-burst&: mightbe-valid)
;;;    (prop ?segment (2nd-burst ?2nd-burst-from ?2nd-burst-to ?2nd-burst-freq)))

```

```

;;; (prop ?segment (has-aspiration-power-in-low-frequency yes))
;;; (not (applied ?segment sg-uvstop-bfr-vowel-3d))
;;; ;;
;;; ;;[hatazaki 172] Use power-increase instead of power-start, to find ?vowel-region,
;;; ;; since voicing power may be small.
;;; ;; <--
;;; ;; (power-start ?burst-region&~NONE
;;; ;;   after =(- ?2nd-burst-from 20) 50 0 500 ?*voicing-power-after-stop-lower*)
;;; ;; (power-start ?vowel-region&~NONE
;;; ;;   after ?burst-region 200 0 500 ?*voicing-power-after-stop-lower*)
;;; ;;
;;; ;;[hatazaki 150] after =(- ?2nd-burst-from 20) <-- =(- ?2nd-burst-to 20).
;;; ;;
;;; (power-increase (?aspiration&~NONE ?aspiration-change)
;;;   after =(- ?2nd-burst-from 20) 50 0 500
;;;     ?*normal-smoothing-size* ?*default-min-change*)
;;; (power-increase (?vowel-region&~NONE ?vowel-region-change)
;;;   after ?aspiration 100 0 500
;;;     ?*normal-smoothing-size* ?*default-min-change*)
;;; (power-strength =(+ ?vowel-region 20) =(+ ?vowel-region 40) 0 500 ?voicing-power)
;;; (CF (vowel 0-500-power ?voicing-power) ?CFvowel)
;;; (power-start ?vowel-start&~NONE
;;;   before =(+ ?vowel-region 20) 100 0 500 =(- ?voicing-power 10))
;;; ;;
;;; ;;[hatazaki 150] VOT=(- ?vowel-start ?2nd-burst-from) <-- (- ?vowel-start ?2nd-burst-to).
;;; ;;[hatazaki 130] Added checking of voice-onset-time.
;;; ;;
;;; (CF (unvoiced-stop voice-onset-time =(- ?vowel-start ?2nd-burst-from)) ?CFvot)
;;; =>
;;; (assert (applied ?segment sg-uvstop-bfr-vowel-3d))
;;; (retract ?x)
;;; (assert (prop ?segment (second-burst ?2nd-burst-from ?2nd-burst-to ?2nd-burst-freq)))
;;; (assert (prop ?segment (following-vowel-start ?vowel-start)))
;;; ;;
;;; ;; [hatazaki 142] New CF of (right-context ?segment vowel),
;;; ;;   =(CFcomb ?CFcontext (CFand ?CFvowel ?CFvot)) -->
;;; ;;   =(CFcomb ?CFcontext
;;; ;;     (CFweight ?*evidence* ?CFvowel)
;;; ;;     (CFweight ?*evidence* ?CFvot))
;;; ;;
;;; (assert (CF (right-context ?segment vowel)
;;;   =(CFand (CFcomb ?CFcontext
;;;     (CFweight ?*evidence* ?CFvowel)
;;;     (CFweight ?*weak-evidence* ?CFvot))
;;;   ?CFvowel ?CFvot ?CFd-burst))))
(defrule sg-uvstop-bfr-vowel-4a
  "Checks there is another closure before the vowel."
  ;;
  ;; - If there is another closure (low power) before the vowel,
  ;; the vowel is not what we want, that is, a vowel is devoiced. [hatazaki 108]
  ;;
  (declare (saliency ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
  (join
    (right-context ?segment vowel)
    ?x <- (CF (right-context ?segment vowel) ?CFcontext&:mightbe-valid))
    (not (applied ?segment sg-uvstop-bfr-vowel-4a))
    ;(prop ?segment (candidate-loc ? ?cand-to))
    (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
    (has-burst ?segment no)
    (prop ?segment (following-vowel-start ?vowel-start))
  )
  ;;
  ;;[hatazaki 165] Checks (< ?bu-from ?another-closure-start).
  ;;[hatazaki 161] Uses (closure-exists ...) pattern to checks the existence of a closure.
  ;; <--
  ;; (power-start ?another-closure-end&~NONE
  ;;   before =(+ ?vowel-start 20) 60
  ;;   1000 6000
  ;;   ?*closure-power-upper*)
  ;; (power-end ?another-closure-start&~NONE
  ;;   before ?another-closure-end =(- ?another-closure-end ?cl-to)
  ;;   1000 6000
  ;; )

```

```

;;      ?*closure-power-upper*)
;; (power-strength ?another-closure-start ?another-closure-end 0 6000
;;      ?another-closure-power)
;; (CF (unvoiced-stop closure-0-6000-power ?another-closure-power) ?CFclosure-power)
;;
;; - Search the burst backward from ?vowel-start, since there are some
;; power change in the devoiced vowel region. [hatazaki 115]
;; Old code ...
;; (power-end ?another-closure-start&~none
;;      after ?cand-to =(- ?vowel-start ?cand-to)
;;      1000 6000
;;      ?*closure-power-upper*)
;; (power-start ?another-closure-end&~none
;;      after ?another-closure-start =(- ?vowel-start ?another-closure-start)
;;      1000 6000
;;      ?*closure-power-upper*)
(closure-exists ?another-closure-start:< ?cl-to ?another-closure-start)
      ?another-closure-end
      before ?vowel-start =(- ?vowel-start ?cl-to 10)
      ?CFanother-closure)
=>
(assert (applied ?segment sg-uvstop-bfr-vowel-4a))
(retract ?x)
;;
;; [hatazaki 129] Changed (CFand ?CF (- ?CFclosure-power)) to (CFcomb
;; ?CF (- ?CFclosure-power)).
;;
(assert (CF (right-context ?segment vowel)
      =(CFand (CFcomb ?CFcontext
      (CFweight ?*strong-evidence*
      (- (CFvalue ?CFanother-closure))))
      (- (CFvalue ?CFanother-closure))))))

(defrule 'sg-uvstop-bfr-vowel-4b
  "Checks there is another closure before the vowel."
  ;;
  ;; - If there is another closure (low power) between the closure and the vowel,
  ;; the vowel is not what we want, that is, a vowel is devoiced. [hatazaki 108]
  ;;
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
  (join
    (right-context ?segment vowel)
    ?x <- (CF (right-context ?segment vowel) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvstop-bfr-vowel-4b))
  ;;(prop ?segment (candidate-loc ? ?cand-to))
  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
  (has-burst ?segment yes)
  (or (and (prop ?segment (has-double-burst no))
    (prop ?segment (burst ?bu-from ?bu-to ?bu-freq)))
    (and (prop ?segment (has-double-burst yes))
    (prop ?segment (second-burst ?bu-from ?bu-to ?bu-freq))))
  (prop ?segment (following-vowel-start ?vowel-start))
  ;;
  ;; [hatazaki 165] Checks (< ?bu-from ?another-closure-start).
  ;; [hatazaki 161] Uses (closure-exists ...) pattern to checks the existence of a closure.
  ;; <--
  ;; (power-start ?another-closure-end&~NONE
  ;;      before =(+ ?vowel-start 20) =(- ?vowel-start ?cand-to -20)
  ;;      1000 6000
  ;;      ?*closure-power-upper*)
  ;; (power-end ?another-closure-start&~NONE
  ;;      before ?another-closure-end =(- ?another-closure-end ?cand-to)
  ;;      1000 6000
  ;;      ?*closure-power-upper*)
  ;; (power-strength ?another-closure-start ?another-closure-end 0 6000
  ;;      ?another-closure-power)
  ;; (CF (unvoiced-stop closure-0-6000-power ?another-closure-power) ?CFclosure-power)
  ;;
  ;; [hatazaki 133] Changed the search range for ?another-closure-end,
  ;; from 60 to =(- ?vowel-start ?cand-to -20)
  ;; - Search the burst backward from ?vowel-start, since there are some
  ;; power change in the devoiced vowel region. [hatazaki 115]

```

```

;; Old code ...
;; (power-end ?another-closure-start&-none
;;   after ?cand-to =(- ?vowel-start ?cand-to)
;;   1000 6000
;;   ?*closure-power-upper*)
;; (power-start ?another-closure-end&-none
;;   after ?another-closure-start =(- ?vowel-start ?another-closure-start)
;;   1000 6000
;;   ?*closure-power-upper*)
;;
(closure-exists ?another-closure-start&:(< ?bu-from ?another-closure-start)
  ?another-closure-end
  before ?vowel-start =(- ?vowel-start ?bu-from 10)
  ?CFanother-closure)
;;[hatazaki 161] Removed checking aspiration-duration-before-another-stop, since
;; I don't know the cases when this check work.
;; ;;
;; ;;[hatazaki 150] aspiration-duration-before-another-stop
;; ;; =(- ?another-closure-start ?bu-from)
;; ;; <-- =(- ?another-closure-start ?bu-to)
;; ;;
;; (CF (unvoiced-stop aspiration-duration-before-another-stop
;;   =(- ?another-closure-start ?bu-from))
;;   ?CFaspiration)
=>
(assert (applied ?segment sg-uvstop-bfr-vowel-4b))
(retract ?x)
;;[hatazaki 161] Use ?CFanother-closure.
;; <-- (assert (CF (right-context ?segment vowel)
;;   =(CFand (CFcomb ?CFcontext
;;     (CFweight ?*strong-evidence*
;;       (- (CFvalue (CFfor ?CFclosure-power ?CFaspiration))))))
;;   (- (CFvalue (CFfor ?CFclosure-power ?CFaspiration))))))
;; [hatazaki 129] Changed (CFand ?CF'(- (CFand ?CFclosure-power ?CFaspiration)))
;; to (CFcomb ?CF (- (CFfor ?CFclosure-power ?CFaspiration)))
;;
(assert (CF (right-context ?segment vowel)
  =(CFand (CFcomb ?CFcontext
    (CFweight ?*strong-evidence*
      (- (CFvalue ?CFanother-closure))))))
  (- (CFvalue ?CFanother-closure))))))

(defrule sg-uvstop-bfr-vowel-5
  "Checks there is no voicing power before the vowel."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment vowel)
    ?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvstop-bfr-vowel-5))
  (prop ?segment (following-vowel-start ?vowel-start))
  (power-strength =(- ?vowel-start 30) =(- ?vowel-start 10) 0 500 ?0-500-power)
  (power-strength =(- ?vowel-start 30) =(- ?vowel-start 10) 500 1000 ?500-1000-power)
  (CF (unvoiced-stop 0-500-power-before-vowel ?0-500-power) ?CFpower)
  (CF (unvoiced-stop 0-500-500-1000-power-ratio-before-vowel
    =(- ?0-500-power ?500-1000-power)
    ?CFPR)
    ?CFPR)
  =>
  (assert (applied ?segment sg-uvstop-bfr-vowel-5))
  (retract ?x)
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFcontext
      (CFweight ?*no-evidence* ?CFpower))
      ?CFpower
      ?CFPR))))))

(defrule sg-uvstop-bfr-vowel-done-1
  ""
  ;;
  ;; [hatazaki 107]
  ;;
  (declare (salience ?*right-segmentation-end*))
  (segment-status ?segment segmentation)

```

```
(join
  (category ?segment unvoiced-stop)
  (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
(join
  (right-context ?segment vowel)
  (CF (right-context ?segment vowel) ?CFcontext&:mightbe-valid))
(not (applied ?segment sg-uvstop-bfr-vowel-done-1))
(prop ?segment (following-vowel-start ?vowel-start))
=>
(assert (applied ?segment sg-uvstop-bfr-vowel-done-1))
(hypothesize
  (assert (end-time ?segment ?vowel-start))
  (assert (CF (end-time ?segment ?vowel-start)
    =(CFcomb (CFweight ?*no-evidence* ?CFcategory)
      (CFweight ?*definite-evidence* ?CFcontext))))))
```

-----

```

(defrule sg-uvstop-bfr-stop-0
  "Finds a vowel after the stop."
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment stop)
    ?x <- (CF (right-context ?segment stop) ?CF&: mightbe-valid))
  (not (applied ?segment sg-uvstop-bfr-stop-0))
  ;;(prop ?segment (candidate-loc ? ?cand-to))
  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
  ;;
  ;;[hatazaki 166] Searches ?vowel-region longer.
  ;; after =(- ?cl-to 20) 300 <-- after =(- ?cl-to 20) 200.
  (power-start ?vowel-region&~NONE
    after =(- ?cl-to 20) 300 0 500 ?*voicing-power-after-stop-lower*)
  =>
  (assert (applied ?segment sg-uvstop-bfr-stop-0))
  (assert (prop ?segment (following-vowel-region ?vowel-region))))

;;;New rule for MAU_1_4516.
;;;
;(defrule sg-uvstop-bfr-stop-1
;  "right context is unvoiced-vowel and stop"
;  (declare (saliency ?*left-segmentation*))
;  (segment-status ?segment segmentation)
;  (join
;    (category ?segment unvoiced-stop)
;    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
;  (join
;    (right-context ?segment stop)
;    ?x <- (CF (right-context ?segment stop) ?CFcontext&: mightbe-valid))
;  (not (applied ?segment sg-uvstop-bfr-stop-1))
;  ;;(prop ?segment (candidate-loc ? ?cand-to))
;  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
;  (prop ?segment (following-vowel-region ?vowel-region))
;  ;;
;  ;;[hatazaki 176] User a backward-chaining rule for finding the closure.
;  ;; <--
;  ;; (power-start ?another-closure-end&~none
;  ;;   before =(+ ?vowel-region 20) =(- ?vowel-region ?cl-to -20)
;  ;;   1000 6000
;  ;;   ?*closure-power-upper*)
;  ;; (power-end ?another-closure-start&~none
;  ;;   before ?another-closure-end =(- ?another-closure-end ?cl-to)
;  ;;   1000 6000
;  ;;   ?*closure-power-upper*)
;  ;; (power-strength ?another-closure-start ?another-closure-end 0 6000
;  ;;   ?another-closure-power)
;  ;; (cf (unvoiced-stop closure-0-6000-power ?another-closure-power) ?cfclosure-power)
;  ;; (power-decrease (?closure-start&~none ?)
;  ;;   before =(+ ?another-closure-start 20) 50 1000 6000
;  ;;   ?*normal-smoothing-size* ?*closure-start-min-change*)
;  ;;[hatazaki 133] Changed the search range for ?another-closure-end,
;  ;; from 60 to =(- ?vowel-region ?cand-to -20)
;  ;;- Search the burst backward from ?vowel-region, since there are some
;  ;; power change in the devoiced vowel region. [hatazaki 115]
;  ;; Old code ...
;  ;; (power-end ?another-closure-start&~NONE
;  ;;   after ?cand-to 150
;  ;;   1000 6000
;  ;;   ?*closure-power-upper*)
;  ;; (power-start ?another-closure-end&~NONE
;  ;;   after ?another-closure-start 300
;  ;;   1000 6000
;  ;;   ?*closure-power-upper*)
;  ;;
;  (closure-exists ?following-closure-start ?following-closure-end
;    before =(+ ?vowel-region 20) 70 ?CFfollowing-closure)
;  =>
;  (assert (applied ?segment sg-uvstop-bfr-stop-1))
;  (assert (prop ?segment (following-closure-start ?following-closure-start)))

```

```

; (retract ?x)
; (assert (CF (right-context ?segment stop)
;           =(CFand (CFcomb ?CFcontext
;                       (CFweight ?*evidence* ?CFfollowing-closure)
;                       ?CFfollowing-closure))))
;; Old code.
(defrule sg-uvstop-bfr-stop-1
  "right context is unvoiced-vowel and stop"
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment stop)
    ?x <- (CF (right-context ?segment stop) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvstop-bfr-stop-1))
  ;;(prop ?segment (candidate-loc ? ?cand-to))
  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
  (prop ?segment (following-vowel-region ?vowel-region))
  ;;
  ;; [hatazaki 133] Changed the search range for ?another-closure-end,
  ;; from 60 to =(- ?vowel-region ?cand-to -20)
  ;; - Search the burst backward from ?vowel-region, since there are some
  ;; power change in the devoiced vowel region. [hatazaki 115]
  ;; Old code ...
  ;; (power-end ?another-closure-start&~NONE
  ;;   after ?cand-to 150
  ;;   1000 6000
  ;;   ?*closure-power-upper*)
  ;; (power-start ?another-closure-end&~NONE
  ;;   after ?another-closure-start 300
  ;;   1000 6000
  ;;   ?*closure-power-upper*)
  ;;
  ;;
  (power-start ?another-closure-end&~NONE
    before =(+ ?vowel-region 20) =(- ?vowel-region ?cl-to -20)
    1000 6000
    ?*closure-power-upper*)
  (power-end ?another-closure-start&~NONE
    before ?another-closure-end =(- ?another-closure-end ?cl-to)
    1000 6000
    ?*closure-power-upper*)
  (power-strength ?another-closure-start ?another-closure-end 0 6000
    ?another-closure-power)
  (CF (unvoiced-stop closure-0-6000-power ?another-closure-power) ?CFclosure-power)
  (power-decrease (?closure-start&~NONE ?) before =(+ ?another-closure-start 20) 50 1000 6000
    ?*normal-smoothing-size* ?*closure-start-min-change*)
  =>
  (assert (applied ?segment sg-uvstop-bfr-stop-1))
  (assert (prop ?segment (following-closure-start ?closure-start)))
  (retract ?x)
  (assert (CF (right-context ?segment stop)
    =(CFand (CFcomb ?CFcontext
      (CFweight ?*evidence* ?CFclosure-power)
      ?CFclosure-power))))
(defrule sg-uvstop-bfr-stop-2
  "Verify devoiced vowel."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment stop)
    ?x <- (CF (right-context ?segment stop) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvstop-bfr-stop-2))
  ;;(prop ?segment (candidate-loc ? ?cand-to))
  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
  (prop ?segment (following-closure-start ?another-closure-start))
  (power-start ?vowel-region&~NONE
    after =(- ?cl-to 20) =(- ?another-closure-start ?cl-to)
    0 500

```

```

        ?*voicing-power-after-stop-lower*)
(power-strength ?vowel-region =(+ ?vowel-region 30) 0 500 ?vowel-power)
(CF (vowel 0-500-power ?vowel-power) ?CFvowel-power)
=>
(assert (applied ?segment sg-uvstop-bfr-stop-2))
(retract ?x)
(assert (CF (right-context ?segment stop)
            =(CFand (CFcomb ?CFcontext
                          (CFweight ?*strong-evidence* (- (CFvalue ?CFvowel-power))))
                  (- (CFvalue ?CFvowel-power))))))

(defrule sg-uvstop-bfr-stop-3
  "Checks power before the end boundary."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
  (join
    (right-context ?segment stop)
    ?x <- (CF (right-context ?segment stop) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvstop-bfr-stop-3))
  (prop ?segment (following-closure-start ?another-closure-start))
  (power-strength =(- ?another-closure-start 30) =(- ?another-closure-start 10) 0 500
    ?0-500-power)
  (CF (unvoiced-stop 0-500-power-before-vowel ?0-500-power) ?CFpower)
  =>
  (assert (applied ?segment sg-uvstop-bfr-stop-3))
  (retract ?x)
  (assert (CF (right-context ?segment stop)
              =(CFand (CFcomb ?CFcontext
                            (CFweight ?*no-evidence* ?CFpower))
                    ?CFpower))))

(defrule sg-uvstop-bfr-stop-done-1
  ""
  (declare (salience ?*right-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&:mightbe-valid))
  (join
    (right-context ?segment stop)
    (CF (right-context ?segment stop) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvstop-bfr-stop-done-1))
  (prop ?segment (following-closure-start ?closure-start))
  =>
  (assert (applied ?segment sg-uvstop-bfr-stop-done-1))
  (hypothesize
    (assert (end-time ?segment ?closure-start))
    (assert (CF (end-time ?segment ?closure-start)
                =(CFcomb (CFweight ?*no-evidence* ?CFcategory)
                        (CFweight ?*definite-evidence* ?CFcontext))))))

```

-----



```

(defrule sg-uvstop-bfr-fricative-1
  "right context is unvoiced-vowel and fricative(/s/,/sh/)"
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment fricative)
    (CF (right-context ?segment fricative) ?CFcontext&: mightbe-valid))
  ;;(has-burst ?segment no)
  (not (applied ?segment sg-uvstop-bfr-fricative-1))
  ;;(prop ?segment (candidate-loc ? ?cand-to))
  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
  (power-start ?vowel-region&~NONE after ?cl-to 300 0 500 ?*voicing-power-min*)
  (power-increase (?vowel-start&~NONE ?vs-change) before =(+ ?vowel-region 30) 100
    0 500 ?*normal-smoothing-size* ?*voicing-power-min*)
  (power-increase (?frication-start&~NONE ?fs-change) before ?vowel-start 300 4000 6000
    ?*normal-smoothing-size* ?*frication-start-min-change-after-unvoiced-stop*)
  (power-decrease (?x&:(eq ?x 'NONE) ?c)
    before =(- ?vowel-start 20) =(- ?vowel-start 20 ?frication-start)
    4000 6000 ?*normal-smoothing-size* ?*closure-start-min-change*)
  =>
  (assert (applied ?segment sg-uvstop-bfr-fricative-1)
    (prop ?segment (frication-start ?frication-start))
    (prop ?segment (next-vowel-start ?vowel-start))))

(defrule sg-uvstop-bfr-fricative-2
  ""
  ;;
  ;; [hatazaki 143] Removed checking that ?CFaspiration and ?CFfricative-duration are larger
  ;; than ?*mightbe-valid*, since these values should effect to CF of
  ;; (right-context ?segment fricative) even if they are negatives.
  ;;
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment fricative)
    ?x <- (CF (right-context ?segment fricative) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvstop-bfr-fricative-2))
  (join
    (has-burst ?segment yes)
    (prop ?segment (burst ?bu-from ?bu-to ?bu-freq)))
  (prop ?segment (frication-start ?frication-start))
  (prop ?segment (next-vowel-start ?vowel-start))
  (CF (unvoiced-stop aspiration-duration-before-fricative =(- ?frication-start ?bu-from)
    ?CFaspiration)
    (CF (unvoiced-stop frication-duration-when-followed-by-fricative
      =(- ?vowel-start ?frication-start)
      ?CFfricative-duration)
    =>
  (assert (applied ?segment sg-uvstop-bfr-fricative-2))
  (retract ?x)
  ;;
  ;; [hatazaki 178] Make CF value of this context smaller, to select this hypothesis only if
  ;; no other context survives.
  ;;
  ;;      =(CFand (CFcomb ?CFcontext
  ;;                (CFweight ?*weak-evidence* ?CFaspiration)
  ;;                (CFweight ?*weak-evidence* ?CFfricative-duration))
  ;;      ?CFfricative-duration)
  ;;
  ;;      <-- =(CFand (CFcomb ?CFcontext
  ;;                (CFweight ?*evidence* ?CFaspiration)
  ;;                (CFweight ?*evidence* ?CFfricative-duration))
  ;;      ?CFfricative-duration)
  ;;
  ;; [hatazaki 143] (CFweight ?*evidence* ?CFaspiration) -->
  ;;      (CFweight ?*weak-evidence* ?CFaspiration),
  ;;      since duration of ASPIRATION is reasonable even if
  ;;      fricative does not follow.
  ;; [hatazaki 142] Changed new CF of (right-context ?segment fricative),
  ;;      =(CFcomb ?CFcontext (CFand ?CFaspiration ?CFfricative-duration)) -->
  ;;      =(CFcomb ?CFcontext

```

```

;;          (CFweight ?*evidence* ?CFaspiration)
;;          (CFweight ?*evidence* ?CFfricative-duration))
;;
(assert (CF (right-context ?segment fricative)
           =(CFand (CFcomb ?CFcontext
                     (CFweight ?*weak-evidence* ?CFaspiration)
                     (CFweight ?*weak-evidence* ?CFfricative-duration))
                 ?CFfricative-duration))))

(defrule sg-uvstop-bfr-fricative-3
  ""
  ;;
  ;; [hatazaki 143] Removed checking that ?CFaspiration and ?CFfricative-duration are larger
  ;; than ?*mightbe-valid*, since these values should effect to CF of
  ;; (right-context ?segment fricative) even if they are negatives.
  ;;
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
   (category ?segment unvoiced-stop)
   (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
   (right-context ?segment fricative)
   ?x <- (CF (right-context ?segment fricative) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvstop-bfr-fricative-3))
  (has-burst ?segment no)
  ;;(prop ?segment (candidate-loc ? ?cand-to))
  (prop ?segment (frication-start ?frication-start))
  (prop ?segment (next-vowel-start ?vowel-start))
  (prop ?segment (closure ?cl-from ?cl-to ?CFclosure))
  ;;(power-increase (?closure-end&~NONE ?change) after =(- ?cand-to 30) 100 4000 6000
  ;; ?*normal-smoothing-size* ?*closure-end-min-change*)
  (CF (unvoiced-stop aspiration-duration-before-fricative =(- ?frication-start ?cl-to))
      ?CFaspiration)
  (CF (unvoiced-stop frication-duration-when-followed-by-fricative
                  =(- ?vowel-start ?frication-start))
      ?CFfricative-duration)
  =>
  (assert (applied ?segment sg-uvstop-bfr-fricative-3))
  (retract ?x)
  ;;
  ;; [hatazaki 178] Make CF value of this context smaller, to select this hypothesis only if
  ;; no other context survives.
  ;;
  ;;          =(CFand (CFcomb ?CFcontext
  ;;                    (CFweight ?*weak-evidence* ?CFaspiration)
  ;;                    (CFweight ?*weak-evidence* ?CFfricative-duration))
  ;;              ?CFfricative-duration)
  ;;
  ;;          <-- =(CFand (CFcomb ?CFcontext
  ;;                       (CFweight ?*evidence* ?CFaspiration)
  ;;                       (CFweight ?*evidence* ?CFfricative-duration))
  ;;                ?CFfricative-duration)
  ;;
  ;; [hatazaki 143] (CFweight ?*evidence* ?CFaspiration) -->
  ;; (CFweight ?*weak-evidence* ?CFaspiration),
  ;; since duration of ASPIRATION is reasonable even if
  ;; fricative does not follow.
  ;; [hatazaki 142] Changed new CF of (right-context ?segment fricative),
  ;; =(CFcomb ?CFcontext (CFand ?CFaspiration ?CFfricative-duration))
  ;; =(CFcomb ?CFcontext
  ;;      (CFweight ?*evidence* ?CFaspiration)
  ;;      (CFweight ?*evidence* ?CFfricative-duration))
  ;;
  (assert (CF (right-context ?segment fricative)
             =(CFand (CFcomb ?CFcontext
                       (CFweight ?*weak-evidence* ?CFaspiration)
                       (CFweight ?*weak-evidence* ?CFfricative-duration))
                 ?CFfricative-duration))))

(defrule sg-uvstop-bfr-fricative-4
  "Checks there is no voicing power before the fricative."
  ;;
  ;; [hatazaki 158] Newly defined.
  ;;
  (declare (saliency ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
   (category ?segment unvoiced-stop)

```

```

(CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
(join
  (right-context ?segment fricative)
  ?x <- (CF (right-context ?segment fricative) ?CFcontext&: mightbe-valid))
(not (applied ?segment sg-uvstop-bfr-fricative-4))
(prop ?segment (frication-start ?frication-start))
(power-strength =(- ?frication-start 30) =(- ?frication-start 10) 0 500 ?0-500-power)
(CF (unvoiced-stop 0-500-power-before-vowel ?0-500-power) ?CFpower)
=>
(assert (applied ?segment sg-uvstop-bfr-fricative-4))
(retract ?x)
;;[hatazaki 178] (CFweight ?*evidence* ?CFpower)
;;              <-- (CFweight ?*strong-evidence* ?CFpower)
(assert (CF (right-context ?segment fricative)
            =(CFand (CFcomb ?CFcontext
                       (CFweight ?*evidence* ?CFpower))
                  ?CFpower))))))

(defrule sg-uvstop-bfr-fricative-done-1
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment fricative)
    (CF (right-context ?segment fricative) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvstop-bfr-fricative-done-1))
  (prop ?segment (frication-start ?frication-start))
  =>
  (assert (applied ?segment sg-uvstop-bfr-fricative-done-1))
  (hypothesize
    (assert (end-time ?segment ?frication-start))
    (assert (CF (end-time ?segment ?frication-start)
                =(CFcomb (CFweight ?*no-evidence* ?CFcategory)
                        (CFweight ?*definite-evidence* ?CFcontext))))))

;(defrule sg-uvstop-bfr-fricative-1
;  "right context is unvoiced-vowel and fricative(/s/,/sh/)"
;  (declare (salience ?*left-segmentation*))
;  (segment-status ?segment segmentation)
;  (category ?segment unvoiced-stop)
;  (right-context ?segment fricative)
;  (has-burst ?segment no)
;  (CF (right-context ?segment fricative) ?CF&:(?CF >= 0))
;  (not (applied ?segment sg-uvstop-bfr-fricative-1))
;  (prop ?segment (candidate-loc ? ?cand-to))
;  (power-strength ?cand-to =(+ ?cand-to 30) 0 500
;    ?strength&:(?strength < ?*low-freq-vowel-power-upper* ))
;  (unvoiced-candidate ?uv-from ?uv-to)
;  (test (<= (abs (- ?uv-from ?cand-to)) 10.0))
;  (test (> (- ?uv-to ?uv-from) 40.0))
;  (power-strength =(+ ?uv-from 10) ?uv-to 4500 6000
;    ?strength-high&:(?strength-high >= ?*high-freq-burst-power-upper*))
;  ; Next phoneme is /sh/
;  ; (i. e.) MAU_B_0036
;  ;
;  (case ((power-increase (?start&~NONE ?change)
;    after =(+ ?uv-from 20) 100 2000 6000
;    ?*normal-smoothing-size* 2.0)
;  =>
;  (hypothesize
;    (assert (end-time ?segment ?start))
;    (assert (CF (end-time ?segment ?start) 0.8))
;    (assert (applied ?segment sg-uvstop-bfr-fricative-1))))
;  ; Next phoneme is /s/
;  ; (i. e.) MAU_B_0154
;  ;
;  ((power-decrease (?start&~NONE ?change)
;    after =(+ ?uv-from 20) 100 500 1000
;    ?*normal-smoothing-size* -1.0)
;  =>
;  (hypothesize

```

```
; (assert (end-time ?segment ?start))  
; (assert (CF (end-time ?segment ?start) 0.8))  
; (assert (applied ?segment sg-uvstop-bfr-fricative-1))))))
```

-----

```
;(defrule unvoiced-stop-right-done-1
;   "Searches the time location when the vowel energy goes down."
;   (declare (salience ?*right-segmentation-end*))
;   (segment-status ?segment segmentation)
;   (category ?segment unvoiced-stop)
;   (right-context ?segment vowel)
;   (not (applied ?segment unvoiced-stop-segmentation-before-vowel-done-1))
;   (end-time ?segment ?end)
;   =>
;   (assert (applied ?segment unvoiced-stop-segmentation-before-vowel-done-1))
;   (hypothesize
;     (assert (end-time ?segment ?end))
;     (assert (CF (end-time ?segment ?end) 0.8))))
;-----
```



LM17: >hatazaki>sre>seg-unvoiced-stop>seg-stop-end. art. 1

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 12/01/88 16:15:05

Queued at: 3/28/89 11:01:07

```
;;; -*- Mode: ART; Base: 10; Package: ART-USER -*-
```

```
(defrule sg-unstop-at-word-initial-end-1
  "Checks the duration."
  (declare (salience ?*before-end-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-stop)
    ?x <- (CF (category ?segment unvoiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (start-time ?segment ?start)
    (explicit (CF (start-time ?segment ?start) ?CFstart&: mightbe-valid)))
  (join
    (end-time ?segment ?end)
    (explicit (CF (end-time ?segment ?end) ?CFend&: mightbe-valid)))
  (join
    (left-context ?segment word-initial)
    (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-unstop-at-word-initial-end-1))
  (CF (unvoiced-stop duration-at-word-initial =(- ?end ?start)) ?CFduration)
  =>
  (retract ?x)
  (assert (CF (category ?segment unvoiced-stop)
    =(CFand ?CFcategory ?CFduration)))
  (assert (applied ?segment sg-unstop-at-word-initial-end-1)))
```



LM17: >hatazaki>sre>seg-unvoiced-fricative>seg-fricative. art. 29

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/01/89 15:56:57

Queued at: 3/28/89 10:47:39

```
;;- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*
```

```
(defrelation UNVOICED-FRICATIVE-CANDIDATE (?from ?to)
  implicit
  (syntax output
    (Unvoiced fricative candidate at ?from - ?to)))
```

```
(defrule sg-unvfric-0
  "Finds a candidate of unvoiced-fricative segment."
```

```
;;
;;[hatazaki 148] Checks (unvoiced-fricative 0-500-power ?0-500-P) to reject /z/.
;;
;;(declare (salience ?*find-candidate*))
;;(segmenting unvoiced-fricative)
;;(unvoiced-fricative-candidate ?from ?to&:(>= (- ?to ?from) 20))
;;
;; - Use CF value for determining this is unvoiced fricative. [hatazaki 7/22/88 15:55:21]
;; - Old code ...
;; (power-strength ?from ?to 4000 6000 ?strength&:(?strength > ?*high-freq-power-lower*))
;; (cutoff ?from ?to ?*cutoff-th* ?cutoff&:(?cutoff > ?*fricative-cutoff-lower*))
;;
;;(power-strength ?from ?to 4000 6000 ?fricative)
;;(CF (unvoiced-fricative 4000-6000-power ?fricative) ?CFfricative&:mightbe-valid)
;;(cutoff ?from ?to ?*cutoff-th* ?cutoff)
;;(CF (unvoiced-fricative cutoff ?cutoff) ?CFcutoff&:mightbe-valid)
;;(power-strength ?from ?to 0 500 ?0-500-P)
;;(CF (unvoiced-fricative 0-500-power ?0-500-P) ?CF0-500-P&:mightbe-valid)
;;(power-strength ?from ?to 1000 2000 ?1000-2000-P)
;;(CF (unvoiced-fricative 0-500-1000-2000-power-ratio =(- ?0-500-P ?1000-2000-P))
  ?CF-0-500-1000-2000-PR&:mightbe-valid)
;;
;;[hatazaki 146] Removed checks of 4000-6000-1000-2000-power-ratio, since some /sh/ has
;; power at 1000-2000Hz, then 4000-6000-1000-2000-power-ratio has no sense.
;; <--
;; (CF (unvoiced-fricative 4000-6000-1000-2000-power-ratio =(- ?fricative ?1000-2000-P))
  ?CF-4000-6000-1000-2000-PR&:(?CF-4000-6000-1000-2000-PR >= ?*mightbe-valid*))
=>
(bind ?segment (gensym "SEG-"))
(assert (schema ?segment
  (instance-of phoneme-segment)))
(hypothesize
  (assert (category ?segment unvoiced-fricative))
  ;; - UNVOICED-FRICATIVE includes /h/ (ex. MAU_B_0129)
  ;; [11/16/87 11:15:07].
  ;;
  ;;
  (assert (prop ?segment (high-freq-power-cutoff ?cutoff))
    (prop ?segment (candidate-loc ?from ?to))
    (segment-status ?segment segmentation))
  ;;
  ;;[hatazaki 178]
  ;; (CFweight ?*weak-evidence* ?CFfricative)
  ;; <-- (CFweight ?*evidence* ?CFfricative)
  ;;[hatazaki 167]
  ;; =(CFand (CFcomb (CFunknown)
  ;; (CFweight ?*weak-evidence* ?CFcutoff)
  ;; (CFweight ?*weak-evidence* ?CFfricative)
  ;; (CFweight ?*weak-evidence* ?CF0-500-P)
  ;; (CFweight ?*weak-evidence* ?CF-0-500-1000-2000-PR))
  ;; ?CFfricative
  ;; ?CF0-500-P
  ;; ?CF-0-500-1000-2000-PR)
  ;; <--
  ;; =(CFcomb (CFweight ?*weak-evidence* ?CFcutoff)
  ;; (CFweight ?*strong-evidence*
  ;; (CFand ?CFfricative ?CF0-500-P ?CF-0-500-1000-2000-PR
  ;; ;?CF-4000-6000-1000-2000-PR)))
  ;;[hatazaki 145]
  ;; =(CFcomb (CFweight ?*weak-evidence* ?CFcutoff)
  ;; (CFweight ?*evidence*
  ;; (CFand ?CFfricative ?CF-0-500-1000-2000-PR
  ;; ?CF-4000-6000-1000-2000-PR)))
  ;; <--
  ;; [hatazaki 144] New CF of (category ?segment unvoiced-fricative),
  ;; =(CFcomb (CFweight ?*evidence* ?CFfricative)
  ;; (CFweight ?*weak-evidence* ?CFcutoff)
  ;; (CFweight ?*evidence*
```

```

;;                                     (CFand ?CF-0-500-1000-2000-PR
;;                                     ?CF-4000-6000-1000-2000-PR)))
;; <--
;; =(CFand ?CFfrication ?CFcutoff ?CF-0-500-1000-2000-PR
;;       ?CF-4000-6000-1000-2000-PR)
;;
;; (CF (category ?segment unvoiced-fricative)
;;     =(CFand (CFcomb (CFunknown)
;;                   (CFweight ?*weak-evidence* ?CFcutoff)
;;                   (CFweight ?*weak-evidence* ?CFfrication)
;;                   (CFweight ?*weak-evidence* ?CF0-500-P)
;;                   (CFweight ?*weak-evidence* ?CF-0-500-1000-2000-PR))
;;           ?CFfrication
;;           ?CF0-500-P
;;           ?CF-0-500-1000-2000-PR))))))

(defrule sg-unvfri-1
  "Checks this is not stop."
  (declare (salience ?*start-segmentation*)
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    ?x <- (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (not (applied ?segment sg-unvfri-1))
  (prop ?segment (candidate-loc ?from ?to))
  ;;
  ;;[hatazaki 145] Closure is searched by get-closure-1 defined in seg-stop.art.
  ;;
  (closure-exists ?cl-from ?cl-to before ?from 20 ?CFclosure)
  =>
  (assert (applied ?segment sg-unvfri-1))
  (retract ?x)
  ;;
  ;;[hatazaki 167]
  ;;
  ;; =(CFand (CFcomb ?CFcategory
  ;;               (CFweight ?*weak-evidence* (- ?CFclosure)))
  ;;         (- ?CFclosure))
  ;; <--
  ;; =(CFand ?CFcategory
  ;;       (CFweight ?*definite-evidence* (- ?CFclosure)))
  ;;
  ;;
  (assert (CF (category ?segment unvoiced-fricative)
              =(CFand (CFcomb ?CFcategory
                            (CFweight ?*weak-evidence* (- ?CFclosure))
                            (- ?CFclosure))))))
  ;;
  ;;-----
  ;;

```

```
(defrule sg-uvfric-bottom-up
  ""
  (declare (salience ?*top-down*))
  (goal (UNVOICED-FRICATIVE-CANDIDATE ?from&?? ?to&??))
  (not (explicit (UNVOICED-FRICATIVE-CANDIDATE ? ?)))
  (BB ?BB)
  =>
  (at ?BB
    (for REGION in (intersected-regions
      (list
        (caddr (#USER::call-speechfunc (list "POWER-ISLAND-IN-TIME"
          0 2000
          5000 6000
          -40 0
          11)))
        (caddr (#USER::call-speechfunc (list "POWER-ISLAND-IN-TIME"
          0 2000
          0 500
          -99 -45
          11))))))
    do (assert (UNVOICED-FRICATIVE-CANDIDATE =(car REGION) =(cadr REGION))))))
;-----
```

```

(defrule sg-uvfric-more
  "Do segmentation for another segment in this unvoiced period."
  (declare (salience ?*start-segmentation*))
  (viewpoint ?v1
    (join
      (category ?segment unvoiced-fricative)
      (segment-status ?segment segmentation-ended))
    (join
      (end-time ?segment ?end)
      (CF (end-time ?segment ?end) ?CFend&:is-valid))
      (unvoiced-fricative-candidate ?from&:(?from < ?end) ?to&:(>= (- ?to ?end) 20)))
    (power-strength ?end ?to 4000 6000 ?frication)
    (CF (unvoiced-fricative 4000-6000-power ?frication) ?CFfrication&:mightbe-valid)
    (cutoff ?end ?to ?*cutoff-th* ?cutoff)
    (CF (unvoiced-fricative cutoff ?cutoff) ?CFcutoff&:mightbe-valid)
    (power-strength ?end ?to 0 500 ?0-500-P)
    (power-strength ?end ?to 1000 2000 ?1000-2000-P)
    (CF (unvoiced-fricative 0-500-1000-2000-power-ratio =(- ?0-500-P ?1000-2000-P))
      ?CF-0-500-1000-2000-PR&:mightbe-valid)
    ;;
    ;;[hatazaki 146] Removed checks of 4000-6000-1000-2000-power-ratio, since some /sh/ has
    ;; power at 1000-2000Hz, then 4000-6000-1000-2000-power-ratio has no sense.
    ;; <--
    ;; (CF (unvoiced-fricative 4000-6000-1000-2000-power-ratio =(- ?frication ?1000-2000-P))
    ;;   ?CF-4000-6000-1000-2000-PR&:(?CF-4000-6000-1000-2000-PR >= ?*mightbe-valid*))
    =>
    (bind ?new-segment (gensym "SEG-"))
    (assert (schema ?new-segment
      (instance-of phoneme-segment)))
    (at ?v1
      (hypothesize
        (assert (category ?new-segment unvoiced-fricative))
        (assert (prop ?new-segment (high-freq-power-cutoff ?cutoff))
          (prop ?new-segment (candidate-loc ?end ?to))
          (left-context ?new-segment unvoiced-fricative)
          (start-time ?new-segment ?end)
          (CF (start-time ?new-segment ?end) ?CFend)
          (segment-status ?new-segment segmentation)
          ;;
          ;;[hatazaki 167]
          ;; =(CFand (CFcomb (CFunknown)
          ;;   (CFweight ?*weak-evidence* ?CFcutoff)
          ;;   (CFweight ?*evidence* ?CFfrication)
          ;;   (CFweight ?*weak-evidence* ?CF-0-500-1000-2000-PR))
          ;;   ?CFfrication
          ;;   ?CF-0-500-1000-2000-PR)
          ;; =(CFcomb (CFweight ?*evidence* ?CFfrication)
          ;;   (CFweight ?*weak-evidence* ?CFcutoff)
          ;;   (CFweight ?*evidence*
          ;;     (CFand ?CF-0-500-1000-2000-PR
          ;;       ;;?CF-4000-6000-1000-2000-PR
          ;;     )))
          ;;
          (CF (category ?new-segment unvoiced-fricative)
            =(CFand (CFcomb (CFunknown)
              (CFweight ?*weak-evidence* ?CFcutoff)
              (CFweight ?*evidence* ?CFfrication)
              (CFweight ?*weak-evidence* ?CF-0-500-1000-2000-PR))
              ?CFfrication
              ?CF-0-500-1000-2000-PR))))))

```

LM17: >hatazaki >sre>seg-unvoiced-fricative>seg-fricative-left. art. 48

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/24/89 15:44:28

Queued at: 3/28/89 10:48:28

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
;;;
;;;[hatazaki 144]
;;; - (left-context ?segment word-initial) <-- (left-context ?segment silence).
;;;
(defrule sg-uvfric-left-1
  "Hypothesizes left phoneme."
  (declare (salience ?*left-segmentation*))
  ;(declare (salience ?*start-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (not (start-time ?segment ?))
  (not (applied ?segment sg-uvfric-left-1))
  =>
  (assert (applied ?segment sg-uvfric-left-1))
  (conflicting-alternatives
    (hypothesize
      ;;
      ;;- This includes syllabic nasal which has middle frequency power.
      ;;
      (assert (left-context ?segment vowel))
      (assert (CF (left-context ?segment vowel) =(CFunknown))))
    (hypothesize
      (assert (left-context ?segment word-initial))
      (assert (CF (left-context ?segment word-initial) =(CFunknown))))
    (hypothesize
      ;; - For the case this phoneme is preceeded by stop with devoiced vowel.
      ;; cf. MAU_B_0154 [hatazaki 3/31/88 09:44:55]
      (assert (left-context ?segment stop))
      (assert (CF (left-context ?segment stop) =(CFunknown))))
    ;;
    ;;[hatazaki 144] Removed the following hypothesis, since affricates are treated as stops.
    ;; (hypothesize
    ;;   (assert (left-context ?segment closure))
    ;;   (assert (CF (left-context ?segment closure) 0.0)))
    ;;
    ;;-
    ;; (hypothesize
    ;;   ;;- This is for syllabic nasal which has no middle frequency power.
    ;;   (assert (left-context ?segment syllabic-nasal))
    ;;   (assert (CF (left-context ?segment syllabic-nasal) 0.0)))
    ))
  )
```

```
-----
```

```

;(defrule sg-uvfric-aft-vowel-1
;   "There should not be a closure before this segment."
;   ;; - This rule should be applied before the other segmentation rules below are applied.
;   (declare (salience (+ ?*left-segmentation* 1)))
;   (segment-status ?segment segmentation)
;   (join
;     (category ?segment unvoiced-fricative)
;     (CF (category ?segment unvoiced-fricative) ?CFcategory&:(?CFcategory >= ?*mightbe-valid*)))
;   (join
;     (left-context ?segment vowel)
;     ?x <- (CF (left-context ?segment vowel) ?CFcontext&:(?CFcontext >= ?*mightbe-valid*)))
;   (not (applied ?segment sg-uvfric-aft-vowel-1))
;   (closure-exists ?from ?to left-of ?segment ?closure-CF&:(?closure-CF >= ?*mightbe-valid*))
;   =>
;   (assert (applied ?segment sg-uvfric-aft-vowel-1))
;   (retract ?x)
;   ;;
;   ;;[hatazaki 144] New CF of (left-context ?segment vowel),
;   ;;
;   ;;      =(CFcomb ?CFcontext
;   ;;        (CFweight ?*strong-evidence* (- ?closure-CF)))
;   ;;
;   ;;      <--
;   ;;      =( #USER::comb-CF ?CF -0.9)
;   ;;
;   (assert (CF (left-context ?segment vowel)
;               =(CFcomb ?CFcontext
;                   (CFweight ?*strong-evidence* (- ?closure-CF)))))
)

(defrule sg-uvfric-aft-vowel-0
  "Get the spectrum peak."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment vowel)
    (CF (left-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-aft-vowel-0))
  (prop ?segment (candidate-loc ?from ?to))
  (spectrum-global-peak (?freq ?peak-power) ?from =(+ ?from 20)
    ?*spectrum-global-peak-moving-average-size*)
  =>
  (assert (prop ?segment (spectrum-global-peak (?freq ?peak-power))))
  (assert (applied ?segment sg-uvfric-aft-vowel-0)))

(defrule sg-uvfric-aft-vowel-2
  "find the high-freq-power start time."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment vowel)
    (CF (left-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-aft-vowel-2))
  (prop ?segment (candidate-loc ?from ?to))
  ;;
  ;;- If a spectrum has global peak around 3000-4000Hz.
  ;; The left boundary should be determined by spectral power change around the spectrum
  ;; peak. This is mainly for /h/.(cf. MAU_B_0200) [hatazaki 6/16/88 18:55:10]
  ;;
  ;;- Closure start time is searched using 4000-6000Hz power. [hatazaki 4/26/88 16:44:48]
  ;;- Closure start time is searched using 5000-6000Hz power, instead ?cutoff-6000Hz power.
  ;;                                     [hatazaki 4/19/88 00:50:59]
  ;;- Searches up to 150msec before the start time. (cf. MAU_B_0034)
  ;;
  (or (and (prop ?segment (spectrum-global-peak (?peak-freq&~NONE ?peak-power)))
    (power-increase (?hfe-start&~NONE ?change)
      before =(+ ?from 20) 150 =(- ?peak-freq 500) =(+ ?peak-freq 500)
      ?*normal-smoothing-size*
      ?*frication-start-min-change*))
    (and (prop ?segment (spectrum-global-peak (NONE ?))))

```



```

        (power-increase (?hfe-start&~NONE ?change)
          before =(+ ?from 20) 150 4000 6000
          ?*normal-smoothing-size*
          ?*fricative-start-min-change*))
;; (power-strength =(- ?hfe-start 20) ?hfe-start 1000 6000
;;   ?power&:(?power >= ?*vowel-mid-power-lower*))
=>
(assert (prop ?segment (high-freq-power-start ?hfe-start)))
(assert (applied ?segment sg-uvfric-aft-vowel-2)))

(defrule sg-uvfric-aft-vowel-3
  "Find the time the preceding voicing disappears."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment vowel)
    (CF (left-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-aft-vowel-3))
  (prop ?segment (candidate-loc ?from ?))
  ;;
  ;;- Changed the frequency range for searching vowel to 1000-2500Hz instead of 1500-2500Hz.
  ;; [hatazaki 5/24/88 18:17:45]
  ;;- Searched the time the vowel ends using 1500-2500Hz frequency spectral change rather than
  ;; 1000-2000Hz, and use ?*vowel-F2-end-min-change* instead of ?*default-min-change*.
  ;; [hatazaki 5/24/88 13:45:54]
  ;;- Searches the time when the preceding vowel ends.
  ;; Various methods were tried.
  ;; (1) (power-end ?here&~NONE after =(- ?from 30) 60 500 1000 0.65)
  ;; (2) (power-decrease ?here&~NONE after =(- ?from 30) 60 500 1000)
  ;; (3) (power-decrease ?here&~NONE before =(+ ?from 20) 100 1000 1500)
  ;; (4) (power-decrease ?here&~NONE before =(+ ?from 20) 100 1000 2000)
  ;; - Spectrum change can be used, since the power goes down suddenly.
  ;; - (2) is not good, since the voicing of the preceding vowel may remains in the fricative
  ;;   area between 0 - 1000 Hz. (c.f. MAU_1_1157)
  ;; (3) is not good, since the frequency range is too small, then the result is
  ;;   not stable.
  ;;
  (power-decrease (?here&~NONE ?change) before =(+ ?from 20) 100 1000 2500
    ?*normal-smoothing-size* ?*vowel-F2-end-min-change*)
  ;;(test (<= (abs (- ?from ?here)) 50.0))
  =>
  (assert (prop ?segment (preceding-vowel-end-time ?here)))
  (assert (applied ?segment sg-uvfric-aft-vowel-3)))

(defrule sg-uvfric-aft-vowel-3b
  "Find the time the preceding voicing disappears using smaller threshold."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment vowel)
    (CF (left-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-aft-vowel-3b))
  (prop ?segment (candidate-loc ?from ?))
  (power-decrease (NONE ?) before =(+ ?from 20) 100 1000 2500
    ?*normal-smoothing-size* ?*vowel-F2-end-min-change*)
  (power-decrease (?here&~NONE ?change) before =(+ ?from 20) 100 1000 2500
    ?*normal-smoothing-size* ?*vowel-F2-end-min-change-smaller*)
  =>
  (assert (prop ?segment (preceding-vowel-end-time ?here)))
  (assert (applied ?segment sg-uvfric-aft-vowel-3b)))

(defrule sg-uvfric-aft-vowel-4
  "Checks vowel power."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment vowel)

```

```

?x <- (CF (left-context ?segment vowel) ?CFcontext&: mightbe-valid)
(not (applied ?segment sg-uvfric-aft-vowel-4))
(prop ?segment (preceding-vowel-end-time ?vowel-end))
;;
;;[hatazaki 144]
;; <--
;; (power-strength =(- ?vowel-end 30) ?vowel-end 0 500
;;   ?strength1&:(?strength1 >= ?*voicing-power-min-before-fricative*))
;;
;;- Use threshold ?*voicing-power-min-before-fricative* instead of ?*voicing-power-min*,
;; since ?*voicing-power-min* is slightly small for accept the vowel at word end.
;; [hatazaki 5/24/88 13:51:44]
;;- The following checks of vowel existence failed for MAU_1_1157, in which
;; another fricative and an imcomplete closure preceds to this fricative.
;; The frequency range 1000-3000 includes the
;; frication power of the preceding fricative, and its power exceeds
;; ?*no-power-upper*, although this is not an vowel. [hatazaki 1/08/88 15:28:02]
;; (power-strength =(- ?vowel-end 30) ?vowel-end 1000 3000
;;   ?strength1&:(?strength1 >= ?*no-power-upper*))
;;
;; (power-strength =(- ?vowel-end 30) ?vowel-end 0 500 ?vowel-0-500-power)
;; (CF (vowel 0-500-power ?vowel-0-500-power) ?CF0-500-P)
;; (power-strength =(- ?vowel-end 30) ?vowel-end 0 6000 ?vowel-0-6000-power)
;; (CF (vowel 0-6000-power ?vowel-0-6000-power) ?CF0-6000-P)
;;
;; - The cutoff value stored in (high-freq-power ? ? ?cutoff) is not
;; necessarily correct, since this value is obtained by assuming the
;; whole unvoiced period is the fricative segment and the cutoff is
;; horizontal (cf. MAU_1_1157). [hatazaki 1/08/88 16:10:40]
;; (prop ?segment
;;   (high-freq-power ? ? ?cutoff))
;; (power-strength ?vowel-end =(+ ?vowel-end 50) 1000 ?cutoff
;;   ?strength2&:(?strength2 < ?*fricative-middle-power-upper*))
;;
;;(prop ?segment (high-freq-power-cutoff ?cutoff))
;;(power-strength ?vowel-end =(+ ?vowel-end 50) 1000 =(min ?cutoff 1500)
;;   ?strength2&:(?strength2 < ?*fricative-middle-power-upper*))
=>
(retract ?x)
;;
;;[hatazaki 167]
;; =(CFand (CFcomb ?CFcontext
;;   (CFweight ?*evidence* ?CF0-500-P)
;;   (CFweight ?*evidence* ?CF0-6000-P))
;;   ?CF0-500-P
;;   ?CF0-6000-P)
;; <-- =(CFcomb ?CFcontext
;;   (CFweight ?*strong-evidence*
;;     (CFand ?CF0-500-P ?CF0-6000-P)))
;;
(assert (CF (left-context ?segment vowel)
  =(CFand (CFcomb ?CFcontext
    (CFweight ?*evidence* ?CF0-500-P)
    (CFweight ?*evidence* ?CF0-6000-P))
    ?CF0-500-P
    ?CF0-6000-P)))
(assert (applied ?segment sg-uvfric-aft-vowel-4)))

(defrule sg-uvfric-aft-vowel-5
  "Checks vowel-offset-time and vowel-offset-1000-6000-power"
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment vowel)
    ?x <- (CF (left-context ?segment vowel) ?CFcontext&: mightbe-valid)
    (not (applied ?segment sg-uvfric-aft-vowel-5))
    (prop ?segment (preceding-vowel-end-time ?vowel-end))
    (prop ?segment (high-freq-power-start ?hfe-start&:(?vowel-end < ?hfe-start)))
    (CF (unvoiced-fricative vowel-offset-time =(- ?hfe-start ?vowel-end)) ?CFvoff-time)
    (power-strength ?vowel-end ?hfe-start 1000 6000 ?voff-power)
    (CF (unvoiced-fricative vowel-offset-1000-6000-power ?voff-power) ?CFvoff-power)
    =>
    (retract ?x)

```

```

;;
;;[hatazaki 167]
;;      =(CFand (CFcomb ?CFcontext
;;              (CFweight ?*weak-evidence* ?CFvoff-time)
;;              (CFweight ?*weak-evidence* ?CFvoff-power))
;;          ?CFvoff-time
;;          ?CFvoff-power)
;;  <--  =(CFcomb ?CFcontext
;;        (CFweight ?*evidence* ?CFvoff-time)
;;        (CFweight ?*evidence* ?CFvoff-power))
;;
(assert (CF (left-context ?segment vowel)
           =(CFand (CFcomb ?CFcontext
                     (CFweight ?*weak-evidence* ?CFvoff-time)
                     (CFweight ?*weak-evidence* ?CFvoff-power))
                 ?CFvoff-time
                 ?CFvoff-power)))
(assert (applied ?segment sg-uvfric-aft-vowel-5)))

(defrule sg-uvfric-aft-vowel-done-1
  ""
  ;;
  ;; - This rule is for the case when the high frequency power is strong, then
  ;; the boundary can be detected by spectrum-change (see MAU_B_0033).
  ;; - This segment might be /sh/.
  ;;
  ;; - Accept ?hfe-start iff this is larger than ?vowel-end.
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
   (category ?segment unvoiced-fricative)
   (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (join
   (left-context ?segment vowel)
   (CF (left-context ?segment vowel) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvfric-aft-vowel-done-1))
  (prop ?segment (preceding-vowel-end-time ?vowel-end))
  ;;
  ;; - Accept ?hfe-start as start time if (and (<= (- ?hfe-start ?vowel-end) 50)
  ;;                                           (<= (- ?vowel-end ?hfe-start) 20)) is T,
  ;; instead (test (<= (abs (- ?hfe-start ?vowel-end)) 50)) is T.
  ;; [hatazaki 4/25/88 11:28:34]
  ;; - When hfe-start is found, makes it this segment start time even if hfe-start < pv-end.
  ;; Checks only the fact hfe-start and pv-end is not separated so far.
  ;; Next statement is commented. [hatazaki 3/15/88 21:00:52]
  ;;(prop ?segment (high-freq-power-start ?hfe-start&:(?vowel-end <= ?hfe-start)))
  (prop ?segment
   (high-freq-power-start ?hfe-start&(and (<= (- ?hfe-start ?vowel-end) 50)
                                           (<= (- ?vowel-end ?hfe-start) 20))))
  =>
  (assert (applied ?segment sg-uvfric-aft-vowel-done-1))
  (hypothesize
   (assert (start-time ?segment ?hfe-start))
   (assert (CF (start-time ?segment ?hfe-start) =(CFvalue ?CFcontext))))))

(defrule sg-uvfric-aft-vowel-done-2
  ""
  ;;
  ;; - This rule is for the case when the high frequency power is weak and then
  ;; the high frequency power start tie cannot be found by the spectrum change
  ;; (see MAU_B_0030).
  ;; - It might be said that the segment is /s/.
  ;;
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
   (category ?segment unvoiced-fricative)
   (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (join
   (left-context ?segment vowel)
   (CF (left-context ?segment vowel) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvfric-aft-vowel-done-2))
  (prop ?segment (preceding-vowel-end-time ?vowel-end))
  ;;
  ;; - Accept ?hfe-start as start time if (and (<= (- ?hfe-start ?vowel-end) 50)
  ;;                                           (<= (- ?vowel-end ?hfe-start) 20)) is T,
  ;;

```

```
;; instead (test (<= (abs (- ?hfe-start ?vowel-end)) 50)) is T.
;; [hatazaki 4/25/88 11:28:34]
;; - Accept the case when ?hfe-start is obtained but is far from ?vowel-end.
;; (cf. MAU_B_0089 /s/) [hatazaki 4/22/88 18:09:34]
;; - Condition (?vowel-end <= ?hfe-start) is removed due to the change in
;; sg-uvfric-aft-vowel-done-1.
;; Originally,
;; (not (prop ?segment (high-freq-power-start ?hfe-start& (?vowel-end <= ?hfe-start))))
(not
  (prop ?segment
    (high-freq-power-start ?hfe-start&(and (<= (- ?hfe-start ?vowel-end) 50)
      (<= (- ?vowel-end ?hfe-start) 20))))))
=>
(assert (applied ?segment sg-uvfric-aft-vowel-done-2))
(hypothesize
  (assert (start-time ?segment ?vowel-end))
  (assert (CF (start-time ?segment ?vowel-end) =(CFvalue ?CFcontext))))))
```

-----

```

(defrule sg-uvfric-at-word-initial-0
  "There is no voicing power."
  ;;
  ;;[hatazaki 147] Newly defined to reject cases when an vowel precedes to this fricative.
  ;; (c.f. MAU_B_0066)
  ;;
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)
    ?x <- (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-at-word-initial-0))
  (prop ?segment (candidate-loc ?from ?to))
  ;;
  ;; - Checks there is no voicing power. [hatazaki 5/24/88 17:58:09]
  ;; Old code is ...
  ;; (power-decrease (?z&:(eq ?z 'NONE) ?)
  ;; before =(+ ?from 20) 100 1500 2500
  ;; ?*normal-smoothing-size* ?*vowel-F2-end-min-change-smaller*)
  ;;
  ;; - Frequency range was changed to 1500-2500 from 1000-2500..
  ;; - Frequency range was changed to 1500-2500 from 1000-2000. [hatazaki 5/24/88 14:04:02]
  ;; - Checks that there is no vowel before this segment. [hatazaki 4/28/88 11:59:10]
  (power-decrease (?vowel-end&~NONE ?)
    before =(+ ?from 20) 100 1000 2500
    ?*normal-smoothing-size* ?*vowel-F2-end-min-change-smaller*)
  (power-strength =(- ?vowel-end 30) ?vowel-end 0 500 ?0-500-P)
  (CF (vowel 0-500-power ?0-500-P) ?CFvowel)
  =>
  (retract ?x)
  ;;
  ;;[hatazaki 167]
  ;; = (CFand (CFcomb ?CFcontext
  ;; (CFweight ?*evidence* (- ?CFvowel)))
  ;; (- ?CFvowel))
  ;; <-- = (CFand ?CFcontext
  ;; (CFweight ?*strong-evidence* (- ?CFvowel)))
  ;;
  (assert (CF (left-context ?segment word-initial)
    = (CFand (CFcomb ?CFcontext
      (CFweight ?*evidence* (- ?CFvowel)))
      (- ?CFvowel))))
  (assert (applied ?segment sg-uvfric-at-word-initial-0)))

(defrule sg-uvfric-at-word-initial-1
  "Searches time the silence ends."
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)
    (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-at-word-initial-1))
  (prop ?segment (candidate-loc ?from ?to))
  (power-start ?silence-end&~NONE before =(+ ?from 20) 100 0 6000 ?*silence-power-upper*)
  =>
  (assert (prop ?segment (silence-end ?silence-end)))
  (assert (applied ?segment sg-uvfric-at-word-initial-1)))

(defrule sg-uvfric-at-word-initial-2
  "Get the spectrum peak."
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)

```

```

    (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-at-word-initial-2))
  (prop ?segment (candidate-loc ?from ?to))
  (spectrum-global-peak (?freq ?peak-power) ?from =(+ ?from 20)
    ?*spectrum-global-peak-moving-average-size*)
=>
  (assert (prop ?segment (spectrum-global-peak (?freq ?peak-power))))
  (assert (applied ?segment sg-uvfric-at-word-initial-2)))

(defrule sg-uvfric-at-word-initial-3
  "Searches time the frication starts."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)
    (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-at-word-initial-3))
  (prop ?segment (silence-end ?silence-end))
  ;;
  ;;[hatazaki 146] ?frication-start is searched after ?silence-end.
  ;;  after =(- ?silence-end 20) 50
  ;;  <-- before =(+ ?from 20) 150
  ;; - Searches ?frication-start using the spectral power change around ?peak-freq
  ;; if it is not NONE. This is for /h/. (cf. MAU_B_0036 /h/, MAU_B_0129 /h/).
  ;; [hatazaki 6/20/88 17:38:24]
  ;;
  ;; - Searches ?frication-start by 4000-6000Hz power, instead 0-6000Hz power.
  ;; [hatazaki 4/28/88 11:58:44]
  ;; - Searches ?frication-start by 0-6000Hz power, instead 5000-6000Hz power.
  ;; [hatazaki 4/26/88 16:47:57]
  ;; - Searches up to 150msec before the start time. (cf. MAU_B_0034 /is/)
  (or (and (prop ?segment (spectrum-global-peak (?peak-freq&~NONE ?peak-power)))
    (power-increase (?frication-start&~NONE ?change)
      after =(- ?silence-end 20) 50
      =(- ?peak-freq 500) =(+ ?peak-freq 500)
      ?*normal-smoothing-size* ?*frication-start-min-change*))
    (and (prop ?segment (spectrum-global-peak (NONE ?)))
      (power-increase (?frication-start&~NONE ?change)
        after =(- ?silence-end 20) 50
        4000 6000
        ?*normal-smoothing-size* ?*frication-start-min-change*)))
  =>
  (assert (prop ?segment (frication-start ?frication-start)))
  (assert (applied ?segment sg-uvfric-at-word-initial-3)))

(defrule sg-uvfric-at-word-initial-4
  "There is no power before this segment."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)
    ?x <- (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-at-word-initial-4))
  (prop ?segment (frication-start ?frication-start))
  ;;
  ;;[hatazaki 145]
  ;; (case
  ;; ((and (power-start ?word-start&~NONE
  ;; before =(+ ?frication-start 30) 60 0 6000 ?*silence-power-upper*)
  ;; (power-end ?y&(eq ?y 'NONE) before ?word-start 500 0 6000 ?*silence-power-upper*))
  ;; =>
  ;; (bind ?CFno-power-before 0.8))
  ;; (otherwise
  ;; =>
  ;; (bind ?CFno-power-before -0.8)))
  ;; <--
  ;; (power-strength =(- ?frication-start 20) ?frication-start 0 6000
  ;; ?silence-power&:(?silence-power <= ?*silence-power-upper*))
  ;; (power-end ?y&(eq ?y 'NONE)
  ;; before ?frication-start 500 0 6000 ?*silence-power-upper*)

```

```

;;
;;(power-start ?frication-start&~NONE
;;   before =(+ ?from 20) 4000 6000 ?*silence-power-upper*)
;;(test (< (- ?from ?frication-start) 50.0))
;;
;; - Checks that the power strength here is less than ?*silence-power-upper*,
;;   so that next power-end function works correctly. If the power strength at
;;   the start point of power-end function is already more than
;;   ?*silence-power-upper*, this function cannot detect the time when
;;   the power is more than ?*silence-power-upper*. [12/02/87 12:02:21]
;;   (power-strength =(- ?frication-start 20) ?frication-start 0 6000
;;     ?silence-power&:(?silence-power <= ?*silence-power-upper*))
;;
;;
;; - Checking that there is no power-end point should be done by NOT operation.
;; If next pattern is used,
;;   (power-end ?silence-start&~NONE
;;     before ?frication-start 500 0 6000 ?*silence-power-upper*)) ... (1)
;; a goal like
;;   (goal (power-end NONE before ...))
;; is generated and the top-down rule cannot be fired, since the second argument
;; NONE is not matched with ?? in the goal pattern below.
;;   (goal (power-end ?? before ?start-time&~?? ?time-range&~??
;;     ?bottom-freq&~?? ?top-freq&~?? ?threshold&~??))
;;   [12/03/87 14:44:42 hatazaki][12/04/87 04:03:12 hatazaki]
;;
;; - When we use the above pattern (1), we must define the backward
;; chaining rule as following.
;;   (or (goal (power-end ?? before ...))
;;     (goal (power-end NONE before ...)))
;; This backward-chaining rule works correctly, but it is divided into two
;; rules, each of which matches each element in OR close, by rule compiler.
;; [12/07/87 10:09:36 hatazaki]
;;
;; - When NOT operation is used as following,
;;   (not (power-end ?silence-start&~NONE
;;     before ?frication-start 500 0 6000 ?*silence-power-upper*))
;; this rule is activated by the fact there is no such a pattern, as
;; well as the goal is generated. In other word, this rule is
;; activated before the backward-chaining rule is fired. (See ART
;; Programming Tutorial, Vol 3, page.1-34) In fact, however, the
;; backward-chaining rule is fired prior to this rule, since the
;; salience of the backward-chaining rule is larger than that of this
;; rule. Then, as the result of the backward-chaining rule, if the
;; fact which match this goal is asserted, i.e. the power-end time is
;; found, this rule is fired next. In another case, if the
;; power-end time was not found and the fact (power-end NONE ...)
;; which does not match this goal is asserted, this rule is removed
;; from the agenda as soon as the fact is asserted. As the result,
;; this rule works correctly. However, the efficiency seems bad.
;; [12/04/87 21:06:23 hatazaki]
;;
;;
;;(case
;;  ((and (power-start ?word-start&~NONE
;;        before =(+ ?frication-start 30) 60 0 6000 ?*silence-power-upper*)
;;        (power-end ?y&:(eq ?y 'NONE) before ?word-start 500 0 6000 ?*silence-power-upper*))
;;    =>
;;      (bind ?CFno-power-before 0.8))
;;  (otherwise
;;    =>
;;      (bind ?CFno-power-before -0.8)))
=>
(retract ?x)
;;
;;[hatazaki 167]
;;   =(CFand (CFcomb ?CFcontext
;;                 (CFweight ?*evidence* ?CFno-power-before))
;;           ?CFno-power-before)
;;   <-- =(CFcomb ?CFcontext
;;                (CFweight ?*strong-evidence* ?CFno-power-before))
;;
;;
(assert (CF (left-context ?segment word-initial)
           =(CFand (CFcomb ?CFcontext
                           (CFweight ?*evidence* ?CFno-power-before))
                   ?CFno-power-before)))
(assert (applied ?segment sg-uvfric-at-word-initial-4)))

```

```

;;;
;;;[hatazaki 148] Moved sg-uvfric-at-word-initial-5 to sg-uvfric-end-at-word-initial-2
;;;
;(defrule sg-uvfric-at-word-initial-5
;  "There is no voice-bar (low-frequency power) before this segment."
;  (declare (salience ?*left-segmentation*))
;  (segment-status ?segment segmentation)
;  (join
;    (category ?segment unvoiced-fricative)
;    (CF (category ?segment unvoiced-fricative)
;        ?CFcategory&:(?CFcategory >= ?*mightbe-valid*)))
;  (join
;    (left-context ?segment word-initial)
;    ?x <- (CF (left-context ?segment word-initial)
;              ?CFcontext&:(?CFcontext >= ?*mightbe-valid*)))
;  (not (applied ?segment sg-uvfric-at-word-initial-5))
;  (prop ?segment (candidate-loc ?from ?to))
;  (prop ?segment (frication-start ?frication-start))
;  (power-strength =(- ?frication-start 30) ?frication-start 0 200 ?0-200-P)
;  (CF (unvoiced-fricative 0-200-power-at-word-initial ?0-200-P) ?CF0-200-P)
;  =>
;  (assert (applied ?segment sg-uvfric-at-word-initial-5))
;  (retract ?x)
;  (assert (CF (left-context ?segment word-initial)
;              =(CFand ?CFcontext
;                    (CFweight ?*evidence* ?CF0-200-P))))))

(defrule sg-uvfric-at-word-initial-done
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)
    (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-at-word-initial-done))
  (prop ?segment (frication-start ?frication-start))
  =>
  (assert (applied ?segment sg-uvfric-at-word-initial-done))
  (hypothesize
    (assert (start-time ?segment ?frication-start))
    (assert (CF (start-time ?segment ?frication-start) =(CFvalue ?CFcontext))))))

```

---



```

(defrule sg-uvfric-aft-stop-1
  "Finds frication start time."
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (join
    (left-context ?segment stop)
    (CF (left-context ?segment stop) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvfric-aft-stop-1))
  (prop ?segment (candidate-loc ?from ?to))
  (power-increase (?frication-start&~NONE ?fs-change)
    before ?to 200 4000 6000
    ?*normal-smoothing-size* ?*frication-start-min-change*)
  =>
  (assert (applied ?segment sg-uvfric-aft-stop-1))
  (assert (prop ?segment (frication-start ?frication-start))))

(defrule sg-uvfric-aft-stop-2
  "Checks vowel-offset-time if there is a vowel before here."
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (join
    (left-context ?segment stop)
    ?x <- (CF (left-context ?segment stop) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvfric-aft-stop-2))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (frication-start ?frication-start))
  (power-decrease (?vowel-end&~NONE ?ve-change)
    before =(+ ?from 20) 100 1000 2500
    ?*normal-smoothing-size* ?*vowel-F2-end-min-change*)
  (CF (unvoiced-fricative vowel-offset-time =(- ?frication-start ?vowel-end)) ?CFvoff-time)
  =>
  (assert (applied ?segment sg-uvfric-aft-stop-2))
  (retract ?x)
  (assert (CF (left-context ?segment stop)
    =(CFand (CFcomb ?CFcontext
      (CFweight ?*evidence* (- ?CFvoff-time)))
    (- ?CFvoff-time))))))

(defrule sg-uvfric-aft-stop-3
  "Checks power of the preceding stop."
  (declare (saliency ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (join
    (left-context ?segment stop)
    ?x <- (CF (left-context ?segment stop) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvfric-aft-stop-3))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (frication-start ?frication-start))
  (power-strength =(- ?frication-start 30) ?frication-start 0 500 ?stop-0-500-power)
  (CF (unvoiced-stop 0-500-power-at-aspiration-end ?stop-0-500-power) ?CF0-500-power)
  (power-strength =(- ?frication-start 30) ?frication-start 4000 6000 ?stop-4000-6000-power)
  (CF (unvoiced-stop 4000-6000-power-at-aspiration ?stop-4000-6000-power)
    ?CF4000-6000-power)
  =>
  (assert (applied ?segment sg-uvfric-aft-stop-3))
  (retract ?x)
  (assert (CF (left-context ?segment stop)
    =(CFand (CFcomb ?CFcontext
      (CFweight ?*no-evidence* ?CF0-500-power)
      (CFweight ?*weak-evidence* ?CF4000-6000-power))
    ?CF0-500-power
    ?CF4000-6000-power))))))

(defrule sg-uvfric-aft-stop-4
  "Checks duration of the preceding stop."

```

```

(declare (salience ?*left-segmentation*))
(segment-status ?segment segmentation)
(join
  (category ?segment unvoiced-fricative)
  (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
(join
  (left-context ?segment stop)
  ?x <- (CF (left-context ?segment stop) ?CFcontext&: mightbe-valid))
(not (applied ?segment sg-uvfric-aft-stop-4))
(prop ?segment (candidate-loc ?from ?to))
(prop ?segment (frication-start ?frication-start))
(power-increase (?aspiration-start&~NONE ?as-change)
  before ?frication-start 150 4000 6000
  ?*normal-smoothing-size* ?*closure-end-min-change*)
(CF (unvoiced-stop aspiration-duration-before-fricative
  =(- ?frication-start ?aspiration-start)) ?CFaspiration)
=>
(assert (applied ?segment sg-uvfric-aft-stop-4))
(retract ?x)
(assert (CF (left-context ?segment stop)
  =(CFand (CFcomb ?CFcontext
    (CFweight ?*evidence* ?CFaspiration))
    ?CFaspiration))))

(defrule sg-uvfric-aft-stop-done-1
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment stop)
    ?x <- (CF (left-context ?segment stop) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-aft-stop-done-1))
  (prop ?segment (frication-start ?frication-start))
  =>
  (assert (applied ?segment sg-uvfric-aft-stop-done-1))
  (hypothesize
    (assert (start-time ?segment ?frication-start))
    (assert (CF (start-time ?segment ?frication-start)
      =(CFcomb (CFweight ?*no-evidence* ?CFcategory)
        (CFweight ?*definite-evidence* ?CFcontext))))))
;-----

```

```

;(defrule sg-uvfric-with-closure-0
;  "There is a closure left of the high-freq-power area."
;  (declare (salience ?*left-segmentation*))
;  (segment-status ?segment segmentation)
;  (category ?segment unvoiced-fricative)
;  (left-context ?segment closure)
;  (not (applied ?segment sg-uvfric-with-closure-0))
;  ?old-CF <- (CF (left-context ?segment closure) ?CF)
;  ;;
;  ;; This rule does not work correctly if the next form is
;  ;; (closure-exists ?from ?to left-of ?segment
;  ;;           ?closure-CF&(>= ?closure-CF 0.6))
;  ;; I don't know the reason.
;  ;;
;  (closure-exists ?from ?to left-of ?segment
;    ?closure-CF&(?closure-CF >= 0.6))
;  =>
;  (assert (applied ?segment sg-uvfric-with-closure-0))
;  (retract ?old-CF)
;  (assert (CF (left-context ?segment closure) =(#LUSER::comb-CF ?CF ?closure-CF)))
;  (assert (prop ?segment (closure ?from ?to))))
;
;(defrule sg-uvfric-with-closure-done
;  "Closure-start time is the segment start time."
;  (declare (salience ?*left-segmentation-end*))
;  (segment-status ?segment segmentation)
;  (category ?segment unvoiced-fricative)
;  (left-context ?segment closure)
;  (not (applied ?segment sg-uvfric-with-closure-done))
;  (CF (left-context ?segment closure) ?CF&(>= ?CF 0.6))
;  (prop ?segment (closure ?cl-start ?cl-end))
;  =>
;  (assert (applied ?segment sg-uvfric-with-closure-done))
;  (hypothesize
;    (assert (start-time ?segment ?cl-start))
;    (assert (CF (start-time ?segment ?cl-start) ?CF))))
;
;(defrule get-closure-left-of-unvoiced-frication-0
;  "Gets a closure preceding to the unvoiced frication.
;  (BACKWARD CHAINING RULE)"
;  (declare (salience ?*top-down*))
;  (goal (closure-exists ?from&?? ?to&?? left-of ?segment&~?? ?CF&??))
;  (category ?segment unvoiced-fricative)
;  (segment-status ?segment segmentation)
;  (prop ?segment (candidate-loc ?from ?to))
;  ;;(power-increase ?closure-end before ?hfe-start 50 3000 6000)
;  (power-start ?cl-end&~NONE
;    before =(+ ?from 20) 100 4000 6000 ?*closure-power-upper*)
;  ;;
;  ;;- ?cl-end and ?from should not separates more than 15 msec.
;  ;; (cf. MAU_B_0154)[11/16/87 15:57:52].
;  ;;- 15 msec separation should be OK.
;  ;; (cf. MAU_B_0007)[11/16/87 21:57:17].
;  ;;
;  (test (<= (- ?from ?cl-end) 15))
;  (power-end ?cl-start&~NONE
;    before ?cl-end 400 1000 6000 ?*closure-power-upper*)
;  (test (and (>= (- ?cl-end ?cl-start) 20)
;    (<= (- ?cl-end ?cl-start) 100)))
;  (power-strength ?cl-start ?cl-end 1000 6000
;    ?cl-power&(< ?cl-power ?*closure-power-upper*))
;  (BB ?BB)
;  =>
;  (at ?BB
;    (assert (closure-exists ?cl-start ?cl-end left-of ?segment 0.8))))
;
;-----

```

```

;(defcontradiction sg-uvfric-aft-syllabic-nasal-0
; "There must not be high power before this segment."
; (segment-status ?segment segmentation)
; (category ?segment unvoiced-fricative)
; (left-context ?segment closure)
; (not (applied ?segment sg-uvfric-aft-syllabic-nasal-0))
; (prop segment
;   (high-freq-power ?high-freq-start-time ?high-freq-end-time
;     ?high-freq-bottom ?high-freq-top))
; (power-strength =(- ?high-freq-start-time 20)
;   ?high-freq-start-time
;   4000 6000
;   ?strength&:(> ?strength ?*no-power-upper*)))

;(defrule sg-uvfric-aft-syllabic-nasal-1
; (declare (salience ?*left-segmentation*))
; (segment-status ?segment segmentation)
; (prop ?segment (high-freq-power ?high-freq-start-time ? ? ?))
; (left-context ?segment syllabic-nasal)
; ?old-CF <- (CF ?segment ?CF)
; (not (applied ?segment sg-uvfric-aft-syllabic-nasal-1))
; (burst ?time1&::~: numberp ?type1 after ?high-freq-start-time 20)
; (burst ?time2&::~: numberp ?type2 before ?high-freq-start-time 20)
; =>
; (retract ?old-CF)
; (assert (CF ?segment =(+ ?CF 0.2))
;   (applied ?segment sg-uvfric-aft-syllabic-nasal-1)))

;(defrule sg-uvfric-aft-syllabic-nasal-2
; (declare (salience ?*left-segmentation*))
; (segment-status ?segment segmentation)
; (prop ?segment (high-freq-power ?high-freq-start-time ? ? ?))
; (left-context ?segment syllabic-nasal)
; ?old-CF <- (CF ?segment ?CF)
; (not (applied ?segment sg-uvfric-aft-syllabic-nasal-2))
; (power-end ?nasal-start-time&:: numberp
;   before ?high-freq-start-time 500
;   ?*high-freq-bottom* ?*high-freq-top* 0.65)
; =>
; (assert (applied ?segment sg-uvfric-aft-syllabic-nasal-2))
; (hypothesize
;   (assert (preceded-by ?segment
;     (syllabic-nasal ?nasal-start-time unknown))))))

;(defrule sg-uvfric-aft-syllabic-nasal-3
; (declare (salience ?*left-segmentation*))
; (segment-status ?segment segmentation)
; (prop ?segment (high-freq-power ?high-freq-start-time ? ? ?))
; (preceded-by ?segment
;   (syllabic-nasal ?nasal-start-time&~unknown ?nasal-end-time))
; ?old-CF <- (CF ?segment ?CF)
; (not (applied ?segment sg-uvfric-aft-syllabic-nasal-3))
; (has-formant ?nasal-start-time ?high-freq-start-time
;   ?*F2-bottom* ?*F3-top*
;   ?strength&:(?strength > ?*no-power-upper*))
; =>
; (retract ?old-CF)
; (assert (CF ?segment =(+ ?CF 0.2))
;   (applied ?segment sg-uvfric-aft-syllabic-nasal-3)))

;(defrule sg-uvfric-aft-syllabic-nasal-4
; (declare (salience ?*left-segmentation*))
; (segment-status ?segment segmentation)
; (prop ?segment (high-freq-power ?high-freq-start-time ? ? ?))
; (preceded-by ?segment
;   (syllabic-nasal ?nasal-start-time&~unknown ?nasal-end-time))
; ?old-CF <- (CF ?segment ?CF)
; (not (applied ?segment sg-uvfric-aft-syllabic-nasal-4))
; (has-formant ?nasal-start-time ?high-freq-start-time
;   ?*F1-bottom* ?*F1-top*
;   ?strength&:(?strength > ?*no-power-upper*))
; =>
; (retract ?old-CF)
; (assert (CF ?segment =(+ ?CF 0.2))

```

```

;      (applied ?segment sg-uvfric-aft-syllabic-nasal-4)))
;(defrule sg-uvfric-aft-syllabic-nasal-5
; (declare (salience ?*left-segmentation*))
; (segment-status ?segment segmentation)
; (prop ?segment (high-freq-power ?high-freq-start-time
;                               ?high-freq-end-time
;                               ?high-freq-bottom
;                               ?high-freq-top))
; (preceded-by ?segment
;              (syllabic-nasal ?nasal-start-time&~unknown ?nasal-end-time))
; ?old-CF <- (CF ?segment ?CF)
; (not (applied ?segment sg-uvfric-aft-syllabic-nasal-5))
; (power-start ?start-time&:numberp before ?high-freq-end-time 300
;              ?high-freq-bottom ?high-freq-top 0.65)
; =>
; (assert
; (applied ?segment sg-uvfric-aft-syllabic-nasal-5))
; (hypothesize
; (assert (start-time ?segment ?start-time))))

```

```

;(defrule sg-uvfric-aft-syllabic-nasal-end
; (declare (salience ?*end-segmentation*))
; ?x <- (segment-status ?segment segmentation)
; (prop ?segment (high-freq-power $?))
;
; (preceded-by ?segment (syllabic-nasal $?))
; (start-time ?segment ?)
; (end-time ?segment ?)
; (CF ?segment ?CF&:(?CF >= 0.4))
; =>
; (retract ?x)
; (assert (segment-status ?segment recognition)))

```

LM17:>hatazaki>sre>seg-unvoiced-fricative>seg-fricative-right. art. 37

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 12/12/88 11:58:23

Queued at: 3/28/89 10:49:51

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-uvfric-right-1
  "Hypothesizes right phoneme."
  (declare (salience ?*right-segmentation*))
  ;(declare (salience ?*start-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (not (applied ?segment sg-uvfric-right-1))
  (not (end-time ?segment ?)))
=>
(assert (applied ?segment sg-uvfric-right-1))
(conflicting-alternatives
  (hypothesize
    (assert (right-context ?segment vowel))
    (assert (CF (right-context ?segment vowel) =(CFunknown))))))
(hypothesize
  (assert (right-context ?segment stop-or-affricate))
  (assert (CF (right-context ?segment stop-or-affricate) =(CFunknown))))
; (hypothesize
;   (assert (right-context ?segment fricative))
;   (assert (CF (right-context ?segment fricative) =(CFunknown))))
; (hypothesize
;   (assert (right-context ?segment nasal))
;   (assert (CF (right-context ?segment nasal) =(CFunknown))))
; (hypothesize
;   (assert (right-context ?segment silence))
;   (assert (CF (right-context ?segment silence) =(CFunknown))))
))
```

```
-----
```

```

;(defrule sg-uvfric-bfr-vowel-0
; "Get time range for searching vowel start time."
; (declare (salience ?*right-segmentation*))
; (segment-status ?segment segmentation)
; (category ?segment unvoiced-fricative)
; (right-context ?segment vowel)
; (not (applied ?segment sg-uvfric-bfr-vowel-0))
; (prop ?segment (candidate-loc ?from ?to))
; (power-start ?max-end-time after =(+ ?from 20) 500 0 500 ?*voicing-power-lower*)
; =>
; ; - ?max-end-time may be NONE.
; (assert (prop ?segment (following-vowel-start ?max-end-time)))
; (assert (applied ?segment sg-uvfric-bfr-vowel-2)))

```

```

(defrule sg-uvfric-bfr-vowel-1
  "Searches the vowel start time"
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment vowel)
    ?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-bfr-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))
  ;;
  ;;[hatazaki 145] ?vowel-region is found first, then ?vowel-start precisely before
  ;; ?vowel-region.
  ;; <--
  ;; (power-increase (?vowel-start&~NONE ?change)
  ;;               after =(+ ?from 20) 300
  ;;               0 500
  ;;               ?*normal-smoothing-size*
  ;;               ?*vowel-start-min-change-after-unvoiced-fricative*)
  ;;
  ;;- ?max-end-time seems non-sense, since power-increase
  ;; searches the FIRST local maximum. This sexp is removed and
  ;; =(- ?max-end-time ?from -20) is replaced with just 300.
  ;; [hatazaki 3/18/88 17:38:20]
  ;;(prop ?segment (following-vowel-start ?max-end-time&~NONE))
  ;;
  (power-start ?vowel-region&~NONE
    after =(+ ?from 20) 300 0 500
    ?*voicing-power-lower-after-unvoiced-fricative*)
  (power-increase (?vowel-start&~NONE ?change)
    before =(+ ?vowel-region 20) 60
    0 500
    ?*normal-smoothing-size*
    ?*vowel-start-min-change-after-unvoiced-fricative*)
  (power-strength =(+ ?vowel-start 7.5) =(+ ?vowel-start 22.5) 0 500 ?0-500-P)
  (CF (vowel 0-500-power ?0-500-P) ?CF0-500-P)
  =>
  (assert (prop ?segment (following-vowel-start ?vowel-start)))
  (retract ?x)
  ;;
  ;;[hatazaki 167]
  ;; =(CFand (CFcomb ?CFcontext
  ;;           (CFweight ?*evidence* ?CF0-500-P))
  ;;         ?CF0-500-P)
  ;; <-- =(CFcomb ?CFcontext
  ;;         (CFweight ?*strong-evidence* ?CF0-500-P))
  ;;
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFcontext
      (CFweight ?*evidence* ?CF0-500-P))
      ?CF0-500-P)))
  (assert (applied ?segment sg-uvfric-bfr-vowel-1)))

```

```

(defrule sg-uvfric-bfr-vowel-1b
  "Searches the vowel start time with smaller threshold."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join

```



```

(category ?segment unvoiced-fricative)
(CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
(join
(right-context ?segment vowel)
?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
(not (applied ?segment sg-uvfric-bfr-vowel-1b))
(prop ?segment (candidate-loc ?from ?to))
(power-start ?y&:(eq ?y 'NONE)
after =(+ ?from 20) 300 0 500
?*voicing-power-lower-after-unvoiced-fricative*)
(power-start ?vowel-region&~NONE
after =(+ ?from 20) 300 0 500
?*voicing-power-smaller-lower-after-unvoiced-fricative*)
(power-increase (?vowel-start&~NONE ?change)
before =(+ ?vowel-region 20) 60
0 500
?*normal-smoothing-size*
?*vowel-start-min-change-after-unvoiced-fricative*)
(power-strength =(+ ?vowel-start 7.5) =(+ ?vowel-start 22.5) 0 500 ?0-500-P)
(CF (vowel 0-500-power ?0-500-P) ?CF0-500-P)
=>
(assert (prop ?segment (following-vowel-start ?vowel-start)))
(retract ?x)
(assert (CF (right-context ?segment vowel)
=(CFand (CFcomb ?CFcontext
(Cfweight ?*evidence* ?CF0-500-P))
?CF0-500-P)))
(assert (applied ?segment sg-uvfric-bfr-vowel-1b)))
(defrule sg-uvfric-bfr-vowel-3
"Checks the vowel is not devoiced. (cf. MAU_B_0070)"
;;
;; - If there is closure which precedes to this phoneme,
;; kill this hypothesis by a CFvalue -0.9. [hatazaki 3/18/88 14:40:50]
;;
(declare (salience ?*right-segmentation*))
(segment-status ?segment segmentation)
(join
(category ?segment unvoiced-fricative)
(CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
(join
(right-context ?segment vowel)
?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
(not (applied ?segment sg-uvfric-bfr-vowel-3))
(prop ?segment (candidate-loc ?from ?to))
(prop ?segment (following-vowel-start ?vowel-start)))
;;
;;[hatazaki 145]
;; (closure-exists ?cl-from ?cl-to before ?vowel-start =(- ?vowel-start ?to) ?CFclosure)
;; <--
;; (power-decrease
;; (?cl-start&~NONE ?)
;; after =(+ ?from 20) =(- ?vowel-start ?from 20)
;; 5000 6000 ?*normal-smoothing-size* ?*closure-start-min-change*)
;; (test (>= (- ?vowel-start ?cl-start) 20))
;; (power-strength ?cl-start =(+ ?cl-start 20) 1000 6000
;; ?cl-power&:(?cl-power <= ?*imcomplete-closure-power-upper*))
;;
;; - Closure start time is searched by 5000-6000Hz frequency power, instead ?cutoff-6000Hz.
;; - ?*closure-start-min-change* is changed to 0.5 and ?*normal-smoothing-size* to 19.
;; [hatazaki 4/19/88 00:30:31]
;; (power-decrease
;; (?cl-start&~NONE ?)
;; after =(+ ?from 20) =(- ?vowel-start ?from 20)
;; ?cutoff 6000 ?*normal-smoothing-size* ?*closure-start-min-change*)
;;
(closure-exists ?cl-from ?cl-to before ?vowel-start =(- ?vowel-start ?to) ?CFclosure)
=>
(retract ?x)
;;
;;[hatazaki 167]
;; =(CFand (CFcomb ?CFcontext
;; (CFweight ?*evidence* (- ?CFclosure)))
;; (- ?CFclosure))
;; <-- =(CFcomb ?CFcontext
;; (CFweight ?*definite-evidence* (- ?CFclosure)))

```

```
;;
(assert (CF (right-context ?segment vowel)
            =(CFand (CFcomb ?CFcontext
                          (CFweight ?*evidence* (- ?CFclosure)))
                    (- ?CFclosure))))
(assert (applied ?segment sg-uvfric-bfr-vowel-3)))

(defrule sg-uvfric-bfr-vowel-done-1
  ""
  (declare (salience ?*right-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
   (category ?segment unvoiced-fricative)
   (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (join
   (right-context ?segment vowel)
   (CF (right-context ?segment vowel) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvfric-bfr-vowel-done-1))
  (prop ?segment (following-vowel-start ?vowel-start))
  =>
  (assert (applied ?segment sg-uvfric-bfr-vowel-done-1))
  (hypothesize
   (assert (end-time ?segment ?vowel-start))
   (assert (CF (end-time ?segment ?vowel-start) =(CFvalue ?CFcontext))))))
;-----
```

```

(defrule sg-uvfric-bfr-stop-or-affr-1
  ""
  (declare (saliency ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment unvoiced-fricative)
  (right-context ?segment stop-or-affricate)
  (CF (right-context ?segment stop-or-affricate) ?CF)
  (not (applied ?segment sg-uvfric-bfr-stop-or-affr-1))
  =>
  (assert (applied ?segment sg-uvfric-bfr-stop-or-affr-1))
  (conflicting-alternatives
  ;
  ;;
  ;; - Complete closure can be included in incomplete closure cases.
  ;; [hatazaki 1/11/88 19:29:59]
  ;; (hypothesize
  ;;   (assert (right-closure ?segment complete))
  ;;   (assert (CF (right-closure ?segment complete) 0.0)))
  (hypothesize
  ;; - Closure may have weak power.
  (assert (right-closure ?segment incomplete))
  (assert (CF (right-closure ?segment incomplete) 0.0)))
  (hypothesize
  ;; - Closure cannot be seen.
  (assert (right-closure ?segment none))
  (assert (CF (right-closure ?segment none) 0.0))))))

(defrule sg-uvfric-bfr-stop-or-affr-2
  "Get the maximum time of the right boundary."
  (declare (saliency ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
  (category ?segment unvoiced-fricative)
  (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (join
  (right-context ?segment stop-or-affricate)
  (CF (right-context ?segment stop-or-affricate) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvfric-bfr-stop-or-affr-2))
  (prop ?segment (candidate-loc ?from ?to))
  ;;
  ;; - Get the time next vowel starts. The right boundary of this
  ;; fricative should not be searched beyond this time.
  ;;
  ;; - Search starts a little after ?from, since there may be weak
  ;; power in the unvoiced area (cf. MAU_1_3139, 505sec) [hatazaki
  ;; 1/08/88 11:12:23].
  (power-start ?max-end-time&-NONE after =(+ ?from 20) 500 0 500 ?*voicing-power-lower*)
  =>
  (assert (applied ?segment sg-uvfric-bfr-stop-or-affr-2))
  ;;
  ;; - Max-end-time should be little bit right of the ?max-end-time,
  ;; since there is a case when the closure end time and vowel start
  ;; time are almost same (cf. MAU_B_0100) [hatazaki 3/18/88 16:15:34]
  ;;
  (assert (prop ?segment (max-end-time =(+ ?max-end-time 20.0)))))

(defrule sg-uvfric-bfr-stop-or-affr-3
  ""
  (declare (saliency ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
  (category ?segment unvoiced-fricative)
  (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (join
  (right-context ?segment stop-or-affricate)
  ?x <- (CF (right-context ?segment stop-or-affricate) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvfric-bfr-stop-or-affr-3))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (max-end-time ?max-end-time))
  ;;
  ;; [hatazaki 145]
  ;; (closure-exists ?cl-from ?cl-to after ?to =(- ?max-end-time ?to) ?CFclosure)
  ;; <--
  ;; (power-decrease
  ;; (?cl-start&-NONE ?cl-start-change))

```

```

;; after ?to =(- ?max-end-time ?to) 5000 6000
;; ?*normal-smoothing-size* ?*closure-start-min-change*
;; (power-increase
;;   (?cl-end&~NONE ?cl-end-change)
;;   after ?cl-start =(- ?max-end-time ?cl-start) 5000 6000
;;   ?*normal-smoothing-size* ?*closure-end-min-change*)
;; ;;
;; ;; - Accept this is closure only if the duration is longer than ?*closure-min-duration*.
;; ;; (cf. MAU_B_0122 should not be a closure (duration is 25msec))
;; ;; [hatazaki 5/17/88 09:16:06]
;; (test (>= (- ?cl-end ?cl-start) ?*closure-min-duration*))
;; (power-strength ?cl-start ?cl-end 1000 6000
;;   ?cl-power&(<= ?cl-power ?*imcomplete-closure-power-upper*))
;; ;;
;; ;;- In case of phoneme which follows to strong power (cf.
;; MAU_1_2067, 500ms, /ts/), spectrum power goes down near ?from,
;; then the closure start time of next phoneme should be searched
;; starting time a little after ?from, that is, (+ ?from 20).
;; [hatazaki 1/19/88 17:00:58]
;; ;;
;;- Power-change should be larger than a predetermined value.
;; (cf. MAU_1_2067, 487.5, ?cl-start-change=1.80 should be accepted.)
;; (cf. MAU_1_2067, 570ms, ?cl-start-change=0.73 should be ignored.)
;; ;;
;;- Power start/end time is searched by 5000-6000Hz power, instead ?cutoff-6000
;; or 4000-6000Hz. [hatazaki 4/19/88 00:38:21]
;; ;;
(closure-exists ?cl-from ?cl-to after =(- ?to 20) =(- ?max-end-time ?to) ?CFclosure)
=>
(assert (applied ?segment sg-uvfric-bfr-stop-or-affr-3))
(retract ?x)
;;
;;[hatazaki 167]
;;   =(CFand (CFcomb ?CFcontext
;;             (CFweight ?*strong-evidence* ?CFclosure))
;;           ?CFclosure)
;; <--   =(CFcomb ?CFcontext
;;             (CFweight ?*strong-evidence* ?CFclosure))
;; ;;
(assert (CF (right-context ?segment stop-or-affricate)
  =(CFand (CFcomb ?CFcontext
    (CFweight ?*strong-evidence* ?CFclosure))
    ?CFclosure)))
(assert (prop ?segment (right-closure ?cl-from ?cl-to)))

(defrule sg-uvfric-bfr-stop-or-affr-4
  "Checks if the vowel is surely devoiced."
  ;;
  ;; If there is vowel power between this fricative and the following closure,
  ;; rejects this hypothesis. (cf. MAU_B_0100 [hatazaki 5/17/88 09:00:38]
  ;;
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment unvoiced-fricative)
    (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
  (join
    (right-context ?segment stop-or-affricate)
    ?x <- (CF (right-context ?segment stop-or-affricate) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-uvfric-bfr-stop-or-affr-4))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (right-closure ?cl-start ?cl-end))
  (power-increase (?vowel-start&~NONE ?)
    after =(- ?to 20) =(- ?cl-start ?to -20) 0 500
    ?*normal-smoothing-size*
    ?*vowel-start-min-change-after-unvoiced-fricative*)
  (power-strength =(+ ?vowel-start 7.5) =(+ ?vowel-start 22.5) 0 500 ?0-500-P)
  (CF (vowel 0-500-power ?0-500-P) ?CFvowel)
  =>
  (retract ?x)
  ;;
  ;;[hatazaki 167] Fixed bug (- ?CFvowel) <-- ?CFvowel.
  ;;   =(CFand (CFcomb ?CFcontext
  ;;             (CFweight ?*evidence* (- ?CFvowel)))
  ;;           (- ?CFvowel))
  ;;

```

```

;; <--   =(CFcomb ?CFcontext
;;         (CFweight ?*evidence* ?CFvowel))
;;
(assert (CF (right-context ?segment stop-or-affricate)
           =(CFand (CFcomb ?CFcontext
                       (CFweight ?*evidence* (- ?CFvowel)))
                 (- ?CFvowel))))
(assert (applied ?segment sg-uvfric-bfr-stop-or-affr-4)))

(defrule sg-uvfric-bfr-stop-or-affr-done-1
  ""
  (declare (saliency ?*right-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
   (category ?segment unvoiced-fricative)
   (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
  (join
   (right-context ?segment stop-or-affricate)
   (CF (right-context ?segment stop-or-affricate) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-uvfric-bfr-stop-or-affr-done-1))
  (prop ?segment (right-closure ?cl-start ?cl-end))
  =>
  (assert (applied ?segment sg-uvfric-bfr-stop-or-affr-done-1) )
  (hypothesize
   (assert (end-time ?segment ?cl-start))
   (assert (CF (end-time ?segment ?cl-start) =(CFvalue ?CFcontext))))))

;-----
;;; - Cases where the following stop or affricate has complete closure
;;; can be included in cases whrer they has imcomplete closure.
;(defrule sg-uvfric-bfr-stop-or-affr-with-comp-closure-1
;  "Finds closure start and end time"
;  (declare (saliency ?*right-segmentation*))
;  (segment-status ?segment segmentation)
;  (category ?segment unvoiced-fricative)
;  (right-context ?segment stop-or-affricate)
;  (right-closure ?segment complete)
;  ?x <- (CF (right-closure ?segment complete) ?CF)
;  (not (applied ?segment
;            sg-uvfric-bfr-stop-or-affr-with-comp-closure-1))
;  (prop ?segment (candidate-loc ?from ?to))
;  (prop ?segment (high-freq-power ? ? ?cutoff))
;  (prop ?segment (max-end-time ?max-end-time))
;  (power-end ?cl-start&~NONE
;    after ?from =(- ?max-end-time ?from) ?cutoff 6000 ?*closure-power-upper*)
;  ;; - Checks the power goes down suddenly.
;  (power-decrease (? ?change&:(?change >= ?*closure-start-change-lower*))
;    after =(- ?cl-start 7.5) 15 ?cutoff 6000)
;  (power-start ?cl-end&~NONE
;    after ?cl-start =(- ?max-end-time ?cl-start) 1000 6000 ?*closure-power-upper*)
;  (power-strength ?cl-start ?cl-end 1000 6000
;    ?cl-power&:(?cl-power <= ?*closure-power-upper*))
;  =>
;  (assert (applied ?segment sg-uvfric-bfr-stop-or-affr-with-comp-closure-1))
;  (retract ?x)
;  (assert (CF (right-closure ?segment complete) =( #USER::comb-CF ?CF 0.8)))
;  (assert (prop ?segment (right-closure ?cl-start ?cl-end))))
;
;(defrule sg-uvfric-bfr-stop-or-affr-with-comp-closure-done
;  ""
;  (declare (saliency ?*right-segmentation*))
;  (segment-status ?segment segmentation)
;  (category ?segment unvoiced-fricative)
;  (not (applied ?segment
;            sg-uvfric-bfr-stop-or-affr-with-comp-closure-done))
;  (right-context ?segment stop-or-affricate)
;  (right-closure ?segment complete)
;  (CF (right-closure ?segment complete) ?CF&:(?CF >= 0.6))
;  (prop ?segment (right-closure ?cl-start ?cl-end))
;  =>
;  (assert (applied ?segment sg-uvfric-bfr-stop-or-affr-with-comp-closure-done))
;  (hypothesize
;   (assert (end-time ?segment ?cl-start))
;   (assert (CF (end-time ?segment ?cl-start) ?CF))))
;

```

LM17:>hatazaki>sre>seg-unvoiced-fricative>seg-fricative-end. art. 11

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/06/89 15:12:13

Queued at: 3/28/89 10:50:49

```

;;; -*- Mode: ART; Base: 10; Package: ART-USER -*-

;;;[hatazaki 178] Substituted by sg-uvfric-end-conflicting-with-uvstop-2.
;;;
;;;(defrule sg-uvfric-end-at-word-initial-1
;;;  "Checks power change at the beginning to reject stop."
;;;  ;;
;;;  ;;[hatazaki 148] Checks burst, instead of power-change.
;;;  ;; (burst-exists ?burst-start ?burst-end ?burst-freq
;;;  ;;   after =(- ?fricaction-start 20) 40 700 4000 ?CFburst)
;;;  ;; <--
;;;  ;; (prop ?segment (spectrum-global-peak (?freq ?peak-power)))
;;;  ;; (or (and (prop ?segment (spectrum-global-peak (?peak-freq&~NONE ?peak-power)))
;;;  ;;   (power-increase (?y&~NONE ?change)
;;;  ;;     after =(- ?fricaction-start 20) 40
;;;  ;;       =(- ?peak-freq 500) =(+ ?peak-freq 500)
;;;  ;;       ?*smoothing-for-getting-power-change-of-uvfricative-at-word-initial*
;;;  ;;       ?*fricaction-start-min-change-for-small-smoothing*))
;;;  ;; (and (prop ?segment (spectrum-global-peak (NONE ?)))
;;;  ;;   (power-increase (?y&~NONE ?change)
;;;  ;;     after =(- ?fricaction-start 20) 40
;;;  ;;       4000 6000
;;;  ;;       ?*smoothing-for-getting-power-change-of-uvfricative-at-word-initial*
;;;  ;;       ?*fricaction-start-min-change-for-small-smoothing*))
;;;  ;; (CF (unvoiced-fricative power-change-at-word-initial ?change) ?CFchange)
;;;  ;;[hatazaki 146, 11/09/88 20:23:23]
;;;  ;; Defined sg-uvfric-end-at-word-initial-1 to checks that this segment is
;;;  ;; fricative or stop, after gettig start and end time.
;;;  ;;
;;;  (declare (salience ?*before-end-segmentation*))
;;;  (segment-status ?segment segmentation)
;;;  (join
;;;    (category ?segment unvoiced-fricative)
;;;    ?x <- (CF (category ?segment unvoiced-fricative) ?CFcategory&:mightbe-valid))
;;;  (join
;;;    (start-time ?segment ?start)
;;;    (explicit (CF (start-time ?segment ?start) ?CFstart&:mightbe-valid)))
;;;  (join
;;;    (end-time ?segment ?end)
;;;    (explicit (CF (end-time ?segment ?end) ?CFend&:mightbe-valid)))
;;;  (join
;;;    (left-context ?segment word-initial)
;;;    (CF (left-context ?segment word-initial) ?CFcontext&:mightbe-valid))
;;;  (not (applied ?segment sg-uvfric-end-at-word-initial-1))
;;;  (prop ?segment (fricaction-start ?fricaction-start))
;;;  (burst-exists ?burst-start ?burst-end ?burst-freq
;;;    after =(- ?fricaction-start 20) 40 700 4000 ?CFburst)
;;;  (not (burst-exists ? ? ?
;;;    after =(- ?fricaction-start 20) 40 700 4000 ?y&:(?y > ?CFburst)))
;;;  (CF (unvoiced-stop voice-onset-time =(- ?end ?burst-start)) ?CFvot)
;;;  =>
;;;  (assert (applied ?segment sg-uvfric-end-at-word-initial-1))
;;;  (retract ?x)
;;;  ;;
;;;  ;;[hatazaki 168] Checks voice-onset-time to say this is a stop.
;;;  ;; (- (CFand ?CFburst ?CFvot)) <-- (- ?CFburst).
;;;  ;;[hatazaki 167]
;;;  ;;   =(CFand (CFcomb ?CFcategory
;;;  ;;     (CFweight ?*weak-evidence* (- ?CFburst)))
;;;  ;;     (- ?CFburst))
;;;  ;;   <--   =(CFcomb (CFweight ?*evidence* ?CFcategory)
;;;  ;;     (CFweight ?*strong-evidence* (- ?CFburst)))
;;;  ;;   ;;
;;;  (assert (CF (category ?segment unvoiced-fricative)
;;;    =(CFand (CFcomb ?CFcategory
;;;      (CFweight ?*weak-evidence* (- (CFand ?CFburst ?CFvot))))
;;;      (- (CFand ?CFburst ?CFvot))))))
;;;
(defrule sg-uvfric-end-at-word-initial-2
  "There is no voice-bar (low-frequency power) before this segment."
  ;;
  ;;[hatazaki 148] Moved from sg-uvfric-at-word-initial-5.
  ;;
  (declare (salience ?*before-end-segmentation*))
  (segment-status ?segment segmentation)

```

```

(join
  (category ?segment unvoiced-fricative)
  ?x <- (CF (category ?segment unvoiced-fricative) ?CFcategory&: mightbe-valid))
(join
  (start-time ?segment ?start)
  (explicit (CF (start-time ?segment ?start) ?CFstart&: mightbe-valid)))
(join
  (end-time ?segment ?end)
  (explicit (CF (end-time ?segment ?end) ?CFend&: mightbe-valid)))
(join
  (left-context ?segment word-initial)
  (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
(not (applied ?segment sg-uvfric-end-at-word-initial-2))
(prop ?segment (frication-start ?frication-start))
(power-strength =(- ?frication-start 30) ?frication-start 0 200 ?0-200-P)
(CF (unvoiced-fricative 0-200-power-at-word-initial ?0-200-P) ?CF0-200-P)
=>
(assert (applied ?segment sg-uvfric-end-at-word-initial-2))
(retract ?x)
;;
;;[hatazaki 167]
;;      =(CFand (CFcomb ?CFcategory
;;                (CFweight ?*weak-evidence* ?CF0-200-P)
;;                ?CF0-200-P)
;;      <--      =(CFcomb (CFweight ?*evidence* ?CFcategory)
;;                (CFweight ?*strong-evidence* ?CF0-200-P))
;;
;;
(assert (CF (category ?segment unvoiced-fricative)
           =(CFand (CFcomb ?CFcategory
                         (CFweight ?*weak-evidence* ?CF0-200-P)
                         ?CF0-200-P))))

;;;-----
(defrule sg-uvfric-end-conflicting-with-uvstop-1
  "Solve confliction with unvoiced stop"
  (declare (salience ?*after-end-segmentation*))
  (viewpoint ?v
    (join
      (segment-status ?segment1 segmentation-ended)
      (not (applied ?segment1 sg-uvfric-end-conflicting-with-uvstop-1))
      (join
        (category ?segment1 unvoiced-fricative)
        (CF (category ?segment1 unvoiced-fricative) ?CFcategory1&: mightbe-valid))
      (join
        (left-context ?segment1 stop)
        (CF (left-context ?segment1 stop) ?CFcontext&: mightbe-valid))
      (join
        (segment-result ?segment1 (?start1 ?end1))
        ?x <- (CF (segment-result ?segment1 (?start1 ?end1)) ?CFsegment1))))
  (viewpoint ?
    (join
      (segment-status ?segment2 segmentation-ended)
      (join
        (category ?segment2 unvoiced-stop)
        (CF (category ?segment2 unvoiced-stop) ?CFcategory2&: mightbe-valid))
      (join
        (segment-result ?segment2 (?start2 ?end2))
        (CF (segment-result ?segment2 (?start2 ?end2)) ?CFsegment2))))
  (test (and (< ?start2 ?start1)
             (< ?start1 ?end2)
             (< (/ (+ ?start1 ?end1) 2) ?end2)))
  =>
  (at ?v
    (assert (applied ?segment1 sg-uvfric-end-conflicting-with-uvstop-1))
    (retract ?x)
    (assert (CF (segment-result ?segment1 (?start1 ?end1))
               =(CFcomb ?CFsegment1 -0.5))))

(defrule sg-uvfric-end-conflicting-with-uvstop-2
  "Solve confliction with unvoiced stop at word head."
  (declare (salience ?*after-end-segmentation*))
  (viewpoint ?v
    (join
      (segment-status ?segment1 segmentation-ended)
      (not (applied ?segment1 sg-uvfric-end-conflicting-with-uvstop-2))

```



```

      (join
        (category ?segment1 unvoiced-fricative)
        (CF (category ?segment1 unvoiced-fricative) ?CFcategory1&: mightbe-valid))
      (join
        (left-context ?segment1 word-initial)
        (CF (left-context ?segment1 word-initial) ?CFcontext1&: mightbe-valid))
      (join
        (segment-result ?segment1 (?start1 ?end1))
        ?x <- (CF (segment-result ?segment1 (?start1 ?end1)) ?CFsegment1))))
(viewpoint ?
  (join
    (segment-status ?segment2 segmentation-ended)
    (join
      (category ?segment2 unvoiced-stop)
      (CF (category ?segment2 unvoiced-stop) ?CFcategory2&: mightbe-valid))
    (join
      (left-context ?segment2 word-initial)
      (CF (left-context ?segment2 word-initial) ?CFcontext2&: mightbe-valid))
    (join
      (segment-result ?segment2 (?start2 ?end2))
      (CF (segment-result ?segment2 (?start2 ?end2)) ?CFsegment2))))
(test (or (and (< ?start1 (/ (+ ?start2 ?end2) 2))
              (< (/ (+ ?start2 ?end2) 2) ?end1))
        (and (< ?start2 (/ (+ ?start1 ?end1) 2))
              (< (/ (+ ?start1 ?end1) 2) ?end2))))
=>
(at ?v
  (assert (applied ?segment1 sg-uvfric-end-conflicting-with-uvstop-2))
  (retract ?x)
  (assert (CF (segment-result ?segment1 (?start1 ?end1))
              =(CFcomb ?CFsegment1 -0.5))))))

```

LM17: >hatazaki>SRE>seg-h>get-h-region-candidates.lisp.15

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/02/89 10:55:52

Queued at: 3/28/89 11:25:24

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: ART-USER; Base: 10 -*-
```

```
(defun get-h-region-candidates()
  (let* ((start-time 1)
         (end-time 2000)
         h-start-time h-end-time
         head-section middle-section
         section0 section1 section2)

    (setq head-section
          (cond ((and (/= 0 (setq h-start-time
                                (car (get-power-start 0 2000 0 6000 -70))))
                  (/= 0 (setq h-end-time
                                (find-h-end-candidate h-start-time end-time))))
                (list (list h-start-time h-end-time))))))

    (setq middle-section (loop for i from start-time to end-time by 2.5
                              while (and (/= 0 (setq i (find-h-start-candidate i end-time)))
                                           (/= 0 (setq h-end-time (find-h-end-candidate i end-time)))
                                           (setq h-start-time i))
                              collect (list h-start-time h-end-time)))

    (setq section1
          (caddr (USER::call-speechfunc (list "POWER-ISLAND-IN-TIME" 0 2000
                                             0 1000
                                             -90 -35
                                             11 ))))

    (setq section2
          (caddr (USER::call-speechfunc (list "POWER-ISLAND-IN-TIME" 0 2000
                                             0 500
                                             -90 -45
                                             11 ))))

    (setq section0
          (loop for region in (intersected-regions (list section1 section2))
                collect region))

    (nconc
     (loop for region in (intersected-regions (list head-section section0))
           collect region)
     (loop for region in (intersected-regions (list middle-section section0))
           collect region))
  ))

(defun find-h-start-candidate (start-time end-time)
  (caar (caddr (USER::call-speechfunc
                (list "POWER-CHANGE"
                     start-time end-time
                     0 500
                     2 21
                     11 21
                     0.2 -65 )))))

(defun find-h-end-candidate (start-time end-time)
  (caar (caddr (USER::call-speechfunc
                (list "POWER-CHANGE"
                     start-time end-time
                     0 500
                     1 21
                     11 21
                     0.2 -65 )))))

;
;(defun get-h-region-candidates()
;  (let* (section1
;         section2)
;
;    (setq section1
;          (caddr (USER::call-speechfunc (list "POWER-RATIO-ISLAND" 0 2000
;                                             1000 5000
;                                             0 1000
;                                             0 999
;                                             11 21))))
;
;    (setq section2
;          (caddr (USER::call-speechfunc (list "POWER-ISLAND-IN-TIME" 0 2000
;                                             0 6000
```

```
;          -65    -30
;          15     ))))
;
; (loop for region in (intersected-regions (list section1 section2))
;   collect region)
; ))
```

LM17:>hatazaki>sre>seg-h>seg-h. art. 15

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/06/89 09:21:38

Queued at: 3/28/89 11:22:06

```
;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-h-0
  "Finds a candidate of /h/."
  (declare (salience ?*find-candidate*))
  (segmenting h)
  (h-region-candidate ?from ?to)
  (power-strength ?from ?to 0 1000 ?hpw-0-1000)
  (power-strength ?from ?to 1000 5000 ?hpw-1000-5000)
  (power-strength ?from ?to 5000 6000 ?hpw-5000-6000)

  (CF (h-0-1000-power ?hpw-0-1000) ?CFhpw-0-1000&:mightbe-valid)
  (CF (h-1000-5000-power ?hpw-1000-5000) ?CFhpw-1000-5000&:mightbe-valid)
  (CF (h-1000-5000-0-1000-power-ratio =(- ?hpw-1000-5000 ?hpw-0-1000)
      ?CFhpr-1000-5000-0-1000&:mightbe-valid)
      ?CFhpr-1000-5000-0-1000&:mightbe-valid)
  (CF (h-1000-5000-5000-6000-power-ratio =(- ?hpw-1000-5000 ?hpw-5000-6000)
      ?CFhpr-1000-5000-5000-6000&:mightbe-valid)
      ?CFhpr-1000-5000-5000-6000&:mightbe-valid)
  (BB ?BB)
  =>
  (bind ?segment (gensym "SEG-"))
  (at ?BB
    (assert (h-candidate =(/ (+ ?from ?to) 2) =(/ (+ ?from ?to) 2)))
    (assert (schema ?segment (instance-of phoneme-segment)))
    (hypothesize
      (assert (category ?segment h))
      (assert (prop ?segment (candidate-loc =(/ (+ ?from ?to) 2) =(/ (+ ?from ?to) 2)))
        (segment-status ?segment segmentation)
        (CF (category ?segment h)
            =(CFand (CFcomb
                    (CFunknown)
                    (CFweight ?*evidence* ?CFhpw-0-1000)
                    (CFweight ?*evidence* ?CFhpw-1000-5000)
                    (CFweight ?*evidence* ?CFhpr-1000-5000-0-1000)
                    (CFweight ?*evidence* ?CFhpr-1000-5000-5000-6000)
                    ?CFhpr-1000-5000-0-1000
                    ?CFhpr-1000-5000-5000-6000)))))))

(defrule get-h-region-candidates
  (declare (salience ?*top-down*))
  (goal (h-region-candidate ?from&?? ?to&??))
  (explicit (not (h-region-candidate ?from ?to)))
  =>
  (bind ?h-region-candidates
    (get-h-region-candidates))
  (for region in ?h-region-candidates
    do (assert =(seq$ (cons 'h-region-candidate region))))))
```

LM17:>hatazaki>sre>seg-h>seg-h-left.art.14

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/02/89 21:12:54

Queued at: 3/28/89 11:22:47

;;;-\*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -\*-

```
(defrule sg-h-left
  ""
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (CF (category ?segment h) ?CF-category&:mightbe-valid)
  (not (applied ?segment sg-h-left))
  (not (start-time ?segment ?))
  =>
  (assert (applied ?segment sg-h-left))
  (conflicting-alternatives
   (hypothesize
    (assert (left-context ?segment silence))
    (assert (CF (left-context ?segment silence) =(CFunknown))))
   (hypothesize
    (assert (left-context ?segment vowel))
    (assert (CF (left-context ?segment vowel) =(CFunknown))))
   (hypothesize
    (assert (left-context ?segment fricative))
    (assert (CF (left-context ?segment fricative) =(CFunknown))))
  ))
```

;;;-----



```

(defrule sg-h-left-silence-1
  "Find time of the silence by /h/ transition."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (left-context ?segment silence)
  (not (applied ?segment sg-h-left-silence-1))
  (prop ?segment (candidate-loc ?from ?to))

;;; search segmenetation position of left silence
  (power-start ?power-start&~NONE&:(< (abs (- ?from ?power-start)) 200)
    after 0 2000 0 6000 ?*silence-power-upper*)

  (power-decrease (?vowel-end&:(eq ?vowel-end 'NONE) ?change-decrease)
    after ?power-start =(abs (- ?from ?power-start))
    0 4000 (11 11) ?*vowel-end-min-change-before-h*)

  (spectrum-peak (?peak-freq&~NONE&:(?peak-freq >= 1000)&:(?peak-freq <= 6000)
    ?peak-amp ?peak-Q)
    at ?power-start =(+ ?power-start 10)
    ?*spectrum-peak-smoothing-for-phoneme-h*)
  (not (spectrum-peak (?pk-fq&~NONE&:(?pk-fq >= 1000)&:(?pk-fq <= 6000)
    ?another-peak-amp&:(> ?another-peak-amp ?peak-amp) ?pk-Q)
    at ?power-start =(+ ?power-start 10)
    ?*spectrum-peak-smoothing-for-phoneme-h*))

  (power-start ?h-start&~NONE&:(< (abs (- ?h-start ?power-start)) 50)
    after =(- ?power-start 50) 100
    =(- ?peak-freq 200) =(+ ?peak-freq 200) ?*h-left-is-silence-power*)

;;; check charasteritics of left silence
  (power-strength =(- ?h-start 50) ?h-start 0 6000 ?spw-0-6000)
  (CF (h-power-is-silence ?spw-0-6000) ?CFspw-0-6000&:mightbe-valid)

;;; check charasteritics of /h/
  ?x<-(CF (left-context ?segment silence) ?CFleft-silence)
=>
  (retract ?x)
  (assert (applied ?segment sg-h-left-silence-1))
  (assert (prop ?segment (proceding-silence-end-time ?h-start)))
  (assert (CF (left-context ?segment silence)
    =(CFand
      (CFcomb ?CFleft-silence
        (CFweight ?*evidence* ?CFspw-0-6000))
      ?CFspw-0-6000))))

(defrule sg-h-left-silence-done-1
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (CF (category ?segment h) ?CF-category&:mightbe-valid)
  (left-context ?segment silence)
  (CF (left-context ?segment silence) ?CF-context&:mightbe-valid)
  (not (applied ?segment sg-h-left-silence-done-1))
  (prop ?segment (proceding-silence-end-time ?pos))
=>
  (assert (applied ?segment sg-h-left-silence-done-1))
  (hypothesize
    (assert (start-time ?segment ?pos))
    (assert (CF (start-time ?segment ?pos) =(CFvalue ?CF-context))))))

;;; -----

```

```

(defrule sg-h-left-vowel-1
  "Find time of the vowel by /h/ transition."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (left-context ?segment vowel)
  (not (applied ?segment sg-h-left-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))

;;; search segmenetation position of left vowel
  (power-decrease (?vowel-end&~NONE ?change-decrease-0-4000)
    before ?from 200
    0 4000 (11 11) ?*vowel-end-min-change-before-h*)
  (spectral-change (?sp-chg-pos&~NONE ?value)
    before =(+ ?vowel-end 20) 40
    0 4000 ( 9 5) 7.5 0.0 0.1 2.8)
  (CF (h-spectral-change-left ?value) ?CFsp-change&: mightbe-valid)

  (spectrum-peak (?peak-freq&~NONE&:(?peak-freq >= 0)&:(?peak-freq <= 4000)
    ?peak-amp ?peak-Q)
    at =(- ?sp-chg-pos 10) ?sp-chg-pos
    ?*spectrum-peak-smoothing-for-phoneme-h*)
  (not (spectrum-peak (?pk-fq&~NONE&:(?pk-fq >= 0)&:(?pk-fq <= 4000)
    ?another-peak-amp&:(> ?another-peak-amp ?peak-amp) ?pk-Q)
    at =(- ?sp-chg-pos 10) ?sp-chg-pos
    ?*spectrum-peak-smoothing-for-phoneme-h*))
  (power-decrease (?pos&~NONE ?change-decrease)
    after =(- ?sp-chg-pos 20) 70
    =(- ?peak-freq 200) =(+ ?peak-freq 200)
    (5 3) ?*vowel-end-min-change-before-h*)

  (case
    ((power-decrease (?pos1&~NONE ?change-decrease1&:(> ?change-decrease1 ?change-decrease))
      after ?pos 40
      =(- ?peak-freq 200) =(+ ?peak-freq 200)
      (5 3) ?*vowel-end-min-change-before-h*)
      =>
      (bind ?h-pos ?pos1))
    (otherwise
      =>
      (bind ?h-pos ?pos)))

;;; check characteristics of phoneme /h/

;;; check characteristics of left vowel
  (power-strength =(- ?pos 20) ?pos 0 1000 ?vpw-0-1000)
  (CF (vowel-by-h 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
  (power-strength =(- ?pos 20) ?pos 1000 6000 ?vpw-1000-6000)
  (CF (vowel-by-h 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
  (power-strength =(- ?pos 20) ?pos 5000 6000 ?vpw-5000-6000)
  (CF (h-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
    ?CFvpr-0-1000-5000-6000&: mightbe-valid)

  ?X<-(CF (left-context ?segment vowel) ?CFleft-vowel)
  =>
  (assert (applied ?segment sg-h-left-vowel-1))
  (hypothesize
    (assert (prop ?segment (proceeding-vowel-end-time ?h-pos)))
    (retract ?X)
    (assert (CF (left-context ?segment vowel)
      =(CFand (CFcomb ?CFleft-vowel
        (CFweight ?*strong-evidence* ?CFsp-change)
        (CFweight ?*weak-evidence* ?CFvpw-0-1000)
        (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
        (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
        ?CFsp-change
        )))))

(defrule sg-h-left-vowel-2
  "Find time of the vowel by /h/ transition."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (left-context ?segment vowel)

```

```

(not (applied ?segment sg-h-left-vowel-2))
(prop ?segment (candidate-loc ?from ?to))

;;; search segmenetation position of left vowel
(power-decrease (?vowel-end&~NONE ?change-decrease-0-4000)
  before ?from 200
  0 4000 (11 11) ?*vowel-end-min-change-before-h*)
(spectral-change (?sp-chg-pos&~NONE ?value)
  before =(+ ?vowel-end 20) 40
  0 4000 ( 9 5) 7.5 0.0 0.1 2.8)
(CF (h-spectral-change-left ?value) ?CFsp-change&: mightbe-valid)

(spectrum-peak (?peak-freq&~NONE&:(?peak-freq >= 0)&:(?peak-freq <= 4000)
  ?peak-amp ?peak-Q)
  at =(- ?sp-chg-pos 10) ?sp-chg-pos
  ?*spectrum-peak-smoothing-for-phoneme-h*)
(not (spectrum-peak (?pk-fq&~NONE&:(?pk-fq >= 0)&:(?pk-fq <= 4000)
  ?another-peak-amp:(> ?another-peak-amp ?peak-amp) ?pk-Q)
  at =(- ?sp-chg-pos 10) ?sp-chg-pos
  ?*spectrum-peak-smoothing-for-phoneme-h*))
(power-decrease (?pos&~NONE ?change-decrease)
  after =(- ?sp-chg-pos 20) 70
  =(- ?peak-freq 200) =(+ ?peak-freq 200)
  (5 3) ?*vowel-end-min-change-before-h*)

(power-increase (?h-pos&~NONE ?change-increase-2000-6000)
  after ?sp-chg-pos 70
  2000 6000 (5 5) ?*h-fricative-start-change-2000-6000*)
(CF (h-2000-6000-power-increase ?change-increase-2000-6000)
  ?CFh-2000-6000-pw-inc&: mightbe-valid)

;;; check charasterictics of left vowel
(power-strength =(- ?pos 20) ?pos 0 1000 ?vpw-0-1000)
(CF (vowel-by-h 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength =(- ?pos 20) ?pos 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-h 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
(power-strength =(- ?pos 20) ?pos 5000 6000 ?vpw-5000-6000)
(CF (h-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&: mightbe-valid)

?x<-(CF (left-context ?segment vowel) ?CFleft-vowel)
=>
(bind ?CFleft-value (CFcomb ?CFsp-change ?CFh-2000-6000-pw-inc))
(assert (applied ?segment sg-h-left-vowel-2))
(hypothesize
  (assert (prop ?segment (proceeding-vowel-end-time ?h-pos)))
  (retract ?x)
  (assert (CF (left-context ?segment vowel)
    =(CFand (CFcomb ?CFleft-vowel
      (CFweight ?*strong-evidence* ?CFsp-change)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000)
      (CFweight ?*evidence* ?CFh-2000-6000-pw-inc))
    ?CFleft-value
  )))))

(defrule sg-h-left-vowel-3
  "Find time of the vowel by /h/ transition."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (left-context ?segment vowel)
  (not (applied ?segment sg-h-left-vowel-3))
  (prop ?segment (candidate-loc ?from ?to))

;;; search segmenetation position of left vowel
(power-decrease (?vowel-end&~NONE ?change-decrease-0-4000)
  before ?from 200
  0 4000 (11 11) ?*vowel-end-min-change-before-h*)
(spectral-change (?sp-chg-pos&~NONE ?value)
  before =(+ ?vowel-end 20) 40
  0 4000 ( 9 5) 7.5 0.0 0.1 2.8)
(CF (h-spectral-change-left ?value) ?CFsp-change&: mightbe-valid)

```

```

(spectrum-peak (?peak-freq&~NONE&:(?peak-freq >= 0)&:(?peak-freq <= 4000)
  ?peak-amp ?peak-Q)
  at =(- ?sp-chg-pos 10) ?sp-chg-pos
  ?*spectrum-peak-smoothing-for-phoneme-h*)
(not (spectrum-peak (?pk-fq&~NONE&:(?pk-fq >= 0)&:(?pk-fq <= 4000)
  ?another-peak-amp&:(> ?another-peak-amp ?peak-amp) ?pk-Q)
  at =(- ?sp-chg-pos 10) ?sp-chg-pos
  ?*spectrum-peak-smoothing-for-phoneme-h*))
(power-decrease (?pos&~NONE ?change-decrease)
  after =(- ?sp-chg-pos 20) 70
  =(- ?peak-freq 200) =(+ ?peak-freq 200)
  (5 3) ?*vowel-end-min-change-before-h*)

(power-increase (?fric-pos&~NONE ?change-increase-2000-6000)
  after ?sp-chg-pos 70
  2000 6000 (5 5) ?*h-fricative-start-change-2000-6000*)
(CF (h-2000-6000-power-increase ?change-increase-2000-6000)
  ?CFh-2000-6000-pw-inc&: mightbe-valid)

(power-decrease (?h-pos&~NONE ?change-decrease-0-500)
  after ?sp-chg-pos =(- ?fric-pos ?sp-chg-pos)
  0 500 (5 3) ?*vowel-end-min-change-before-h*)

;;; check characteristics of left vowel
(power-strength =(- ?pos 20) ?pos 0 1000 ?vpw-0-1000)
(CF (vowel-by-h 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength =(- ?pos 20) ?pos 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-h 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
(power-strength =(- ?pos 20) ?pos 5000 6000 ?vpw-5000-6000)
(CF (h-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&: mightbe-valid)

?x<-(CF (left-context ?segment vowel) ?CFleft-vowel)
=>
(bind ?CFleft-value (CFcomb 0.01 ?CFsp-change ?CFh-2000-6000-pw-inc))
(assert (applied ?segment sg-h-left-vowel-3))
(hypothesize
  (assert (prop ?segment (preceding-vowel-end-time ?h-pos)))
  (retract ?x)
  (assert (CF (left-context ?segment vowel)
    =(CFand (CFcomb ?CFleft-vowel
      (CFweight ?*strong-evidence* ?CFsp-change)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000)
      (CFweight ?*evidence* ?CFh-2000-6000-pw-inc))
    ?CFleft-value
  )))))

(defcontradiction 1st-or-2nd-peaks-h-left ""
  (prop ?segment (preceding-vowel-end-time ?pos))
  (prop ?segment (preceding-vowel-end-time ~?pos)))

(defrule sg-h-left-vowel-done-1
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (CF (category ?segment h) ?CF-category&: mightbe-valid)
  (left-context ?segment vowel)
  (CF (left-context ?segment vowel) ?CF-context&: mightbe-valid)
  (not (applied ?segment sg-h-left-vowel-done-1))
  (prop ?segment (preceding-vowel-end-time ?pos))
  =>
  (assert (applied ?segment sg-h-left-vowel-done-1))
  (hypothesize
    (assert (start-time ?segment ?pos))
    (assert (CF (start-time ?segment ?pos) =(CFvalue ?CF-context))))))

;;;-----

```



LM17:>hatazaki>sre>seg-h>seg-h-right. art. 14

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/02/89 21:09:39

Queued at: 3/28/89 11:23:36

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-h-right
  "Hypothesizes right phoneme."
  (declare (salience ?*right-segmentation*))
  (viewpoint ?v
    (segment-status ?segment segmentation)
    (category ?segment h)
    (CF (category ?segment h) ?CF-category&: mightbe-valid))
  ;;;;
  (viewpoint ?
    (exists (start-time ?segment ?)))
  ;;;;
  (not (applied ?segment sg-h-right))
  (not (end-time ?segment ?))
  =>
  (at ?v
    (assert (applied ?segment sg-h-right))
    (conflicting-alternatives
      (hypothesize
        (assert (right-context ?segment silence))
        (assert (CF (right-context ?segment silence) =(CFunknown))))
      (hypothesize
        (assert (right-context ?segment vowel))
        (assert (CF (right-context ?segment vowel) =(CFunknown))))
      (hypothesize
        (assert (right-context ?segment fricative))
        (assert (CF (right-context ?segment fricative) =(CFunknown))))
    )))
```

```
-----
```

```

(defrule sg-h-right-silence-1
  "Find time of the silence by /h/ transition."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (right-context ?segment silence)
  (not (applied ?segment sg-h-right-silence-1))
  (prop ?segment (candidate-loc ?from ?to))

;;; search segmenetation position of right silence
(power-end ?power-end~NONE after ?to 150 0 6000 ?*h-right-is-silence-power*)
(power-increase (?vowel-start&:(eq ?vowel-start 'NONE) ?change-increase)
  after ?to =(abs (- ?power-end ?to))
  0 4000 (11 11) ?*vowel-start-min-change-after-h*)

(spectrum-peak (?peak-freq~NONE&:(?peak-freq >= 1000)&:(?peak-freq <= 6000)
  ?peak-amp ?peak-Q)
  at =(- ?power-end 10) ?power-end
  ?*spectrum-peak-smoothing-for-phoneme-h*)
(not (spectrum-peak (?pk-fq~NONE&:(?pk-fq >= 1000)&:(?pk-fq <= 6000)
  ?another-peak-amp&:(> ?another-peak-amp ?peak-amp) ?pk-Q)
  at =(- ?power-end 10) ?power-end
  ?*spectrum-peak-smoothing-for-phoneme-h*))

(power-end ?h-end~NONE&:(< (abs (- ?h-end ?power-end)) 50)
  after =(- ?power-end 50) 100
  =(- ?peak-freq 200) =(+ ?peak-freq 200) ?*h-right-is-silence-power*)

;;; check charasterictics of right silence
(power-strength ?h-end =(+ ?h-end 30) 0 6000 ?spw-0-6000)
(CF (h-power-is-silence-closure ?spw-0-6000) ?CFspw-0-6000&: mightbe-valid)

;;; check charasteritics of /h/

?x<-(CF (right-context ?segment silence) ?CFright-silence)
=>
(retract ?x)
(assert (applied ?segment sg-h-right-silence-1))
(assert (prop ?segment (following-silence-start-time ?power-end)))
(assert (CF (right-context ?segment silence)
  =(CFand
    (CFcomb ?CFright-silence
      (CFweight ?*evidence* ?CFspw-0-6000))
    ?CFspw-0-6000))))

(defrule sg-h-right-silence-done-1
  "
  (declare (salience ?*right-segmentation-end*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (CF (category ?segment h) ?CF-category&: mightbe-valid)
  (right-context ?segment silence)
  (CF (right-context ?segment silence) ?CF-context&: mightbe-valid)
  (not (applied ?segment sg-h-right-silence-done-1))
  (prop ?segment (following-silence-start-time ?pos))
  =>
  (assert (applied ?segment sg-h-right-silence-done-1))
  (hypothesize
    (assert (end-time ?segment ?pos))
    (assert (CF (end-time ?segment ?pos) =(CFvalue ?CF-context))))))

;;; -----

```



```

(defrule sg-h-right-vowel-1
  "Find time of the vowel by /h/ transition."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (right-context ?segment vowel)
  (not (applied ?segment sg-h-right-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))

;;; search segmenetation position of right vowel
(power-increase (?vowel-start&~NONE ?change-increase-0-4000)
  after ?to 200
  0 4000 (11 11) ?*vowel-start-min-change-after-h*)
(spectral-change (?sp-chg-pos&~NONE ?value)
  after =(- ?vowel-start 20) 40
  0 4000 ( 9 5) 7.5 0.0 0.1 2.8)
(CF (h-spectral-change-right ?value) ?CFsp-change&: mightbe-valid)

(spectrum-peak (?peak-freq&~NONE&:(?peak-freq >= 0)&:(?peak-freq <= 4000)
  ?peak-amp ?peak-Q)
  at ?sp-chg-pos =(+ ?sp-chg-pos 10)
  ?*spectrum-peak-smoothing-for-phoneme-h*)
(not (spectrum-peak (?pk-fq&~NONE&:(?pk-fq >= 0)&:(?pk-fq <= 4000)
  ?another-peak-amp&:(> ?another-peak-amp ?peak-amp) ?pk-Q)
  at ?sp-chg-pos =(+ ?sp-chg-pos 10)
  ?*spectrum-peak-smoothing-for-phoneme-h*))
(power-increase (?pos&~NONE ?change-increase)
  before =(+ ?sp-chg-pos 20) 70
  =(- ?peak-freq 200) =(+ ?peak-freq 200)
  (5 3) ?*vowel-start-min-change-after-h*)

(case
  ((power-increase (?pos1&~NONE ?change-increase1&:(> ?change-increase1 ?change-increase))
    before ?pos 30
    =(- ?peak-freq 200) =(+ ?peak-freq 200)
    (5 3) ?*vowel-start-min-change-after-h*)
    =>
  (bind ?h-pos ?pos1))
  (otherwise
    =>
  (bind ?h-pos ?pos)))

;;; check charasteristics of phoneme /h/

;;; check charasterictics of right vowel
(power-strength ?pos =(+ ?pos 20) 0 1000 ?vpw-0-1000)
(CF (vowel-by-h 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength ?pos =(+ ?pos 20) 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-h 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
(power-strength ?pos =(+ ?pos 20) 5000 6000 ?vpw-5000-6000)
(CF (h-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&: mightbe-valid)

?x<-(CF (right-context ?segment vowel) ?CFright-vowel)
=>
(assert (applied ?segment sg-h-right-vowel-1))
(hypothesize
  (assert (prop ?segment (following-vowel-start-time ?h-pos)))
  (retract ?x)
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFright-vowel
      (CFweight ?*strong-evidence* ?CFsp-change)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
      ?CFsp-change
      )))))

(defrule sg-h-right-vowel-2
  "Find time of the vowel by /h/ transition."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)

```

```

(category ?segment h)
(right-context ?segment vowel)
(not (applied ?segment sg-h-right-vowel-2))
(prop ?segment (candidate-loc ?from ?to))

;;; search segmenetation position of right vowel
(power-increase (?vowel-start&~NONE ?change-increase-0-4000)
  after ?to 200
    0 4000 (11 11) ?*vowel-start-min-change-after-h*)
(spectral-change (?sp-chg-pos&~NONE ?value)
  after =(- ?vowel-start 20) 40
    0 4000 ( 9 5) 7.5 0.0 0.1 2.8)
(CF (h-spectral-change-right ?value) ?CFsp-change&: mightbe-valid)

(spectrum-peak (?peak-freq&~NONE&:(?peak-freq >= 0)&:(?peak-freq <= 4000)
  ?peak-amp ?peak-Q)
  at ?sp-chg-pos =(+ ?sp-chg-pos 10)
  ?*spectrum-peak-smoothing-for-phoneme-h*)
(not (spectrum-peak (?pk-fq&~NONE&:(?pk-fq >= 0)&:(?pk-fq <= 4000)
  ?another-peak-amp&:(> ?another-peak-amp ?peak-amp) ?pk-Q)
  at ?sp-chg-pos =(+ ?sp-chg-pos 10)
  ?*spectrum-peak-smoothing-for-phoneme-h*))
(power-increase (?pos&~NONE ?change-increase)
  before =(+ ?sp-chg-pos 20) 70
  =(- ?peak-freq 200) =(+ ?peak-freq 200)
  (5 3) ?*vowel-start-min-change-after-h*)

(power-increase (?pos1&~NONE ?change-increase1&:(> ?change-increase1 ?change-increase))
  before ?pos 30
  =(- ?peak-freq 200) =(+ ?peak-freq 200)
  (5 3) ?*vowel-start-min-change-after-h*)

(power-increase (?h-pos&~NONE ?change-increase-0-500)
  before ?sp-chg-pos =(- ?sp-chg-pos ?pos1)
  0 500 (5 3) ?*vowel-end-min-change-before-h*)

;;; check characteristics of phoneme /h/

;;; check characteristics of right vowel
(power-strength ?pos =(+ ?pos 20) 0 1000 ?vpw-0-1000)
(CF (vowel-by-h 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength ?pos =(+ ?pos 20) 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-h 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
(power-strength ?pos =(+ ?pos 20) 5000 6000 ?vpw-5000-6000)
(CF (h-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&: mightbe-valid)

?x<-(CF (right-context ?segment vowel) ?CFright-vowel)
=>
(bind ?CFright-value (CFcomb 0.01 ?CFsp-change))
(assert (applied ?segment sg-h-right-vowel-2))
(hypothesize
  (assert (prop ?segment (following-vowel-start-time ?h-pos)))
  (retract ?x)
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFright-vowel
      (CFweight ?*strong-evidence* ?CFsp-change)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
    ?CFright-value
  )))))

(defcontradiction 1st-or-2nd-peaks-h-right ""
  (prop ?segment (following-vowel-start-time ?pos))
  (prop ?segment (following-vowel-start-time ~?pos)))

(defrule sg-h-right-vowel-done-1
  ""
  (declare (salience ?*right-segmentation-end*))
  (segment-status ?segment segmentation)
  (category ?segment h)
  (CF (category ?segment h) ?CF-category&: mightbe-valid)
  (right-context ?segment vowel)
  (CF (right-context ?segment vowel) ?CF-context&: mightbe-valid)
  (not (applied ?segment sg-h-right-vowel-done-1))

```

```
(prop ?segment (following-vowel-start-time ?pos))
=>
(assert (applied ?segment sg-h-right-vowel-done-1))
(hypothesize
  (assert (end-time ?segment ?pos))
  (assert (CF (end-time ?segment ?pos) =(CFvalue ?CF-context))))
;;-----
```



LM17: >hatazaki>sre>seg-h>seg-h-end. art. 8

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 3/08/89 13:12:53

Queued at: 3/28/89 11:24:24

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-h-end-conflicting-with-uvstop-1
  "Solve confliction with unvoiced stop"
  (declare (salience ?*after-end-segmentation*))
  (viewpoint ?v
    (join
      (segment-status ?segment1 segmentation-ended)
      (not (applied ?segment1 sg-h-end-conflicting-with-uvstop-1))
      (join
        (category ?segment1 h)
        (CF (category ?segment1 h) ?CFcategory1&:mightbe-valid))
      ;; (join
      ;; (left-context ?segment1 vowel)
      ;; (CF (left-context ?segment1 vowel) ?CFcontext&:mightbe-valid))
      (join
        (segment-result ?segment1 (?start1 ?end1))
        ?x <- (CF (segment-result ?segment1 (?start1 ?end1)) ?CFsegment1))))
  (viewpoint ?
    (join
      (segment-status ?segment2 segmentation-ended)
      (join
        (category ?segment2 unvoiced-stop)
        (CF (category ?segment2 unvoiced-stop) ?CFcategory2&:mightbe-valid))
      (join
        (segment-result ?segment2 (?start2 ?end2))
        (CF (segment-result ?segment2 (?start2 ?end2)) ?CFsegment2))))
  (test (or (and (< ?start1 (/ (+ ?start2 ?end2) 2))
    (< (/ (+ ?start2 ?end2) 2) ?end1))
    (and (< ?start2 (/ (+ ?start1 ?end1) 2))
    (< (/ (+ ?start1 ?end1) 2) ?end2))))
  =>
  (at ?v
    (assert (applied ?segment1 sg-h-end-conflicting-with-uvstop-1))
    (retract ?x)
    (assert (CF (segment-result ?segment1 (?start1 ?end1))
      =(CFcomb ?CFsegment1 -0.5))))))
```

LM17: >hatazaki>sre>seg-voiced-fricative>seg-voiced-fricative. art. 21

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 12/02/88 10:01:03

Queued at: 3/28/89 10:53:07

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrelation VOICED-FRICATIVE-CANDIDATE (?time ?peak-value)
  implicit
  (syntax output
    (Voiced fricative candidate at ?time with ?peak-value)))

(defrule sg-vfric-0 ; Segmentation of voiced fricative.
  "Finds voiced friactive (/z/) candidates."
  (declare (salience ?*find-candidate*))
  (segmenting voiced-fricative)
  (VOICED-FRICATIVE-CANDIDATE ?time ?value)
  (CF (voiced-fricative high-middle-freq-power-ratio-peak ?value)
    ?CF-High-Middle-PR&:mightbe-valid)
  (power-strength =(- ?time 20) =(+ ?time 20) 0 500 ?0-500-P)
  (CF (voiced-fricative vocing-power ?0-500-P) ?CF0-500-P&:mightbe-valid)
  (power-strength =(- ?time 20) =(+ ?time 20) 1000 2000 ?1000-2000-P)
  (CF (voiced-fricative 1000-2000-power ?1000-2000-P) ?CF1000-2000-P&:mightbe-valid)
  (CF (voiced-fricative 0-500-1000-2000-power-ratio =(- ?0-500-P ?1000-2000-P)
    ?CF0-500-1000-2000-PR)
    ?CF0-500-1000-2000-PR)
  (BB ?BB)
  =>
  (bind ?segment (gensym "SEG-"))
  (at ?BB
    (assert (schema ?segment (instance-of phoneme-segment)))
    (hypothesize
      (assert (category ?segment voiced-fricative))
      ;;
      ;;[hatazaki 167]
      ;;      =(CFand (CFcomb (CFunknown)
      ;;                    (CFweight ?*weak-evidence*
      ;;                    (CFweight ?CF-High-Middle-PR)
      ;;                    (CFweight ?CF0-500-P)
      ;;                    (CFweight ?CF1000-2000-P)
      ;;                    (CFweight ?CF0-500-1000-2000-PR))
      ;;                    ?CF-High-Middle-PR
      ;;                    ?CF0-500-P
      ;;                    ?CF1000-2000-P
      ;;                    ?CF0-500-1000-2000-PR)
      ;;      <--      =(CFweight ?*strong-evidence*
      ;;                    (CFand ?CF-High-Middle-PR
      ;;                    ?CF0-500-P
      ;;                    ?CF1000-2000-P
      ;;                    ?CF0-500-1000-2000-PR))
      ;;
      (assert (prop ?segment (candidate-loc ?time ?time))
        (segment-status ?segment segmentation)
        (CF (category ?segment voiced-fricative)
          =(CFand (CFcomb (CFunknown)
            (CFweight ?*weak-evidence* ?CF-High-Middle-PR)
            (CFweight ?*weak-evidence* ?CF0-500-P)
            (CFweight ?*weak-evidence* ?CF1000-2000-P)
            (CFweight ?*weak-evidence* ?CF0-500-1000-2000-PR))
            ?CF-High-Middle-PR
            ?CF0-500-P
            ?CF1000-2000-P
            ?CF0-500-1000-2000-PR))))))
      ;;
      ;;-----
```



```

(defrule sg-vfric-bottom-up-1 ""
;;
;; - Narrow peak may be voiced stop /dz/, which need not to be detected
;; by this function. Then, smoothing factors can be (0, 15). Changed
;; to (0, 15) from (11, 5) again.
;; [hatazaki 7/28/88 13:16:52]
;; - Smoothing factors were changed so that narrow peak can be detected.
;; (cf. MAU_1_2372 470msec)
;; Old ...
;; (smoothing-size 0)
;; (moving-ave-size 15) ; Changed from 19, since some peaks cannot be
;; ; found with 19 and change-allowance=5.
;; ; (cf. MAU_1_1466 /DJ/) [hatazaki 7/27/88 15:19:14]
;;
(declare (salience ?*top-down*))
(goal (VOICED-FRICATIVE-CANDIDATE ?time&?? ?peak-val&??))
(not (explicit (VOICED-FRICATIVE-CANDIDATE ?time ?peak-val)))
(BB ?BB)
=>
(at ?BB
  (for PEAK in (caddr (#USER::call-speechfunc
    (list "POWER-RATIO-PEAKS"
          0 1500 ; Start, end time.
          5000 6000 ; Frequency region A.
          1000 3000 ; Frequency region B.
          0 ; Lower bound.
          5 ; Change allowance.
          0 ; Smoothing size.
          15 ; Moving average size.
          )))
    do (assert (VOICED-FRICATIVE-CANDIDATE =(car PEAK) =(cadr PEAK))))))
;;-----

```



LM17: >hatazaki>sre>seg-voiced-fricative>seg-voiced-fricative-left. art. 26

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 12/09/88 16:16:02

Queued at: 3/28/89 10:54:00

```
;;;-*- Mode:ART; Package:art-user; Base:10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-vfric-left ;voiced-fricative-segmentation-left
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&:mightbe-valid))
  (not (applied ?segment sg-vfric-left))
  (not (start-time ?segment ?))
  =>
  (assert (applied ?segment sg-vfric-left))
  (conflicting-alternatives
    (hypothesize
      (assert (left-context ?segment word-initial))
      (assert (CF (left-context ?segment word-initial) =(CFunknown))))))
  (hypothesize
    (assert (left-context ?segment vowel))
    (assert (CF (left-context ?segment vowel) =(CFunknown))))))
```

```
;;;-----
```

```

(defrule sg-vfric-aft-vowel-1
  "Find time the preceding vowel ends."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment vowel)
    ?x <- (CF (left-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-vfric-aft-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))
  ;;
  ;; - Search 150msec before ?to. (cf. MAU_1_0172 /DJ/) [hatazaki 7/27/88 12:08:26]
  (power-decrease (?here&~NONE ?change) before =(+ ?to 20) 200 1000 3000
    ?*normal-smoothing-size* ?*vowel-end-change-lower-before-voiced-fricative*)
  (power-strength =(- ?here 30) ?here 0 500 ?power)
  (CF (vowel 0-500-power ?power) ?CF0-500-P)
  =>
  (assert (prop ?segment (preceding-vowel-end-time ?here)))
  (retract ?x)
  ;;
  ;;[hatazaki 167]
  ;;      =(CFand (CFcomb ?CFcontext
  ;;              (CFweight ?*strong-evidence* ?CF0-500-P))
  ;;      ?CF0-500-P)
  ;; <--      =(CFcomb ?CFcontext
  ;;              (CFweight ?*strong-evidence* ?CF0-500-P))
  ;;
  (assert (CF (left-context ?segment vowel)
    =(CFand (CFcomb ?CFcontext
      (CFweight ?*strong-evidence* ?CF0-500-P))
      ?CF0-500-P)))
  (assert (applied ?segment sg-vfric-aft-vowel-1)))

(defrule sg-vfric-aft-vowel-2
  "Find a voiced closure."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment vowel)
    ?x <- (CF (left-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-vfric-aft-vowel-2))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (preceding-vowel-end-time ?vowel-end))
  (power-increase (?cl-end&~NONE ?cl-end-change)
    before = ?to =(- ?from ?to -100) 1000 6000
    ?*normal-smoothing-size*
    ?*default-min-change*)
  (test (> (- ?cl-end ?vowel-end) -20))
  (power-decrease (?cl-start&~NONE ?cl-start-change)
    before ?cl-end 100 1000 6000
    ?*normal-smoothing-size*
    ?*default-min-change*)
  =>
  (assert (prop ?segment (closure ?cl-start ?cl-end)))
  (assert (applied ?segment sg-vfric-aft-vowel-2)))

(defrule sg-vfric-aft-vowel-done-1
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment vowel)
    (CF (left-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-vfric-aft-vowel-done-1))
  (case
    ((prop ?segment (closure ?start ?cl-end)))

```

```
(otherwise
  (prop ?segment (preceding-vowel-end-time ?start)))
=>
(assert (applied ?segment sg-vfric-aft-vowel-done-1))
(hypothesize
  (assert (start-time ?segment ?start))
  (assert (CF (start-time ?segment ?start) =(CFvalue ?CFcontext))))
;;-----
```

```

(defrule sg-vfric-aft-word-initial-0
  "Has burst or not."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)
    (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-vfric-aft-word-initial-0))
  (prop ?segment (candidate-loc ?from ?to))
  (case
    ((and (power-decrease (?burst-end&~NONE ?bu-end-change)
      before =(+ ?to 20) 200 1000 3000
      ?*dj-smoothing-for-finding-burst*
      ?*dj-burst-end-min-change*)
      (power-increase (?burst-start&~NONE ?bu-start-change)
        before ?burst-end 30 1000 3000
        ?*dj-smoothing-for-finding-burst*
        ?*dj-burst-start-min-change*))
      =>
      (assert (prop ?segment (has-burst yes))
        (prop ?segment (burst ?burst-start ?burst-end))))
    (otherwise
      =>
      (assert (prop ?segment (has-burst no))))))
  =>
  (assert (applied ?segment sg-vfric-aft-word-initial-0)))

(defrule sg-vfric-aft-word-initial-1a
  "Find time the silence ends."
  ;;[hatazaki 101]
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)
    (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-vfric-aft-word-initial-1a))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (has-burst no))
  (power-decrease (?x&:(eq ?x 'NONE) ?change)
    before =(+ ?to 20) 200 1000 3000
    ?*normal-smoothing-size*
    ?*vowel-end-change-lower-before-voiced-fricative*)
  (power-start ?power-start&~NONE before =(+ ?to 20) 200 0 6000 ?*silence-power-upper*)
  (power-start ?low-power-start&~NONE before =(+ ?to 20) 200 0 500 ?*silence-power-upper*)
  =>
  (assert (applied ?segment sg-vfric-aft-word-initial-1a))
  (assert (prop ?segment (silence-end-time =(min ?power-start ?low-power-start))))))

(defrule sg-vfric-aft-word-initial-1c
  "Find time the silence ends."
  ;;[hatazaki 101]
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)
    (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-vfric-aft-word-initial-1a))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (has-burst yes))
  (prop ?segment (burst ?bu-start ?bu-end))
  ;;
  ;;[hatazaki 174] Checks there is no power before here.
  ;;
  (power-decrease (?x&:(eq ?x 'NONE) ?change)
    before ?bu-start 200 1000 3000

```

```

                ?*normal-smoothing-size*
                ?*vowel-end-change-lower-before-voiced-fricative*)
(power-start ?power-start&~NONE before ?bu-end 200 0 6000 ?*silence-power-upper*)
(power-start ?low-power-start&~NONE before ?bu-end 200 0 500 ?*silence-power-upper*)
=>
(assert (applied ?segment sg-vfric-aft-word-initial-1a))
(assert (prop ?segment (silence-end-time =(min ?power-start ?low-power-start))))))

(defrule sg-vfric-aft-word-initial-1b
  "Find time the power starts."
  ;;[hatazaki 101]
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)
    (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-vfric-aft-word-initial-1b))
  (prop ?segment (silence-end-time ?silence-end))
  (power-increase (?power-start&~NONE ?) after =(- ?silence-end 50) 200 0 6000
    ?*normal-smoothing-size* ?*z-after-silence-power-change-lower*)
  (power-increase (?low-power-start&~NONE ?) after =(- ?silence-end 50) 200 0 500
    ?*normal-smoothing-size* ?*z-after-silence-power-change-lower*)
  =>
  (assert (applied ?segment sg-vfric-aft-word-initial-1b))
  (assert (prop ?segment (power-start-time =(min ?power-start ?low-power-start))))))

(defrule sg-vfric-aft-word-initial-2
  "Checks if here is really at the beginning of word."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (left-context ?segment word-initial)
    ?x <- (CF (left-context ?segment word-initial) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-vfric-aft-word-initial-2))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (power-start-time ?power-start))
  (case
    ;;
    ;;[hatazaki 167]
    ;;   =(CFand (CFcomb ?CFcontext
    ;;             (CFweight ?*strong-evidence* 0.9)
    ;;             0.9)
    ;;   <-- =(CFcomb ?CFcontext (CFweight ?*strong-evidence* 0.9))
    ;;
    ;;[hatazaki 154] Simplify the checks of word initial.
    ;; (power-start ?y&(eq ?y 'NONE) before ?power-start 500 0 6000)
    ;; <--
    ;;   ;; Next condition for ?y should be ?y&(eq ?y 'NONE), not ?y&NONE.
    ;;   (and (power-decrease (?vowel-end&(eq ?vowel-end 'NONE) 0) before ?power-start 200
    ;;         1000 3000
    ;;         ?*normal-smoothing-size*
    ;;         ?*vowel-end-change-lower-before-voiced-fricative*)
    ;;   (power-strength =(- ?power-start 30) =(- ?power-start 10) 0 6000
    ;;     ?silence-power&(?silence-power <= ?*silence-power-upper*))
    ;;   (power-end ?y&(eq ?y 'NONE) before ?power-start 500 0 6000
    ;;     ?*silence-power-upper*))
    ((power-start ?y&(eq ?y 'NONE) before ?power-start 500 0 6000
      ?*silence-power-upper*)
    =>
    (retract ?x)
    (assert (CF (left-context ?segment word-initial)
      =(CFand (CFcomb ?CFcontext
        (CFweight ?*strong-evidence* 0.9)
        0.9))))))
  (otherwise
    =>
    (retract ?x)
    (assert (CF (left-context ?segment word-initial)
      =(CFand (CFcomb ?CFcontext
        (CFweight ?*strong-evidence* -0.9)

```



```

                                -0.9))))))
=>
(assert (applied ?segment sg-vfric-aft-word-initial-2)))
(defrule sg-vfric-aft-word-initial-done-1
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&:mightbe-valid))
  (join
    (left-context ?segment word-initial)
    (CF (left-context ?segment word-initial) ?CFcontext&:mightbe-valid))
  (not (applied ?segment sg-vfric-aft-word-initial-done-1))
  (prop ?segment (power-start-time ?power-start))
  =>
  (assert (applied ?segment sg-vfric-aft-word-initial-done-1))
  (hypothesize
    (assert (start-time ?segment ?power-start))
    (assert (CF (start-time ?segment ?power-start) =(CFvalue ?CFcontext))))))
```

LM17: >hatazaki>sre>seg-voiced-fricative>seg-voiced-fricative-right. art. 1  
1

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 11/30/88 15:38:07

Queued at: 3/28/89 10:54:52

```
;;;-*- Mode:ART; Package:art-user; Base:10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-vfric-right
  "Hypothesizes right phoneme."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&:mightbe-valid))
  (not (applied ?segment sg-vfric-right))
  (not (end-time ?segment ?))
  =>
  (assert (applied ?segment sg-vfric-right))
  (conflicting-alternatives
    (hypothesize
      (assert (right-context ?segment vowel))
      (assert (CF (right-context ?segment vowel) =(CFunknown))))))
```

```
;;;-----
```

```

(defrule sg-vfric-bfr-vowel-1
  "Find time the following vowel starts."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment vowel)
    ?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-vfric-bfr-vowel-1))
  (prop ?segment(candidate-loc ?from ?to))
  (power-increase (?vowel-start&~NONE ?change)
    after =(- ?to 20) 150
    0 500
    ?*normal-smoothing-size*
    ?*vowel-start-min-change-after-voiced-fricative*)
  (power-strength =(+ ?vowel-start 7.5) =(+ ?vowel-start 22.5) 0 500 ?strength)
  (CF (vowel 0-500-power ?strength) ?CF0-500-P)
  =>
  (assert (prop ?segment (following-vowel-start ?vowel-start)))
  (retract ?x)
  ;;
  ;;[hatazaki 167]
  ;;      =(CFand (CFcomb ?CFcontext
  ;;              (CFweight ?*strong-evidence* ?CF0-500-P))
  ;;      ?CF0-500-P)
  ;; <--      =(CFcomb ?CFcontext
  ;;              (CFweight ?*strong-evidence* ?CF0-500-P))
  ;;
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFcontext
      (CFweight ?*strong-evidence* ?CF0-500-P))
      ?CF0-500-P)))
  (assert (applied ?segment sg-vfric-bfr-vowel-1)))

(defrule sg-vfric-bfr-vowel-done-1
  ""
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (join
    (category ?segment voiced-fricative)
    (CF (category ?segment voiced-fricative) ?CFcategory&: mightbe-valid))
  (join
    (right-context ?segment vowel)
    ?x <- (CF (right-context ?segment vowel) ?CFcontext&: mightbe-valid))
  (not (applied ?segment sg-vfric-bfr-vowel-done-1))
  (prop ?segment (following-vowel-start ?vowel-start))
  =>
  (assert (applied ?segment sg-vfric-bfr-vowel-done-1))
  (hypothesize
    (assert (end-time ?segment ?vowel-start))
    (assert (CF (end-time ?segment ?vowel-start) =(CFvalue ?CFcontext))))))

```

LM17: >hatazaki>SRE>SEG-VOICED-STOP>get-voiced-stop-peak-candidates.lisp.  
6

For: Kaichiro Hatazaki  
Printed on: LGP8-J ON ATR-LM01  
Number of copies: 1  
Data created at: 1/19/89 16:29:57  
Queued at: 3/28/89 11:20:03

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: ART-USER; Base: 10 -*-  
(defun get-voiced-stop-peak-candidates ()  
  (let ((voiced-stop-peak-candidates (cddr (USER::call-speechfunc  
    (list "POWER-RATIO-PEAKS"  
          0 2000  
          0 500  
          1000 6000  
          5 1.0  
          11 15))))))  
    (loop for peak in voiced-stop-peak-candidates  
          collect peak)))
```

LM17: >hatazaki>sre>seg-voiced-stop>seg-voiced-stop. art. 3

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/20/89 16:33:38

Queued at: 3/28/89 11:15:46

```
;;;-*- Mode:ART; Package:art-user; Base:10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-voiced-stop-0
  "Finds a candidate of Voiced-stop."
  (declare (salience ?*find-candidate*))
  (segmenting voiced-stop)
  (voiced-stop-peak-candidate ?peak-loc ?peak-value)
  (CF (voiced-stop-peak-candidate ?peak-value) ?CFvoiced-stop-peak-candidate:&:mightbe-valid)
  (BB ?BB)
  =>
  (bind ?segment (gensym "SEG-"))
  (at ?BB
    (assert (voiced-stop-candidate ?peak-loc ?peak-loc))
    (assert (schema ?segment (instance-of phoneme-segment)))
    (hypothesize
      (assert (category ?segment voiced-stop))
      (assert (prop ?segment (candidate-loc ?peak-loc ?peak-loc))
        (segment-status ?segment segmentation)
        (CF (category ?segment voiced-stop)
          =(CFand (CFcomb
            (CFunknown)
            (CFweight ?*strong-evidence* ?CFvoiced-stop-peak-candidate)
            ?CFvoiced-stop-peak-candidate)))))))
```

```
;;;-----
(defrule voiced-stop-pos-no-buzz-head-in-word
  "FIND CANDIDATE OF NO BUZZ-BAR IN WORD INITIAL"
  (declare (salience ?*find-candidate*))
  (segmenting voiced-stop)
  (not (applied ?segment voiced-stop-pos-no-buzz-head-in-word))
  (power-start ?power-start&~NONE
    after 0 2000 0 500 ?*silence-power-upper*)
  (power-strength =(- ?power-start 50) ?power-start 0 6000 ?pw-0-6000)
  (CF (voiced-stop-power-is-silence ?pw-0-6000) ?CFpw-0-6000:&:mightbe-valid)
  (power-increase (?vowel-start&~NONE&:(>= ?vowel-start ?power-start) ?change-increase)
    after =(- ?power-start 10) 40 0 6000 (11 11)
    ?*vowel-start-min-change-after-voiced-stop*)
  (CF (voiced-stop-no-buzz-head-power-increase ?change-increase) ?CFpw-increase)
  (BB ?BB)
  =>
  (bind ?segment (gensym "SEG-"))
  (at ?BB
    (assert (voiced-stop-candidate ?power-start ?power-start))
    (assert (schema ?segment (instance-of phoneme-segment)))
    (hypothesize
      (assert (category ?segment voiced-stop))
      (assert (prop ?segment (phoneme-pos-in-word no-buzz-head)))
      (assert (CF (prop ?segment (phoneme-pos-in-word no-buzz-head))
        =(CFcomb (CFunknown) 1.0)))
      (assert (prop ?segment (candidate-loc ?power-start ?power-start))
        (segment-status ?segment segmentation)
        (CF (category ?segment voiced-stop)
          =(CFand (CFcomb
            (CFunknown)
            (CFweight ?*strong-evidence* ?CFpw-increase)
            ?CFpw-increase)))))))
```

```
;;;-----
;;;(defrule get-voiced-stop-region-candidates
;;;  ""
;;;  (declare (salience ?*top-down*))
;;;  (goal (voiced-stop-region-candidate ?from&?? ?to&??))
;;;  (explicit (not (voiced-stop-region-candidate ?from ?to)))
;;;  =>
;;;  (bind ?voiced-stop-region-candidates
;;;    (get-voiced-stop-region-candidates))
;;;  (for region in ?voiced-stop-region-candidates
;;;    do (assert =(seq$ (cons 'voiced-stop-region-candidate
;;;      (list ?from ?to))))))
```

```
(defrule get-voiced-stop-peak-candidates
  (declare (salience ?*top-down*))
  (goal (voiced-stop-peak-candidate ?peak-loc&?? ?peak-value&??))
  (explicit (not (voiced-stop-peak-candidate ?peak-loc ?peak-value)))
```



```

=>
(bind ?voiced-stop-peak-candidates
 (get-voiced-stop-peak-candidates))
(for peak in ?voiced-stop-peak-candidates
 do (assert =(seq$ (cons 'voiced-stop-peak-candidate peak))))))

;;;-----
(defrelation search-voiced-stop-burst-after (?here ?range ?freq-min ?freq-max))

(defrule get-voiced-stop-burst-right-of-voiced-stop-0
  "Gets a voiced-stop-burst following the Voiced Stop.
  (BACKWARD CHAINING RULE)"
  (declare (salience ?*top-down*))
  (goal (voiced-stop-burst-exists ?from&?? ?to&?? ?freq&??
    after ?here&~?? ?range&~?? ?freq-min&~?? ?freq-max&~??
    ?CF&??))
  (explicit (not (voiced-stop-burst-exists ? ? ? after ?here ?range ?freq-min ?freq-max ?)))
  (BB ?BB)
  =>
  (at ?BB
    (assert (search-voiced-stop-burst-after ?here ?range ?freq-min ?freq-max))))

(defrule get-voiced-stop-burst-right-of-voiced-stop-1
  ""
  ;;
  ;; [hatazaki 143] Searches a voiced-stop-burst between ?freq-min and ?freq-max, which are speci
  fied in
  ;; (voiced-stop-burst-exists ?from&?? ?to&?? ?freq&??
  ;; after ?here&~?? ?range&~?? ?freq-min&~?? ?freq-max&~??
  ;; ?CF&??)
  ;;
  ;; [hatazaki 131] Uses ?*voiced-stop-burst-1000-6000-start-min-change* instead of
  ;; ?*voiced-stop-burst-start-min-change* for finding ?voiced-stop-burst-1000-6000-start.
  ;;
  ;; [hatazaki 127] Uses ?*smoothing-for-finding-voiced-stop-burst-end* (is no
  ;;smoothing) for finding voiced-stop-burst end.
  ;;
  ;; [hatazaki 119] Checks the voiced-stop-burst duration.
  ;;
  ;; [hatazaki 116] Modified get-voiced-stop-burst-right-of-voiced-stop-1 to check
  ;;power change at the spectral peaks for both voiced-stop-burst start and end
  ;;time. Power change at the voiced-stop-burst start time should be checked if this
  ;;is
  ;;enough large, to say it is a voiced-stop-burst. Power change at the voiced-stop-burst end
  ;;may not be so large (cf. MAU_B_0169), then it is OK even if the power
  ;;change at the voiced-stop-burst end is nearly 0.
  ;;
  (declare (salience ?*top-down*))
  (search-voiced-stop-burst-after ?here ?range ?freq-min ?freq-max)
  (power-increase (?voiced-stop-burst-1000-6000-start&~NONE
    ?voiced-stop-burst-1000-6000-start-change)
    after ?here ?range ?freq-min ?freq-max
    ?*smoothing-for-finding-voiced-stop-burst*
    ?*voiced-stop-burst-1000-6000-start-min-change*)
  (CF (voiced-stop burst-1000-6000-start-change
    ?voiced-stop-burst-1000-6000-start-change)
    ?CF-1000-6000-start&: mightbe-valid)
  (spectrum-peak (?peak-freq&~NONE&:(?peak-freq >= ?freq-min)&:(?peak-freq <= ?freq-max)
    ?peak-amp ?peak-Q)
    at ?voiced-stop-burst-1000-6000-start
    =(+ ?voiced-stop-burst-1000-6000-start 10)
    ?*spectrum-peak-smoothing-for-getting-voiced-stop-burst*)
  (power-increase (?voiced-stop-burst-start&~NONE ?voiced-stop-burst-start-change)
    after ?here ?range
    =(- ?peak-freq 100) =(+ ?peak-freq 100)
    ?*smoothing-for-finding-voiced-stop-burst*
    ?*voiced-stop-burst-start-min-change*)
  (CF (voiced-stop burst-start-change ?voiced-stop-burst-start-change)
    ?CF-start&: mightbe-valid)
  ;;
  ;;[hatazaki 150] ?voiced-stop-burst-end may not be found.
  ;; <--
  ;; (power-decrease (?voiced-stop-burst-end&~NONE ?voiced-stop-burst-end-change)
  ;; after ?voiced-stop-burst-start 50
  ;;

```

```

;;          =(- ?peak-freq 100) =(+ ?peak-freq 100)
;;          ?*smoothing-for-finding-voiced-stop-burst-end*
;;          ?*voiced-stop-burst-end-min-change*
;; (CF (voiced-stop burst-end-change ?voiced-stop-burst-end-change)
;;     ?CF-end&(?CF-end >= ?*mightbe-valid*))
;; (CF (voiced-stop burst-duration
;;     =(- ?voiced-stop-burst-end ?voiced-stop-burst-start))
;;     ?CFduration&(?CFduration >= ?*mightbe-valid*)))
;;
(case
  ((power-decrease (?voiced-stop-burst-end&~NONE ?voiced-stop-burst-end-change)
    after ?voiced-stop-burst-start&~NONE 15
    =(- ?peak-freq 100) =(+ ?peak-freq 100)
    ?*smoothing-for-finding-voiced-stop-burst-end* 0.0))
  (otherwise
    =>
    (bind ?voiced-stop-burst-end 'NONE)))
; (case
;   ((power-decrease (?voiced-stop-burst-end&~NONE ?voiced-stop-burst-end-change)
;     after ?voiced-stop-burst-start 30
;     =(- ?peak-freq 100) =(+ ?peak-freq 100)
;     ?*smoothing-for-finding-voiced-stop-burst-end*
;     ?*voiced-stop-burst-end-min-change*
;     (CF (voiced-stop burst-end-change ?voiced-stop-burst-end-change)
;         ?CF-end) ;;&:mightbe-valid)
;     (CF (voiced-stop burst-duration
;         =(- ?voiced-stop-burst-end ?voiced-stop-burst-start))
;         ?CFduration)) ;;&:mightbe-valid)
;   (otherwise
;     =>
;     (bind ?voiced-stop-burst-end 'NONE)
;     (bind ?CF-end ?*CFunknown*)
;     (bind ?CFduration ?*CFunknown*)))

(BB ?BB)
=>
;;
;;[hatazaki 150] =(CFcomb (CFweight ?*strong-evidence* ?CF-start)
;;                      (CFweight ?*weak-evidence* ?CF-end )
;;                      (CFweight ?*weak-evidence* ?CFduration))
;;
;;          <--
;;          =(CFand ?CF-start ?CF-end ?CFduration).
;;[hatazaki 135] Changed (CFand (CFcomb ?CF-start ?CF-end) ?CFduration) to
;;                      (CFand ?CF-start ?CF-end ?CFduration).
;;
; (at ?BB
;   (assert
;     (voiced-stop-burst-exists ?voiced-stop-burst-start ?voiced-stop-burst-end ?peak-freq
;       after ?here ?range ?freq-min ?freq-max
;       =(CFvalue (CFand (CFcomb (CFunknown)
;         (CFweight ?*strong-evidence* ?CF-start)
;         ?CF-start))))))
; (at ?BB
;   (assert
;     (voiced-stop-burst-exists ?voiced-stop-burst-start ?voiced-stop-burst-end ?peak-freq
;       after ?here ?range ?freq-min ?freq-max
;       =(CFvalue (CFand (CFcomb (CFunknown)
;         (CFweight ?*strong-evidence* ?CF-start)
;         (CFweight ?*weak-evidence* ?CF-end )
;         (CFweight ?*weak-evidence* ?CFduration))
;         ?CF-start))))))

```

LM17: >hatazaki>sre>seg-voiced-stop>seg-voiced-stop-pos-in-word. art. 29

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/23/89 11:50:20

Queued at: 3/28/89 11:16:36

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-voiced-stop-pos-in-word
  "Hypothesizes position phoneme in word."
  (declare (salience ?*start-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (not (applied ?segment sg-voiced-stop-pos-in-word))
  (not (prop ?segment (phoneme-pos-in-word no-buzz-head)))
  =>
  (assert (applied ?segment sg-voiced-stop-pos-in-word))
  (conflicting-alternatives
    (hypothesize
      (assert (prop ?segment (phoneme-pos-in-word head)))
      (assert (CF (prop ?segment (phoneme-pos-in-word head)) =(CFunknown))))
    (hypothesize
      (assert (prop ?segment (phoneme-pos-in-word middle)))
      (assert (CF (prop ?segment (phoneme-pos-in-word middle)) =(CFunknown))))))
```

```
;;;-----
```

```
(defrule voiced-stop-pos-head-in-word
  ""
  (declare (salience ?*start-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (not (applied ?segment voiced-stop-pos-head-in-word))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (phoneme-pos-in-word head))
  ?x <- (CF (prop ?segment (phoneme-pos-in-word head&~no-buzz-head)) ?)
  (power-start ?power-start&~NONE&:(>= 100 (- ?from ?power-start))
    after 0 2000 0 500 ?*silence-power-upper*)
  (power-strength =(- ?power-start 50) ?power-start 0 6000 ?pw-0-6000)
  (CF (voiced-stop-power-is-silence ?pw-0-6000) ?CFpw-0-6000&:mightbe-valid)
  (power-increase (?vowel-start&~NONE ?change-increase)
    after ?from 100 0 6000 (11 11)
    ?*vowel-start-min-change-after-voiced-stop* )
  (power-decrease (?vowel-end&:(eq ?vowel-end 'NONE) ?change-decrease)
    before ?vowel-start =(- ?vowel-start ?power-start) 3500 6000 (11 11)
    (?*vowel-end-min-change-before-voiced-stop* -60))
  =>
  (retract ?x)
  (assert (applied ?segment voiced-stop-pos-head-in-word))
  (assert (CF (prop ?segment (phoneme-pos-in-word head)) =(CFcomb (CFunknown) 1.0))))
```

```
(defrule voiced-stop-pos-middle-in-word
  ""
  (declare (salience ?*start-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (not (applied ?segment voiced-stop-pos-middle-in-word))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (phoneme-pos-in-word middle))
  ?x <- (CF (prop ?segment (phoneme-pos-in-word middle&~no-buzz-head)) ?)
  (power-increase (?vowel-start&~NONE ?change-increase)
    after ?from 200 0 6000 (11 11)
    (?*vowel-start-min-change-after-voiced-stop* -65))
  (power-decrease (?vowel-end&~NONE ?change-decrease)
    before ?to 200 3500 6000 (11 11)
    (?*vowel-end-min-change-before-voiced-stop* -65))
  =>
  (retract ?x)
  (assert (applied ?segment voiced-stop-pos-middle-in-word))
  (assert (CF (prop ?segment (phoneme-pos-in-word middle)) =(CFcomb (CFunknown) 1.0))))
```

LM17: >hatazaki>sre>seg-voiced-stop>seg-voiced-stop-left. art. 60

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/30/89 19:50:16

Queued at: 3/28/89 11:17:27

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-voiced-stop-left
  ""
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (CF (category ?segment voiced-stop) ?CF-category&:mightbe-valid)
  (not (prop ?segment (phoneme-pos-in-word no-buzz-head)))
  (not (applied ?segment sg-voiced-stop-left))
  (not (start-time ?segment ?))
  =>
  (assert (applied ?segment sg-voiced-stop-left))
  (conflicting-alternatives
   (hypothesize
    (assert (left-context ?segment silence))
    (assert (CF (left-context ?segment silence) =(CFunknown))))))
  (hypothesize
   (assert (left-context ?segment vowel))
   (assert (CF (left-context ?segment vowel) =(CFunknown))))))
```

```
(defrule sg-no-buzz-head-voiced-stop-left
  ""
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (CF (category ?segment voiced-stop) ?CF-category&:mightbe-valid)
  (prop ?segment (phoneme-pos-in-word no-buzz-head))
  (not (applied ?segment sg-no-buzz-head-voiced-stop-left))
  (not (start-time ?segment ?))
  =>
  (assert (applied ?segment sg-no-buzz-head-voiced-stop-left))
  (conflicting-alternatives
   (hypothesize
    (assert (left-context ?segment silence))
    (assert (CF (left-context ?segment silence) =(CFunknown))))))
```

```
;;;-----
```

```

(defrule sg-no-buzz-head-voiced-stop-left-silence-1
  "Find time of the silence of no buzz-bar voiced-stop transition."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (prop ?segment (phoneme-pos-in-word no-buzz-head))
  (left-context ?segment silence)
  (not (applied ?segment sg-no-buzz-head-voiced-stop-left-silence-1))
  (prop ?segment (candidate-loc ?from ?to))
  (CF (prop ?segment (phoneme-pos-in-word no-buzz-head)) ?CFno-buzz-head&: mightbe-valid)

;;; search segmenetation position of left silence
(power-start ?power-start&~NONE
  after 0 2000 0 500 ?*silence-power-upper*)

;;; check characteristics of left silence
(power-strength =(- ?power-start 50) ?power-start 0 6000 ?spw-0-6000)
(CF (voiced-stop-power-is-silence ?spw-0-6000) ?CFspw-0-6000&: mightbe-valid)

;;; check characteristics of voiced-stop
(power-strength ?power-start =(+ ?power-start 20) 0 500 ?vspw-0-500)
(CF (no-buzz-head-voiced-stop-0-500-power ?vspw-0-500) ?CFvspw-0-500&: mightbe-valid)
?x<-(CF (left-context ?segment silence) ?CFleft-silence)
=>
(retract ?x)
(assert (applied ?segment sg-no-buzz-head-voiced-stop-left-silence-1))
(assert (prop ?segment (proceeding-silence-end-time ?power-start)))
(assert (CF (left-context ?segment silence)
  =(CFand
    (CFcomb ?CFleft-silence
      (CFweight ?*weak-evidence* ?CFspw-0-6000)
      (CFweight ?*strong-evidence* ?CFvspw-0-500))
    ?CFspw-0-6000
    ?CFvspw-0-500))))))

(defrule sg-no-buzz-head-voiced-stop-left-silence-done-1
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (CF (category ?segment voiced-stop) ?CF-category&: mightbe-valid)
  (prop ?segment (phoneme-pos-in-word no-buzz-head))
  (left-context ?segment silence)
  (CF (left-context ?segment silence) ?CF-context&: mightbe-valid)
  (not (applied ?segment sg-no-buzz-head-voiced-stop-left-silence-done-1))
  (CF (prop ?segment (phoneme-pos-in-word no-buzz-head)
    ?CFno-buzz-head&:(CFequal ?CFno-buzz-head 1.0))
  (prop ?segment (proceeding-silence-end-time ?pos))
  =>
  (assert (applied ?segment sg-no-buzz-head-voiced-stop-left-silence-done-1))
  (hypothesize
    (assert (start-time ?segment ?pos))
    (assert (CF (start-time ?segment ?pos) =(CFvalue ?CF-context))))))

;;; -----

```

```

(defrule sg-head-voiced-stop-left-silence-1
  "Find time of the silence-voiced-stop transition."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (prop ?segment (phoneme-pos-in-word head))
  (not (prop ?segment (phoneme-pos-in-word no-buzz-head)))
  (left-context ?segment silence)
  (not (applied ?segment sg-head-voiced-stop-left-silence-1))
  (prop ?segment (candidate-loc ?from ?to))
  (CF (prop ?segment (phoneme-pos-in-word head)) ?CFhead&: mightbe-valid)

;;; search segmenetation position of left silence
(power-start ?power-start&~NONE after 0 2000 0 500 ?*silence-power-upper*)
; (power-start ?power-start&~NONE before =(+ ?from 10) 200 0 500 ?*silence-power-upper*)

;;; check charasterictics of left silence
(power-strength =(- ?power-start 50) ?power-start 0 6000 ?spw-0-6000)
(CF (voiced-stop-power-is-silence ?spw-0-6000) ?CFspw-0-6000&: mightbe-valid)

;;; check charasteritics of voiced-stop
(power-strength ?power-start =(+ ?power-start 30) 0 500 ?vspw-0-500)
(CF (voiced-stop-0-500-power ?vspw-0-500) ?CFvspw-0-500&: mightbe-valid)
(power-strength ?power-start =(+ ?power-start 30) 500 3500 ?vspw-500-3500)
(CF (voiced-stop-500-3500-power ?vspw-500-3500) ?CFvspw-500-3500&: mightbe-valid)
(CF (voiced-stop-power-ratio-0-500-500-3500 =(- ?vspw-0-500 ?vspw-500-3500))
  ?CFvspr-0-500-500-3500&: mightbe-valid)

?x<-(CF (left-context ?segment silence) ?CFleft-silence)
=>
(retract ?x)
(assert (applied ?segment sg-head-voiced-stop-left-silence-1))
(assert (prop ?segment (proceeding-silence-end-time ?power-start)))
(assert (CF (left-context ?segment silence)
  =(CFand
    (CFcomb ?CFleft-silence
      (CFweight ?*weak-evidence* ?CFspw-0-6000)
      (CFweight ?*weak-evidence* ?CFvspw-0-500)
      (CFweight ?*weak-evidence* ?CFvspw-500-3500)
      (CFweight ?*evidence* ?CFvspr-0-500-500-3500))
    ?CFspw-0-6000))))

(defrule sg-head-voiced-stop-left-silence-done-1
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (CF (category ?segment voiced-stop) ?CF-category&: mightbe-valid)
  (prop ?segment (phoneme-pos-in-word head))
  (not (prop ?segment (phoneme-pos-in-word no-buzz-head)))
  (left-context ?segment silence)
  (CF (left-context ?segment silence) ?CF-context&: mightbe-valid)
  (not (applied ?segment sg-head-voiced-stop-left-silence-done-1))
  (CF (prop ?segment (phoneme-pos-in-word head)) ?CFhead&:(CFequal ?CFhead 1.0))
  (prop ?segment (proceeding-silence-end-time ?pos))
=>
(assert (applied ?segment sg-head-voiced-stop-left-silence-done-1))
(hypothesize
  (assert (start-time ?segment ?pos))
  (assert (CF (start-time ?segment ?pos) =(CFvalue ?CF-context))))

;;;-----

```



```

(defrule sg-mid-voiced-stop-left-vowel-1
  "Find time the preceding vowel starts."
  (declare (saliency (+ ?*left-segmentation* 1)))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (prop ?segment (phoneme-pos-in-word middle))
  (not (prop ?segment (phoneme-pos-in-word no-buzz-head)))
  (left-context ?segment vowel)
  ?x<-(CF (left-context ?segment vowel) ?CFleft-vowel)
  (not (applied ?segment sg-mid-voiced-stop-left-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))

;;; search segmenetation position of left vowel
(power-decrease (?pos&-NONE ?vsp-decrease) before ?from 200 3500 6000 (11 11)
  (?*vowel-end-min-change-before-voiced-stop* -65))
(CF (mid-voiced-stop-power-decrease-left ?vsp-decrease) ?CFvsp-decrease&: mightbe-valid)

;;; check characteristics of voiced-stop
(power-strength ?pos =(+ ?pos 30) 0 500 ?vspw-0-500)
(CF (voiced-stop-0-500-power ?vspw-0-500) ?CFvspw-0-500&: mightbe-valid)
(power-strength ?pos =(+ ?pos 30) 500 3500 ?vspw-500-3500)
(CF (voiced-stop-500-3500-power ?vspw-500-3500) ?CFvspw-500-3500&: mightbe-valid)
(CF (voiced-stop-power-ratio-0-500-500-3500 =(- ?vspw-0-500 ?vspw-500-3500))
  ?CFvspr-0-500-500-3500&: mightbe-valid)

;;; check characteristics of left vowel
(power-strength =(- ?pos 30) ?pos 0 1000 ?vpw-0-1000)
(CF (vowel-by-voiced-stop 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength =(- ?pos 30) ?pos 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-voiced-stop 1000-6000-power ?vpw-1000-6000)
  ?CFvpw-1000-6000&: mightbe-valid)
(power-strength =(- ?pos 30) ?pos 5000 6000 ?vpw-5000-6000)
(CF (voiced-stop-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&: mightbe-valid)
=>
(assert (applied ?segment sg-mid-voiced-stop-left-vowel-1))
(hypothesize
  (assert (prop ?segment (preceding-vowel-end-time ?pos)))
  (retract ?x)
  (assert (CF (left-context ?segment vowel)
    =(CFand (CFcomb ?CFleft-vowel
      (CFweight ?*strong-evidence* ?CFvsp-decrease)
      (CFweight ?*weak-evidence* ?CFvspw-0-500)
      (CFweight ?*weak-evidence* ?CFvspw-500-3500)
      (CFweight ?*evidence* ?CFvspr-0-500-500-3500)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
    ?CFvsp-decrease))))))

(defrule sg-mid-voiced-stop-left-vowel-done-1
  ""
  (declare (saliency ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (CF (category ?segment voiced-stop) ?CF-category&: mightbe-valid)
  (prop ?segment (phoneme-pos-in-word middle))
  (not (prop ?segment (phoneme-pos-in-word no-buzz-head)))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&: mightbe-valid)
  (left-context ?segment vowel)
  (CF (left-context ?segment vowel) ?CF-context&: mightbe-valid)
  (not (applied ?segment sg-mid-voiced-stop-left-vowel-done-1))
  (prop ?segment (preceding-vowel-end-time ?pos))
=>
(assert (applied ?segment sg-mid-voiced-stop-left-vowel-done-1))
(hypothesize
  (assert (start-time ?segment ?pos))
  (assert (CF (start-time ?segment ?pos) =(CFvalue ?CF-context))))))

```

LM17: >hatazaki>sre>seg-voiced-stop>seg-voiced-stop-right. art. 60

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/02/89 15:52:40

Queued at: 3/28/89 11:18:14

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-voiced-stop-right
  "Hypothesizes right phoneme."
  (declare (salience ?*right-segmentation*))
  (viewpoint ?v
    (segment-status ?segment segmentation)
    (category ?segment voiced-stop)
    (CF (category ?segment voiced-stop) ?CF-category&: mightbe-valid))
  ;;;;
  (viewpoint ?
    (exists (start-time ?segment ?)))
  ;;;;
  (not (prop ?segment (phoneme-pos-in-word no-buzz-head)))
  (not (applied ?segment sg-voiced-stop-right))
  (not (end-time ?segment ?))
  =>
  (at ?v
    (assert (applied ?segment sg-voiced-stop-right))
    (conflicting-alternatives
      (hypothesize
        (assert (right-context ?segment vowel))
        (assert (CF (right-context ?segment vowel) =(CFunknown)))))))
```

```
-----
;;;
(defrule sg-no-buzz-head-voiced-stop-right
  "Hypothesizes right phoneme."
  (declare (salience ?*right-segmentation*))
  (viewpoint ?v
    (segment-status ?segment segmentation)
    (category ?segment voiced-stop)
    (CF (category ?segment voiced-stop) ?CF-category&: mightbe-valid))
  ;;;;
  (viewpoint ?
    (exists (start-time ?segment ?)))
  ;;;;
  (not (applied ?segment sg-no-buzz-head-voiced-stop-right))
  (prop ?segment (phoneme-pos-in-word no-buzz-head))
  (not (end-time ?segment ?))
  =>
  (at ?v
    (assert (applied ?segment sg-no-buzz-head-voiced-stop-right))
    (conflicting-alternatives
      (hypothesize
        (assert (right-context ?segment vowel))
        (assert (CF (right-context ?segment vowel) =(CFunknown)))))))
```

```
-----
;;;
```

```

(defrule sg-no-buzz-head-voiced-stop-right-vowel-1
  "Find time the following burst starts."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (right-context ?segment vowel)
  ?x<-(CF (right-context ?segment vowel) ?CFright-vowel)
  (not (applied ?segment sg-no-buzz-head-voiced-stop-right-vowel-1))
  (prop ?segment (phoneme-pos-in-word no-buzz-head))
  (prop ?segment (candidate-loc ?from ?to))

;;; search segmenetation position of right vowel
  (voiced-stop-burst-exists ?voiced-stop-burst-start&:(>= ?voiced-stop-burst-start ?to)
    ?pos&~NONE
    ?voiced-stop-burst-freq
    after =(- ?to 10) 40
    ?*voiced-stop-burst-freq-min*
    ?*voiced-stop-burst-freq-max*
    ?CFvoiced-stop-burst&: mightbe-valid)

  (not (voiced-stop-burst-exists ?v-b-start ?v-b-end&~NONE ?v-b-freq
    after =(- ?to 10) 40
    ?*voiced-stop-burst-freq-min*
    ?*voiced-stop-burst-freq-max*
    ?another-burst&:(CFgreaterp ?another-burst ?CFvoiced-stop-burst)))

;;; check characteristics of right burst
  (power-strength ?pos =(+ ?pos 30) 0 1000 ?vpw-0-1000)
  (CF (vowel-by-voiced-stop 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
  (power-strength ?pos =(+ ?pos 30) 1000 6000 ?vpw-1000-6000)
  (CF (vowel-by-voiced-stop 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
  (power-strength ?pos =(+ ?pos 30) 5000 6000 ?vpw-5000-6000)
  (CF (voiced-stop-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
    ?CFvpr-0-1000-5000-6000&: mightbe-valid)
  =>
  (retract ?x)
  (assert (applied ?segment sg-no-buzz-head-voiced-stop-right-vowel-1))
  (assert (prop ?segment (following-vowel-start-time ?pos)))
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFright-vowel
      (CFweight ?*strong-evidence* ?CFvoiced-stop-burst)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
    ?CFvoiced-stop-burst))))

(defrule sg-no-buzz-head-voiced-stop-right-vowel-done-1
  ""
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (prop ?segment (phoneme-pos-in-word no-buzz-head))
  (CF (category ?segment voiced-stop) ?CF-category&: mightbe-valid)
  (right-context ?segment vowel)
  (CF (right-context ?segment vowel) ?CF-context&: mightbe-valid)
  (not (applied ?segment sg-no-buzz-head-voiced-stop-right-vowel-done-1))
  (prop ?segment (following-vowel-start-time ?pos))
  =>
  (assert (applied ?segment sg-no-buzz-head-voiced-stop-right-vowel-done-1))
  (hypothesize
    (assert (end-time ?segment ?pos))
    (assert (CF (end-time ?segment ?pos) =(CFvalue ?CF-context))))))

;;; -----

```

```

(defrule sg-voiced-stop-right-vowel-1
  "Find time the following burst starts."
  (declare (saliency ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (right-context ?segment vowel)
  ?x<-(CF (right-context ?segment vowel) ?CFright-vowel)
  (not (applied ?segment sg-voiced-stop-right-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (phoneme-pos-in-word ?head-or-mid&~no-buzz-head))

;;; search segmenetation position of right vowel
  (voiced-stop-burst-exists ?vpos
    ?voiced-stop-burst-end
    ?voiced-stop-burst-freq
    after =(- ?to 10) 100
    ?*voiced-stop-burst-freq-min*
    ?*voiced-stop-burst-freq-max*
    ?CFvoiced-stop-burst&: mightbe-valid)

  (not (voiced-stop-burst-exists ? ? ?
    after =(- ?to 10) 100
    ?*voiced-stop-burst-freq-min*
    ?*voiced-stop-burst-freq-max*
    ?another-burst&:(CFgreaterp ?another-burst ?CFvoiced-stop-burst)))

  (power-increase (?pos&~NONE&:(>= ?pos ?vpos) ?change-increase)
    after =(- ?vpos 10) 40 0 500 (11 15) 0.0 )

;;; check characteristics of voiced-stop
  (power-strength =(- ?vpos 30) ?vpos 0 500 ?vspw-0-500)
  (CF (voiced-stop-0-500-power ?vspw-0-500) ?CFvspw-0-500&: mightbe-valid)
  (power-strength =(- ?vpos 30) ?vpos 500 3500 ?vspw-500-3500)
  (CF (voiced-stop-500-3500-power ?vspw-500-3500) ?CFvspw-500-3500&: mightbe-valid)
  (CF (voiced-stop-power-ratio-0-500-500-3500 =(- ?vspw-0-500 ?vspw-500-3500))
    ?CFvspr-0-500-500-3500&: mightbe-valid)

;;; check characteristics of right burst
  (power-strength ?pos =(+ ?pos 30) 0 1000 ?vpw-0-1000)
  (CF (vowel-by-voiced-stop 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
  (power-strength ?pos =(+ ?pos 30) 1000 6000 ?vpw-1000-6000)
  (CF (vowel-by-voiced-stop 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
  (power-strength ?pos =(+ ?pos 30) 5000 6000 ?vpw-5000-6000)
  (CF (voiced-stop-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
    ?CFvpr-0-1000-5000-6000&: mightbe-valid)

=>
  (retract ?x)
  (assert (applied ?segment sg-voiced-stop-right-vowel-1))
  (assert (prop ?segment (following-vowel-start-time ?pos)))
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFright-vowel
      (CFweight ?*strong-evidence* ?CFvoiced-stop-burst)
      (CFweight ?*weak-evidence* ?CFvspw-0-500)
      (CFweight ?*weak-evidence* ?CFvspw-500-3500)
      (CFweight ?*evidence* ?CFvspr-0-500-500-3500)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
    ?CFvoiced-stop-burst))))))

(defrule sg-voiced-stop-right-vowel-2
  "Find time the following burst starts."
  (declare (saliency ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (right-context ?segment vowel)
  ?x<-(CF (right-context ?segment vowel) ?CFright-vowel)
  (not (applied ?segment sg-voiced-stop-right-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (phoneme-pos-in-word ?head-or-mid&~no-buzz-head))

;;; search segmenetation position of right vowel
  (voiced-stop-burst-exists ?pos
    ?voiced-stop-burst-end

```

```

?voiced-stop-burst-freq
after =(- ?to 10) 100
?*voiced-stop-burst-freq-min*
?*voiced-stop-burst-freq-max*
?CFvoiced-stop-burst:&: mightbe-valid)

(not (voiced-stop-burst-exists ? ? ?
      after =(- ?to 10) 100
      ?*voiced-stop-burst-freq-min*
      ?*voiced-stop-burst-freq-max*
      ?another-burst:&:(CFgreaterp ?another-burst ?CFvoiced-stop-burst)))

(power-increase (?pos-inc:&(or (eq ?pos-inc 'NONE) (< ?pos-inc ?pos)) ?change-increase)
  after =(- ?pos 10) 40 0 500 (11 15) 0.0)

;;; check characteristics of voiced-stop
(power-strength =(- ?pos 30) ?pos 0 500 ?vspw-0-500)
(CF (voiced-stop-0-500-power ?vspw-0-500) ?CFvspw-0-500&: mightbe-valid)
(power-strength =(- ?pos 30) ?pos 500 3500 ?vspw-500-3500)
(CF (voiced-stop-500-3500-power ?vspw-500-3500) ?CFvspw-500-3500&: mightbe-valid)
(CF (voiced-stop-power-ratio-0-500-500-3500 =(- ?vspw-0-500 ?vspw-500-3500))
  ?CFvspr-0-500-500-3500&: mightbe-valid)

;;; check characteristics of right burst
(power-strength ?pos =(+ ?pos 30) 0 1000 ?vpw-0-1000)
(CF (vowel-by-voiced-stop 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength ?pos =(+ ?pos 30) 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-voiced-stop 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
(power-strength ?pos =(+ ?pos 30) 5000 6000 ?vpw-5000-6000)
(CF (voiced-stop-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&: mightbe-valid)
=>
(retract ?x)
(assert (applied ?segment sg-voiced-stop-right-vowel-1))
(assert (prop ?segment (following-vowel-start-time ?pos)))
(assert (CF (right-context ?segment vowel)
  =(CFand (CFcomb ?CFright-vowel
    (CFweight ?*strong-evidence* ?CFvoiced-stop-burst)
    (CFweight ?*weak-evidence* ?CFvspw-0-500)
    (CFweight ?*weak-evidence* ?CFvspw-500-3500)
    (CFweight ?*evidence* ?CFvspr-0-500-500-3500)
    (CFweight ?*weak-evidence* ?CFvpw-0-1000)
    (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
    (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
  ?CFvoiced-stop-burst))))

(defcontradiction 1st-or-2nd-pos-voiced-stop-right ""
  (prop ?segment (following-vowel-start-time ?pos))
  (prop ?segment (following-vowel-start-time ~?pos)))

(defrule sg-voiced-stop-right-vowel-done-1
  ""
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment voiced-stop)
  (CF (category ?segment voiced-stop) ?CF-category:&: mightbe-valid)
  (right-context ?segment vowel)
  (CF (right-context ?segment vowel) ?CF-context:&: mightbe-valid)
  (not (applied ?segment sg-voiced-stop-right-vowel-done-1))
  (prop ?segment (following-vowel-start-time ?pos))
  (prop ?segment (phoneme-pos-in-word ?head-or-mid&~no-buzz-head))
  =>
  (assert (applied ?segment sg-voiced-stop-right-vowel-done-1))
  (hypothesize
    (assert (end-time ?segment ?pos))
    (assert (CF (end-time ?segment ?pos) =(CFvalue ?CF-context))))))

```

LM17: >hatazaki>sre>seg-voiced-stop>seg-voiced-stop-end. art. 21

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/25/89 20:49:20

Queued at: 3/28/89 11:19:17

```

;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
;;; check middle voiced-stop isnot R by duration time
(defrule sg-mid-voiced-stop-isnot-R-duration-time
  ""
  (declare (salience (+ ?*end-segmentation* 2)))
  (not (applied ?segment sg-mid-voiced-stop-isnot-R-duration-time))
  (prop ?segment (phoneme-pos-in-word middle))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))
  (join
    (category ?segment voiced-stop)
    ?x<-(CF (category ?segment ?voiced-stop) ?CFcategory&: mightbe-valid))
  (join
    (start-time ?segment ?start)
    (explicit (CF (start-time ?segment ?start) ?CFstart)))
  (join
    (end-time ?segment ?end)
    (explicit (CF (end-time ?segment ?end) ?CFend)))
  ;; not-R-duration-time
  ; (power-decrease (?vowel-end&~NONE ?change-decrease&:(?change-decrease >= 0.5))
  ;   after =(- ?start 10) =(- ?end (- ?start 10)) 1000 4000 (11 11)
  ;   (?*vowel-end-min-change-before-voiced-stop* -65))
  ; (power-increase (?vowel-start&~NONE ?change-increase&:(?change-increase >= 0.5))
  ;   before =(+ ?end 10) =(- (+ ?end 10) ?start) 1000 4000 (11 11)
  ;   (?*vowel-start-min-change-after-voiced-stop* -65))
  (CF (voiced-stop-not-R-duration-time =(- ?end ?start)) ?CFnot-R-duration)
  =>
  (retract ?x)
  (assert (applied ?segment sg-mid-voiced-stop-isnot-R-duration-time))
  (assert (CF (category ?segment voiced-stop)
    =(CFand (CFcomb ?CFcategory
      (CFweight ?*strong-evidence* ?CFnot-R-duration)
      ?CFnot-R-duration))))))

;;; check voiced-stop duration time
;(defrule sg-voiced-stop-duration-time
  ""
  ; (declare (salience (+ ?*end-segmentation* 1)))
  ; (not (applied ?segment sg-voiced-stop-duration-time))
  ; (join
  ;   (category ?segment voiced-stop)
  ;   ?x<-(CF (category ?segment ?voiced-stop) ?CFcategory &: mightbe-valid))
  ; (join
  ;   (start-time ?segment ?start)
  ;   (explicit (CF (start-time ?segment ?start) ?CFstart)))
  ; (join
  ;   (end-time ?segment ?end)
  ;   (explicit (CF (end-time ?segment ?end) ?CFend)))
  ; (CF (voiced-stop-duration-time =(- ?end ?start)) ?CFduration)
  ; =>
  ; (retract ?x)
  ; (assert (applied ?segment sg-voiced-stop-duration-time))
  ; (assert (CF (category ?segment voiced-stop)
  ;   =(CFand (CFcomb ?CFcategory
  ;     (CFweight ?*evidence* ?CFduration)
  ;     ?CFduration))))))

```



LM17: >hatazaki>SRE>SEG-NASAL>get-nasal-peak-candidates.lisp.12

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/31/89 10:35:06

Queued at: 3/28/89 11:07:57

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: ART-USER; Base: 10 -*-  
(defun get-nasal-peak-candidates ()  
  (let ((nasal-peak-candidates (caddr (USER::call-speechfunc  
                                       (list "POWER-RATIO-PEAKS"  
                                             0 2000  
                                             0 500  
                                             5000 6000  
                                             10 2.5  
                                             15 11))))))  
    ;  
    (loop for peak in nasal-peak-candidates  
          collect peak)))
```

LM17:>hatazaki>sr>seg-nasal>seg-nasal.art.30

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/19/89 20:30:43

Queued at: 3/28/89 11:04:08

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-nasal-0
  "Finds a candidate of nasal."
  (declare (salience ?*find-candidate*))
  (segmenting nasal)
  (nasal-peak-candidate ?peak-loc ?peak-value)
  (CF (nasal-peak-candidate ?peak-value) ?CFnasal-peak&:mightbe-valid)
; (nasal-region-candidate ?from ?to&:(?from <= ?peak-loc)&:(?peak-loc <= ?to))
  (BB ?BB)
  =>
  (bind ?segment (gensym "SEG-"))
  (at ?BB
    (assert (nasal-candidate ?peak-loc ?peak-loc))
    (assert (schema ?segment (instance-of phoneme-segment)))
    (hypothesize
      (assert (category ?segment nasal))
      (assert (prop ?segment (candidate-loc ?peak-loc ?peak-loc))
        (segment-status ?segment segmentation)
        (CF (category ?segment nasal)
          =(CFand (CFcomb (CFunknown)
            (CFweight ?*strong-evidence* ?CFnasal-peak)
            ?CFnasal-peak))))))

(defrule get-nasal-peak-candidates
  (declare (salience ?*top-down*))
  (goal (nasal-peak-candidate ?peak-loc?? ?peak-value??))
  (explicit (not (nasal-peak-candidate ?peak-loc ?peak-value)))
  =>
  (bind ?nasal-peak-candidates
    (get-nasal-peak-candidates))
  (for peak in ?nasal-peak-candidates
    do (assert =(seq$ (cons 'nasal-peak-candidate peak)))))

;(defrule get-nasal-region-candidates
;  (declare (salience ?*top-down*))
;  (goal (nasal-region-candidate ?from?? ?to??))
;  (explicit (not (nasal-region-candidate ?from ?to)))
;  =>
;  (bind ?nasal-region-candidates
;    (get-nasal-region-candidates))
;  (for region in ?nasal-region-candidates
;    do (assert =(seq$ (cons 'nasal-region-candidate region)))))
```

LM17: >hatazaki>sre>seg-nasal>seg-nasal-pos-in-word. art. 26

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/23/89 11:50:05

Queued at: 3/28/89 11:04:47

```
;;- Mode:ART; Package:art-user; Base:10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-nasal-pos-in-word
  "Hypothesizes position phoneme in word."
  (declare (saliency ?*start-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (not (applied ?segment sg-nasal-pos-in-word))
  =>
  (assert (applied ?segment sg-nasal-pos-in-word))
  (conflicting-alternatives
    (hypothesize
      (assert (prop ?segment (phoneme-pos-in-word head)))
      (assert (CF (prop ?segment (phoneme-pos-in-word head)) =(CFunknown))))
    (hypothesize
      (assert (prop ?segment (phoneme-pos-in-word middle)))
      (assert (CF (prop ?segment (phoneme-pos-in-word middle)) =(CFunknown))))))
```

```
;;-----
```

```
(defrule nasal-pos-head-in-word
  ""
  (declare (saliency ?*start-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (not (applied ?segment nasal-pos-head-in-word))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (phoneme-pos-in-word head))
  ?x <- (CF (prop ?segment (phoneme-pos-in-word head)) ?)
  (power-start ?power-start&~NONE& (>= 100 (- ?from ?power-start)
    after 0 2000 0 500 ?*silence-power-upper*))
  (power-strength =(- ?power-start 50) ?power-start 0 6000 ?pw-0-6000)
  (CF (nasal-power-is-silence ?pw-0-6000) ?CFpw-0-6000&: mightbe-valid)

  (power-increase (?vowel-start&~NONE ?change-increase)
    after =(- ?to 10) 200 5000 6000 (11 11)
    ?*vowel-start-min-change-after-nasal*)
  (power-decrease (?vowel-end&(eq ?vowel-end 'NONE) ?change-decrease)
    before ?vowel-start =(- ?vowel-start ?power-start)
    5000 6000 (11 11) (?*vowel-end-min-change-before-nasal* -60))
  =>
  (retract ?x)
  (assert (applied ?segment nasal-pos-head-in-word))
  (assert (CF (prop ?segment (phoneme-pos-in-word head)) =(CFcomb (CFunknown) 1.0))))
```

```
(defrule nasal-pos-middle-in-word
  ""
  (declare (saliency ?*start-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (not (applied ?segment nasal-pos-middle-in-word))
  (prop ?segment (candidate-loc ?from ?to))
  (prop ?segment (phoneme-pos-in-word middle))
  ?x <- (CF (prop ?segment (phoneme-pos-in-word middle)) ?)
  (power-increase (?vowel-start&~NONE ?change-increase)
    after ?from 200 5000 6000 (11 11)
    ?*vowel-start-min-change-after-nasal*)
  (power-decrease (?vowel-end&~NONE ?change-decrease)
    before ?to 200 5000 6000 (11 11)
    ?*vowel-end-min-change-before-nasal*)
  =>
  (retract ?x)
  (assert (applied ?segment nasal-pos-middle-in-word))
  (assert (CF (prop ?segment (phoneme-pos-in-word middle)) =(CFcomb (CFunknown) 1.0))))
```

LM17: >hatazaki>sre>seg-nasal>seg-nasal-left. art. 54

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/30/89 19:45:51

Queued at: 3/28/89 11:05:28

```
;;;-*- Mode: ART; Package: art-user; Base:10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-nasal-left
  ""
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (CF (category ?segment nasal) ?CF-category&:mightbe-valid)
  (not (applied ?segment sg-nasal-left))
  (not (start-time ?segment ?))
  =>
  (assert (applied ?segment sg-nasal-left))
  (conflicting-alternatives
   (hypothesize
    (assert (left-context ?segment silence))
    (assert (CF (left-context ?segment silence) =(CFunknown))))
   (hypothesize
    (assert (left-context ?segment vowel))
    (assert (CF (left-context ?segment vowel) =(CFunknown))))))
```

```
;;;-----
```



```

(defrule sg-head-nasal-left-silence-1
  "Find time of the silence-nasal transition."
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (prop ?segment (phoneme-pos-in-word head))
  (left-context ?segment silence)
  (not (applied ?segment sg-head-nasal-left-silence-1))
  (prop ?segment (candidate-loc ?from ?to))
  (CF (prop ?segment (phoneme-pos-in-word head)) ?CFhead&:(CFequal ?CFhead 1.0))

;;; search segmenetation position of left silence
  (power-start ?power-start&~NONE after 0 2000 0 500 ?*silence-power-upper*)
; (power-start ?power-start&~NONE before ?from 100 0 500 ?*silence-power-upper*)

;;; check characteristics of left silence
  (power-strength =(- ?power-start 50) ?power-start 0 6000 ?spw-0-6000)
  (CF (nasal-power-is-silence ?spw-0-6000) ?CFspw-0-6000&: mightbe-valid)

;;; check characteristics of nasal
  (power-strength ?power-start =(+ ?power-start 30) 0 500 ?npw-0-500)
  (CF (head-nasal-0-500-power ?npw-0-500) ?CFnpw-0-500&: mightbe-valid)
  (power-strength ?power-start =(+ ?power-start 30) 500 1000 ?npw-500-1000)
  (CF (head-nasal-500-1000-power ?npw-500-1000) ?CFnpw-500-1000&: mightbe-valid)
  (CF (nasal-power-ratio-0-500-500-1000 =(- ?npw-0-500 ?npw-500-1000))
    ?CFnpr-0-500-500-1000&: mightbe-valid)
  ?x<-(CF (left-context ?segment silence) ?CFleft-silence)
  =>
  (retract ?x)
  (assert (applied ?segment sg-head-nasal-left-silence-1))
  (assert (prop ?segment (proceeding-silence-end-time ?power-start)))
  (assert (CF (left-context ?segment silence)
    =(CFand (CFcomb ?CFleft-silence
      (CFweight ?*weak-evidence* ?CFspw-0-6000)
      (CFweight ?*weak-evidence* ?CFnpw-0-500)
      (CFweight ?*weak-evidence* ?CFnpw-500-1000)
      (CFweight ?*weak-evidence* ?CFnpr-0-500-500-1000))
    ?CFspw-0-6000))))

(defrule sg-head-nasal-left-silence-done-1
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (CF (category ?segment nasal) ?CF-category&: mightbe-valid)
  (prop ?segment (phoneme-pos-in-word head))
  (left-context ?segment silence)
  (CF (left-context ?segment silence) ?CF-context&: mightbe-valid)
  (not (applied ?segment sg-head-nasal-left-silence-done-1))
  (CF (prop ?segment (phoneme-pos-in-word head)) ?CFhead&:(CFequal ?CFhead 1.0))
  (prop ?segment (proceeding-silence-end-time ?pos))
  =>
  (assert (applied ?segment sg-head-nasal-left-silence-done-1))
  (hypothesize
    (assert (start-time ?segment ?pos))
    (assert (CF (start-time ?segment ?pos) =(CFvalue ?CF-context))))))

;;;-----

```

```

(defrule sg-mid-nasal-left-vowel-1
  "Find time the preceding vowel starts."
  (declare (saliency (+ ?*left-segmentation* 1)))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (prop ?segment (phoneme-pos-in-word middle))
  (left-context ?segment vowel)
  ?x<-(CF (left-context ?segment vowel) ?CFleft-vowel)
  (not (applied ?segment sg-mid-nasal-left-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))

;;; search segmenetation position of left vowel
(power-decrease (?vowel-end&~NONE ?change-decrease)
  before ?from 100
  5000 6000 (11 11) ?*vowel-end-min-change-before-nasal*)
(spectral-change (?pos&~NONE ?value)
  before ?from 100
  0 6000 ( 9 5) 7.5 0.0 0.1 2.8)
(power-increase (?pw-inc-pos ?change-increase)
  after =(- ?pos 20) 30 1000 4000 (11 11) 0.5)
(CF (mid-nasal-increase-near-seg-pos ?change-increase) ?CFchange-increase&: mightbe-valid)
(CF (mid-nasal-spectral-change-left ?value) ?CFsp-change&: mightbe-valid)

;;; check characteristics of nasal
(power-strength ?pos =(+ ?pos 30) 0 500 ?npw-0-500)
(CF (mid-nasal-0-500-power ?npw-0-500) ?CFnpw-0-500&: mightbe-valid)
(power-strength ?pos =(+ ?pos 30) 500 1000 ?npw-500-1000)
(CF (mid-nasal-500-1000-power ?npw-500-1000) ?CFnpw-500-1000&: mightbe-valid)
(CF (nasal-power-ratio-0-500-500-1000 =(- ?npw-0-500 ?npw-500-1000))
  ?CFnpr-0-500-500-1000&: mightbe-valid)

;;; check characteristics of left vowel
(power-strength =(- ?pos 20) ?pos 0 1000 ?vpw-0-1000)
(CF (vowel-by-nasal 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength =(- ?pos 20) ?pos 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-nasal 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
(power-strength =(- ?pos 20) ?pos 5000 6000 ?vpw-5000-6000)
(CF (nasal-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&: mightbe-valid)
=>
(assert (applied ?segment sg-mid-nasal-left-vowel-1))
(hypothesize
  (assert (prop ?segment (preceding-vowel-end-time ?pos)))
  (retract ?x)
  (assert (CF (left-context ?segment vowel)
    =(CFand (CFcomb ?CFleft-vowel
      (CFweight ?*strong-evidence* ?CFsp-change)
      (CFweight ?*evidence* ?CFchange-increase)
      (CFweight ?*weak-evidence* ?CFnpw-0-500)
      (CFweight ?*weak-evidence* ?CFnpw-500-1000)
      (CFweight ?*evidence* ?CFnpr-0-500-500-1000)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
      ?CFsp-change
      ?CFchange-increase
    )))))

(defrule sg-mid-nasal-left-vowel-2
  "Find time the preceding vowel starts."
  (declare (saliency (+ ?*left-segmentation* 1)))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (prop ?segment (phoneme-pos-in-word middle))
  (left-context ?segment vowel)
  ?x<-(CF (left-context ?segment vowel) ?CFleft-vowel)
  (not (applied ?segment sg-mid-nasal-left-vowel-2))
  (prop ?segment (candidate-loc ?from ?to))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))

;;; search segmenetation position of left vowel
(power-decrease (?vowel-end&~NONE ?change-decrease)
  before ?from 100

```

```

                    5000 6000 (11 11) ?*vowel-end-min-change-before-nasal*)
(spectral-change (?pos0&~NONE ?value0)   ;;; first spectral-peak backward
before ?from 100
0 6000 ( 9 5) 7.5 0.0 0.1 2.8)
(spectral-change (?pos1&~NONE ?value1)   ;;; second spectral-peak backward
before ?pos0 40
0 6000 ( 9 3) 7.5 0.0 0.1 2.8)
(power-increase (?pw-inc-pos ?change-increase)
after =(- ?pos1 20) 30 1000 4000 (11 11) 0.5)
(CF (mid-nasal-increase-near-seg-pos ?change-increase) ?CFchange-increase&: mightbe-valid)
(CF (mid-nasal-spectral-change-left ?value1) ?CFsp-change&: mightbe-valid)

;;; check characteristics of nasal
(power-strength ?pos1 =(+ ?pos1 30) 0 500 ?npw-0-500)
(CF (mid-nasal-0-500-power ?npw-0-500) ?CFnpw-0-500&: mightbe-valid)
(power-strength ?pos1 =(+ ?pos1 30) 500 1000 ?npw-500-1000)
(CF (mid-nasal-500-1000-power ?npw-500-1000) ?CFnpw-500-1000&: mightbe-valid)
(CF (nasal-power-ratio-0-500-500-1000 =(- ?npw-0-500 ?npw-500-1000))
?CFnpr-0-500-500-1000&: mightbe-valid)

;;; check characteristics of left vowel
(power-strength =(- ?pos1 20) ?pos1 0 1000 ?vpw-0-1000)
(CF (vowel-by-nasal 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength =(- ?pos1 20) ?pos1 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-nasal 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
(power-strength =(- ?pos1 20) ?pos1 5000 6000 ?vpw-5000-6000)
(CF (nasal-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
?CFvpr-0-1000-5000-6000&: mightbe-valid)
=>
(assert (applied ?segment sg-mid-nasal-left-vowel-2))
(hypothesize
(assert (prop ?segment (preceding-vowel-end-time ?pos1)))
(retract ?x)
(assert (CF (left-context ?segment vowel)
=(CFand (CFcomb ?CFleft-vowel
(CFweight ?*strong-evidence* ?CFsp-change)
(CFweight ?*evidence* ?CFchange-increase)
(CFweight ?*weak-evidence* ?CFnpw-0-500)
(CFweight ?*weak-evidence* ?CFnpw-500-1000)
(CFweight ?*evidence* ?CFnpr-0-500-500-1000)
(CFweight ?*weak-evidence* ?CFvpw-0-1000)
(CFweight ?*weak-evidence* ?CFvpw-1000-6000)
(CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
?CFsp-change
?CFchange-increase
))))))

(defcontradiction 1st-or-2nd-peaks-mid-nasal-left ""
(prop ?segment (preceding-vowel-end-time ?pos))
(prop ?segment (preceding-vowel-end-time ~?pos)))

(defrule sg-mid-nasal-left-vowel-done-1
""
(declare (salience ?*left-segmentation-end*))
(segment-status ?segment segmentation)
(category ?segment nasal)
(CF (category ?segment nasal) ?CF-category&: mightbe-valid)
(prop ?segment (phoneme-pos-in-word middle))
(CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&: (CFequal ?CFmiddle 1.0))
(left-context ?segment vowel)
(CF (left-context ?segment vowel) ?CF-context&: mightbe-valid)
(not (applied ?segment sg-mid-nasal-left-vowel-done-1))
(prop ?segment (preceding-vowel-end-time ?pos))
=>
(assert (applied ?segment sg-mid-nasal-left-vowel-done-1))
(hypothesize
(assert (start-time ?segment ?pos))
(assert (CF (start-time ?segment ?pos) =(CFvalue ?CF-context))))))

```

LM17:>hatazaki>sre>seg-nasal>seg-nasal-right.art.54

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/02/89 15:52:38

Queued at: 3/28/89 11:06:16

;;;-\*- Mode:ART; Package:art-user; Base:10; Default-character-style: (:FIX :ROMAN :NORMAL) -\*-

```

(defrule sg-nasal-right
  "Hypothesizes right phoneme."
  (declare (salience ?*right-segmentation*))
  (viewpoint ?v
    (segment-status ?segment segmentation)
    (category ?segment nasal)
    (CF (category ?segment nasal) ?CF-category&: mightbe-valid))
  ;;;;
  (viewpoint ?
    (exists (start-time ?segment ?)))
  ;;;;
  (not (applied ?segment sg-nasal-right))
  (not (end-time ?segment ?))
  =>
  (at ?v
    (assert (applied ?segment sg-nasal-right))
    (conflicting-alternatives
      (hypothesize
        (assert (right-context ?segment vowel))
        (assert (CF (right-context ?segment vowel) =(CFunknown)))))))
  ;;;-----

```

```

(defrule sg-mid-nasal-right-vowel-1
  "Find time the following vowel starts."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (prop ?segment (phoneme-pos-in-word middle))
  (right-context ?segment vowel)
  ?x<-(CF (right-context ?segment vowel) ?CFright-vowel)
  (not (applied ?segment sg-mid-nasal-right-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))

;;; search segmenetation position of right vowel
(power-increase (?vowel-start&~NONE ?change-increase)
  after =(- ?to 10) 100
  5000 6000 (11 11) ?*vowel-start-min-change-after-nasal*)
(spectral-change (?pos&~NONE ?value)
  after =(- ?to 10) 100
  0 6000 ( 9 5) 7.5 0.0 0.1 2.4)
(power-decrease (?pw-dec-pos&(eq ?pw-dec-pos 'NONE) ?change-decrease)
  after =(- ?pos 15) 30 1000 4000 (11 11) 0.5)
(CF (mid-nasal-spectral-change-right ?value) ?CFsp-change&: mightbe-valid)

;;; check charasteristics of nasal
(power-strength =(- ?pos 30) ?pos 0 500 ?npw-0-500)
(CF (mid-nasal-0-500-power ?npw-0-500) ?CFnpw-0-500&: mightbe-valid)
(power-strength =(- ?pos 30) ?pos 500 1000 ?npw-500-1000)
(CF (mid-nasal-500-1000-power ?npw-500-1000) ?CFnpw-500-1000&: mightbe-valid)
(CF (nasal-power-ratio-0-500-500-1000 =(- ?npw-0-500 ?npw-500-1000))
  ?CFnpr-0-500-500-1000&: mightbe-valid)

;;; check charasterictics of right vowel
(power-strength ?pos =(+ ?pos 20) 0 1000 ?vpw-0-1000)
(CF (vowel-by-nasal 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength ?pos =(+ ?pos 20) 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-nasal 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
(power-strength ?pos =(+ ?pos 20) 5000 6000 ?vpw-5000-6000)
(CF (nasal-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&: mightbe-valid)
=>
(assert (applied ?segment sg-mid-nasal-right-vowel-1))
(hypothesize
  (assert (prop ?segment (following-vowel-start-time ?pos)))
  (retract ?x)
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFright-vowel
      (CFweight ?*strong-evidence* ?CFsp-change)
      (CFweight ?*weak-evidence* ?CFnpw-0-500)
      (CFweight ?*weak-evidence* ?CFnpw-500-1000)
      (CFweight ?*evidence* ?CFnpr-0-500-500-1000)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
    ?CFsp-change
  ))))

(defrule sg-mid-nasal-right-vowel-2
  "Find time the following vowel starts."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (prop ?segment (phoneme-pos-in-word middle))
  (right-context ?segment vowel)
  ?x<-(CF (right-context ?segment vowel) ?CFright-vowel)
  (not (applied ?segment sg-mid-nasal-right-vowel-2))
  (prop ?segment (candidate-loc ?from ?to))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))
;;; search segmenetation position of right vowel
(power-increase (?vowel-start&~NONE ?change-increase)
  after =(- ?to 10) 100
  5000 6000 (11 11) ?*vowel-start-min-change-after-nasal*)
(spectral-change (?pos0&~NONE ?value0)
  after =(- ?to 10) 100
  0 6000 ( 9 5) 7.5 0.0 0.1 2.4)

```

```

(spectral-change (?pos1&~NONE ?value1) ;;; second spectral-peak forward
  after ?pos0 70
  0      6000 ( 9 3) 7.5 0.0 0.1 3.0)
(power-decrease (?pw-dec-pos&:(eq ?pw-dec-pos 'NONE) ?change-decrease)
  after =(- ?pos1 15) 30 1000 4000 (11 11) 0.5)

(CF (mid-nasal-spectral-change-right ?value1) ?CFsp-change&:mightbe-valid)

;;; check characteristics of nasal
(power-strength =(- ?pos1 30) ?pos1 0 500 ?npw-0-500)
(CF (mid-nasal-0-500-power ?npw-0-500) ?CFnpw-0-500&:mightbe-valid)
(power-strength =(- ?pos1 30) ?pos1 500 1000 ?npw-500-1000)
(CF (mid-nasal-500-1000-power ?npw-500-1000) ?CFnpw-500-1000&:mightbe-valid)
(CF (nasal-power-ratio-0-500-500-1000 =(- ?npw-0-500 ?npw-500-1000))
  ?CFnpr-0-500-500-1000&:mightbe-valid)

;;; check characteristics of right vowel
(power-strength ?pos1 =(+ ?pos1 20) 0 1000 ?vpw-0-1000)
(CF (vowel-by-nasal 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&:mightbe-valid)
(power-strength ?pos1 =(+ ?pos1 20) 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-nasal 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&:mightbe-valid)
(power-strength ?pos1 =(+ ?pos1 20) 5000 6000 ?vpw-5000-6000)
(CF (nasal-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&:mightbe-valid)

=>
(assert (applied ?segment sg-mid-nasal-right-vowel-2))
(hypothesize
  (assert (prop ?segment (following-vowel-start-time ?pos1)))
  (retract ?x)
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFright-vowel
      (CFweight ?*strong-evidence* ?CFsp-change)
      (CFweight ?*weak-evidence* ?CFnpw-0-500)
      (CFweight ?*weak-evidence* ?CFnpw-500-1000)
      (CFweight ?*evidence* ?CFnpr-0-500-500-1000)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
    ?CFsp-change
    )))))

(defcontradiction 1st-or-2nd-peaks-mid-nasal-right ""
  (prop ?segment (following-vowel-start-time ?pos))
  (prop ?segment (following-vowel-start-time ~?pos)))

(defrule sg-mid-nasal-right-vowel-done-1
  ""
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (CF (category ?segment nasal) ?CF-category&:mightbe-valid)
  (prop ?segment (phoneme-pos-in-word middle))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))
  (right-context ?segment vowel)
  (CF (right-context ?segment vowel) ?CF-context&:mightbe-valid)
  (not (applied ?segment sg-mid-nasal-right-vowel-done-1))
  (prop ?segment (following-vowel-start-time ?pos))
  =>
  (assert (applied ?segment sg-mid-nasal-right-vowel-done-1))
  (hypothesize
    (assert (end-time ?segment ?pos))
    (assert (CF (end-time ?segment ?pos) =(CFvalue ?CF-context))))))

```

```

(defrule sg-head-nasal-right-vowel-1
  "Find time the following vowel starts."
  (declare (saliency ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (prop ?segment (phoneme-pos-in-word head))
  (right-context ?segment vowel)
  ?x<-(CF (right-context ?segment vowel) ?CFright-vowel)
  (not (applied ?segment sg-head-nasal-right-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))
  (CF (prop ?segment (phoneme-pos-in-word head)) ?CFhead&:(CFequal ?CFhead 1.0))

;;; search segmenetation position of right vowel
(power-increase (?vowel-start&~NONE ?change-increase)
  after =(- ?to 10) 100
  5000 6000 (11 11) ?*vowel-start-min-change-after-nasal*)
(spectral-change (?pos&~NONE ?value)
  after =(- ?to 10) 100
  0 6000 ( 9 5) 7.5 0.0 0.1 3.2)
(power-decrease (?pw-dec-pos&:(eq ?pw-dec-pos 'NONE) ?change-decrease)
  after =(- ?pos 15) 30 1000 4000 (11 11) 0.5)

(CF (head-nasal-spectral-change-right ?value) ?CFsp-change&:mightbe-valid)

;;; check characteristics of nasal
(power-strength =(- ?pos 30) ?pos 0 500 ?npw-0-500)
(CF (head-nasal-0-500-power ?npw-0-500) ?CFnpw-0-500&:mightbe-valid)
(power-strength =(- ?pos 30) ?pos 500 1000 ?npw-500-1000)
(CF (head-nasal-500-1000-power ?npw-500-1000) ?CFnpw-500-1000&:mightbe-valid)
(CF (nasal-power-ratio-0-500-500-1000 =(- ?npw-0-500 ?npw-500-1000))
  ?CFnpr-0-500-500-1000&:mightbe-valid)

;;; check characteristics of right vowel
(power-strength ?pos =(+ ?pos 20) 0 1000 ?vpw-0-1000)
(CF (vowel-by-nasal 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&:mightbe-valid)
(power-strength ?pos =(+ ?pos 20) 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-nasal 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&:mightbe-valid)
(power-strength ?pos =(+ ?pos 20) 5000 6000 ?vpw-5000-6000)
(CF (nasal-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&:mightbe-valid)
=>
(assert (applied ?segment sg-head-nasal-right-vowel-1))
(hypothesize
  (assert (prop ?segment (following-vowel-start-time ?pos)))
  (retract ?x)
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFright-vowel
      (CFweight ?*strong-evidence* ?CFsp-change)
      (CFweight ?*weak-evidence* ?CFnpw-0-500)
      (CFweight ?*weak-evidence* ?CFnpw-500-1000)
      (CFweight ?*evidence* ?CFnpr-0-500-500-1000)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
    ?CFsp-change
  )))))

(defrule sg-head-nasal-right-vowel-2
  "Find time the following vowel starts."
  (declare (saliency ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (prop ?segment (phoneme-pos-in-word head))
  (right-context ?segment vowel)
  ?x<-(CF (right-context ?segment vowel) ?CFright-vowel)
  (not (applied ?segment sg-head-nasal-right-vowel-2))
  (prop ?segment (candidate-loc ?from ?to))
  (CF (prop ?segment (phoneme-pos-in-word head)) ?CFhead&:(CFequal ?CFhead 1.0))

;;; search segmenetation position of right vowel
(power-increase (?vowel-start&~NONE ?change-increase)
  after =(- ?to 10) 100
  5000 6000 (11 11) ?*vowel-start-min-change-after-nasal*)
(spectral-change (?pos0&~NONE ?value0)

```



```

        after =(- ?to 10) 100
        0      6000 ( 9 5) 7.5 0.0 0.1 3.2)
(spectral-change (?pos1&~NONE ?value1) ;;; second spectral-peak forward
  after ?pos0 70
  0      6000 ( 9 3) 7.5 0.0 0.1 3.2)
(power-decrease (?pw-dec-pos&:(eq ?pw-dec-pos 'NONE) ?change-decrease)
  after =(- ?pos1 15) 30 1000 4000 (11 11) 0.5)

(CF (head-nasal-spectral-change-right ?value1) ?CFsp-change&:mightbe-valid)

;;; check characteristics of nasal
(power-strength =(- ?pos1 30) ?pos1 0 500 ?npw-0-500)
(CF (head-nasal-0-500-power ?npw-0-500) ?CFnpw-0-500&:mightbe-valid)
(power-strength =(- ?pos1 30) ?pos1 500 1000 ?npw-500-1000)
(CF (head-nasal-500-1000-power ?npw-500-1000) ?CFnpw-500-1000&:mightbe-valid)
(CF (nasal-power-ratio-0-500-500-1000 =(- ?npw-0-500 ?npw-500-1000))
  ?CFnpr-0-500-500-1000&:mightbe-valid)

;;; check characteristics of right vowel
(power-strength ?pos1 =(+ ?pos1 20) 0 1000 ?vpw-0-1000)
(CF (vowel-by-nasal 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&:mightbe-valid)
(power-strength ?pos1 =(+ ?pos1 20) 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-nasal 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&:mightbe-valid)
(power-strength ?pos1 =(+ ?pos1 20) 5000 6000 ?vpw-5000-6000)
(CF (nasal-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&:mightbe-valid)
=>
(assert (applied ?segment sg-head-nasal-right-vowel-2))
(hypothesize
  (assert (prop ?segment (following-vowel-start-time ?pos1)))
  (retract ?x)
  (assert (CF (right-context ?segment vowel)
    =(CFand (CFcomb ?CFright-vowel
      (CFweight ?*strong-evidence* ?CFsp-change)
      (CFweight ?*weak-evidence* ?CFnpw-0-500)
      (CFweight ?*weak-evidence* ?CFnpw-500-1000)
      (CFweight ?*evidence* ?CFnpr-0-500-500-1000)
      (CFweight ?*weak-evidence* ?CFvpw-0-1000)
      (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
      (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
      ?CFsp-change
    )))))
)

(defrule sg-head-nasal-right-vowel-done-1
  ""
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment nasal)
  (CF (category ?segment nasal) ?CF-category&:mightbe-valid)
  (prop ?segment (phoneme-pos-in-word head))
  (CF (prop ?segment (phoneme-pos-in-word head)) ?CFhead&:(CFequal ?CFhead 1.0))
  (right-context ?segment vowel)
  (CF (right-context ?segment vowel) ?CF-context&:mightbe-valid)
  (not (applied ?segment sg-head-nasal-right-vowel-done-1))
  (prop ?segment (following-vowel-start-time ?pos))
  =>
  (assert (applied ?segment sg-head-nasal-right-vowel-done-1))
  (hypothesize
    (assert (end-time ?segment ?pos))
    (assert (CF (end-time ?segment ?pos) =(CFvalue ?CF-context))))))
)

;;;-----

```

LM17:>hatazaki>sre>seg-nasal>seg-nasal-end.art.21

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/25/89 20:49:14

Queued at: 3/28/89 11:07:05

```

;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
;;; check middle nasal isnot R by duration time
(defrule sg-mid-nasal-isnot-R-duration-time
  ""
  (declare (salience (+ ?*end-segmentation* 2)))
  (not (applied ?segment sg-mid-nasal-isnot-R-duration-time))
  (prop ?segment (phoneme-pos-in-word middle))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))
  (join
    (category ?segment nasal)
    ?x<-(CF (category ?segment ?nasal) ?CFcategory&:mightbe-valid))
  (join
    (start-time ?segment ?start)
    (explicit (CF (start-time ?segment ?start) ?CFstart)))
  (join
    (end-time ?segment ?end)
    (explicit (CF (end-time ?segment ?end) ?CFend)))
  ;; not-R-duration-time
  (power-decrease (?vowel-end&~NONE ?change-decrease&:(?change-decrease >= 0.5))
    after =(- ?start 10) =(- ?end (- ?start 10)) 1000 4000 (11 11)
    (?*vowel-end-min-change-before-nasal* -65))
  (power-increase (?vowel-start&~NONE&:(>= (- ?start ?vowel-end) 40)
    ?change-increase&:(?change-increase >= 0.5))
    before =(+ ?end 10) =(- (+ ?end 10) ?start) 1000 4000 (11 11)
    (?*vowel-start-min-change-after-nasal* -65))
  (CF (not-R-duration-time =(- ?vowel-start ?vowel-end)) ?CFnot-R-duration)
  =>
  (retract ?x)
  (assert (applied ?segment sg-mid-nasal-isnot-R-duration-time))
  (assert (CF (category ?segment nasal)
    =(CFand (CFcomb ?CFcategory
      (CFweight ?*strong-evidence* ?CFnot-R-duration))
    ?CFnot-R-duration))))

(defrule sg-mid-nasal-duration-time
  ""
  (declare (salience (+ ?*end-segmentation* 2)))
  (not (applied ?segment sg-mid-nasal-isnot-R-duration-time))
  (prop ?segment (phoneme-pos-in-word middle))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))
  (join
    (category ?segment nasal)
    ?x<-(CF (category ?segment ?nasal) ?CFcategory&:mightbe-valid))
  (join
    (start-time ?segment ?start)
    (explicit (CF (start-time ?segment ?start) ?CFstart)))
  (join
    (end-time ?segment ?end)
    (explicit (CF (end-time ?segment ?end) ?CFend)))
  (CF (mid-nasal-duration-time =(- ?end ?start)) ?CFnasal-duration)
  =>
  (retract ?x)
  (assert (applied ?segment sg-mid-nasal-isnot-R-duration-time))
  (assert (CF (category ?segment nasal)
    =(CFand (CFcomb ?CFcategory
      (CFweight ?*strong-evidence* ?CFnasal-duration))
    ?CFnasal-duration))))

(defrule sg-mid-nasal-isnot-nasal-segment
  ""
  (declare (salience (+ ?*end-segmentation* 2)))
  (not (applied ?segment sg-mid-nasal-isnot-nasal-segment))
  (prop ?segment (phoneme-pos-in-word middle))
  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))
  (join
    (category ?segment nasal)
    ?x<-(CF (category ?segment ?nasal) ?CFcategory&:mightbe-valid))
  (join
    (start-time ?segment ?start)
    (explicit (CF (start-time ?segment ?start) ?CFstart)))
  (join
    (end-time ?segment ?end)
    (explicit (CF (end-time ?segment ?end) ?CFend)))
  (power-decrease (?vowel-end&~NONE ?change-decrease&:(?change-decrease >= 0.5))
    after =(+ ?start 10) =(- ?end (+ ?start 10)) 1000 4000 (11 11)

```

```

                (?*vowel-end-min-change-before-nasal* -65))
=>
(bind ?CFnew-category 0.0)
(retract ?x)
(assert (applied ?segment sg-mid-nasal-isnot-nasal-segment))
(assert (CF (category ?segment nasal)
            =(CFand (CFcomb ?CFnew-category))))

;;; check nasal duration time
;(defrule sg-nasal-duration-time
;  ""
;  (declare (salience (+ ?*end-segmentation* 1)))
;  (not (applied ?segment sg-nasal-duration-time))
;  (join
;    (category ?segment nasal)
;    ?x<-(CF (category ?segment ?nasal) ?CFcategory&: mightbe-valid))
;  (join
;    (start-time ?segment ?start)
;    (explicit (CF (start-time ?segment ?start) ?CFstart)))
;  (join
;    (end-time ?segment ?end)
;    (explicit (CF (end-time ?segment ?end) ?CFend)))
;  (CF (nasal-duration-time =(- ?end ?start)) ?CFduration)
;  =>
;  (retract ?x)
;  (assert (applied ?segment sg-nasal-duration-time))
;  (assert (CF (category ?segment nasal)
              =(CFand (CFcomb ?CFcategory
                          (CFweight ?*strong-evidence* ?CFduration)
                          ?CFduration))))

;(defrule sg-head-nasal-end-1
;  ""
;  ;;
;  ;; komori 1988.10.28 (FRI) insert CF into segmentation
;  ;;
;  (declare (salience (+ ?*end-segmentation* 1)))
;  (applied ?segment sg-nasal-duration-time)
;  (not (applied ?segment sg-head-nasal-end-1))
;  (prop ?segment (phoneme-pos-in-word head))
;  (CF (prop ?segment (phoneme-pos-in-word head)) ?CFhead&:(CFequal ?CFhead 1.0))
;  (join
;    (category ?segment nasal)
;    ?x<-(CF (category ?segment ?nasal) ?CFcategory&: mightbe-valid))
;  (join
;    (start-time ?segment ?start)
;    (explicit (CF (start-time ?segment ?start) ?CFstart)))
;  (join
;    (end-time ?segment ?end)
;    (explicit (CF (end-time ?segment ?end) ?CFend)))
;  ;;; low frequency characteristics of nasal
;  (power-strength ?start ?end 0 500 ?npw-0-500)
;  (CF (head-nasal-0-500-power ?npw-0-500) ?CFnpw-0-500)
;  (power-strength ?start ?end 500 1000 ?npw-500-1000)
;  (CF (head-nasal-500-1000-power ?npw-500-1000) ?CFnpw-500-1000)
;  (CF (nasal-power-ratio-0-500-500-1000 =(- ?npw-0-500 ?npw-500-1000)) ?CFnpr-0-500-500-1000)
;  ;;; high frequency characteristics of nasal
;  (power-strength ?start ?end 4000 6000 ?npw-4000-6000)
;  (CF (nasal-4000-6000-power ?npw-4000-6000) ?CFnpw-4000-6000)
;  ;;; characteristics of nasal
;  (power-strength ?start ?end 0 6000 ?npw-0-6000)
;  (CF (nasal-0-6000-power ?npw-0-6000) ?CFnpw-0-6000)
;  =>
;  (retract ?x)
;  (assert (applied ?segment sg-head-nasal-end-1))
;  (assert (CF (category ?segment nasal)
              =(CFand
                (CFcomb ?CFcategory
                  (CFweight ?*evidence* ?CFnpr-0-500-500-1000)
                  (CFweight ?*weak-evidence* ?CFnpw-0-500)
                  (CFweight ?*weak-evidence* ?CFnpw-500-1000)
                  (CFweight ?*weak-evidence* ?CFnpw-4000-6000)
                  (CFweight ?*weak-evidence* ?CFnpw-0-6000))
                ?CFnpr-0-500-500-1000
                ?CFnpw-0-500
                ?CFnpw-500-1000

```

```
;          ?CFnpw-4000-6000
;          ?CFnpw-0-6000)))
;
;(defrule sg-mid-nasal-end-1
;  ""
;  ;;
;  ;; komori 1988.10.28 (FRI) insert CF into segmentation
;  ;;
;  (declare (salience (+ ?*end-segmentation* 1)))
;  (applied ?segment sg-nasal-duration-time )
;  (not (applied ?segment sg-mid-nasal-end-1))
;  (prop ?segment (phoneme-pos-in-word middle))
;  (CF (prop ?segment (phoneme-pos-in-word middle)) ?CFmiddle&:(CFequal ?CFmiddle 1.0))
;  (join
;    (category ?segment nasal)
;    ?x<-(CF (category ?segment ?nasal) ?CFcategory&:mightbe-valid))
;  (join
;    (start-time ?segment ?start)
;    (explicit (CF (start-time ?segment ?start) ?CFstart)))
;  (join
;    (end-time ?segment ?end)
;    (explicit (CF (end-time ?segment ?end) ?CFend)))
;  ;; low frequency characteristics of nasal
;  (power-strength ?start ?end 0 500 ?npw-0-500)
;  (CF (mid-nasal-0-500-power ?npw-0-500) ?CFnpw-0-500)
;  (power-strength ?start ?end 500 1000 ?npw-500-1000)
;  (CF (mid-nasal-500-1000-power ?npw-500-1000) ?CFnpw-500-1000)
;  (CF (nasal-power-ratio-0-500-500-1000 =(- ?npw-0-500 ?npw-500-1000)) ?CFnpr-0-500-500-1000)
;  ;; high frequency characteristics of nasal
;  (power-strength ?start ?end 4000 6000 ?npw-4000-6000)
;  (CF (nasal-4000-6000-power ?npw-4000-6000) ?CFnpw-4000-6000)
;  ;; characteristics of nasal
;  (power-strength ?start ?end 0 6000 ?npw-0-6000)
;  (CF (nasal-0-6000-power ?npw-0-6000) ?CFnpw-0-6000)
;  =>
;  (retract ?x)
;  (assert (applied ?segment sg-mid-nasal-end-1))
;  (assert (CF (category ?segment nasal)
;    =(CFand (CFcomb ?CFcategory
;      (CFweight ?*evidence* ?CFnpr-0-500-500-1000)
;      (CFweight ?*weak-evidence* ?CFnpw-0-500)
;      (CFweight ?*weak-evidence* ?CFnpw-500-1000)
;      (CFweight ?*weak-evidence* ?CFnpw-4000-6000)
;      (CFweight ?*weak-evidence* ?CFnpw-0-6000))
;      ?CFnpr-0-500-500-1000
;      ?CFnpw-0-500
;      ?CFnpw-500-1000
;      ?CFnpw-4000-6000
;      ?CFnpw-0-6000))))))
;
;
```

LM17: >hatazaki>SRE>SEG-LIQUID>get-mid-liquid-region-candidates.11sp.13

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/06/89 10:22:07

Queued at: 3/28/89 11:13:40

```
;;; -*- Mode: LISP; Syntax: Common-lisp; Package: ART-USER; Base: 10 -*-
```

```
(defun get-mid-liquid-region-candidates()
  (let* ((start-time 1)
         (end-time 2000)
         section1
         section2
         section3)

    (setq section1 (loop for i from start-time to end-time by 2.5
                        with liquid-start-time and liquid-end-time
                        while (and (/= 0 (setq i (find-liquid-start-candidate i end-time)))
                                   (/= 0 (setq liquid-end-time (find-liquid-end-candidate i end-time)))
                                   (setq liquid-start-time i))
                        when (< (- liquid-end-time liquid-start-time) 40)
                        collect (list liquid-start-time liquid-end-time)))

    (setq section2
          (caddr (USER::call-speechfunc (list "POWER-RATIO-ISLAND" 0 2000
                                             0 500
                                             1000 6000
                                             2 999
                                             11 9))))

    (setq section3
          (caddr (USER::call-speechfunc (list "POWER-RATIO-ISLAND" 0 2000
                                             0 500
                                             4000 6000
                                             5 999
                                             11 9))))

    (loop for region in (intersected-regions (list section1 section2 section3))
          collect region)
  ))

(defun find-liquid-start-candidate (start-time end-time)
  (caar (caddr (USER::call-speechfunc
                (list "POWER-CHANGE"
                     start-time end-time
                     1000 4000
                     2 21
                     5 5 ;;; MAU_B_0038
                     5 3 ;;; MAU_B_0050
                     0.5 -70 )))))

(defun find-liquid-end-candidate (start-time end-time)
  (caar (caddr (USER::call-speechfunc
                (list "POWER-CHANGE"
                     start-time end-time
                     1000 4000
                     1 21
                     5 5
                     5 3 ;;; MAU_B_0050
                     0.5 -70 )))))
```

LM17: >hatazaki>sre>seg-liquid>seg-liquid.art.10

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/25/89 20:49:10

Queued at: 3/28/89 11:10:49



```
;;; -*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-mid-liquid-0
  "Finds a candidate of liquid /R/."
  (declare (saliency ?*find-candidate*))
  (segmenting liquid)
  (mid-liquid-region-candidate ?from ?to)
  (power-strength ?from ?to 1000 6000 ?lpw-1000-6000)
  (power-strength ?from ?to 4000 6000 ?lpw-4000-6000)
  (power-strength ?from ?to 0 500 ?lpw-0-500)
  (CF (mid-liquid-0-500-4000-6000-power-ratio =(- ?lpw-0-500 ?lpw-4000-6000))
    ?CF1pr-0-500-4000-6000&:mightbe-valid)
  (CF (mid-liquid-0-500-1000-6000-power-ratio =(- ?lpw-0-500 ?lpw-1000-6000))
    ?CF1pr-0-500-1000-6000&:mightbe-valid)
  (BB ?BB)
  =>
  (bind ?segment (gensym "SEG-"))
  (at ?BB
    (assert (liquid-candidate ?from ?to))
    (assert (schema ?segment (instance-of phoneme-segment)))
    (hypothesize
      (assert (category ?segment liquid))
      (assert (prop ?segment (candidate-loc ?from ?to))
        (segment-status ?segment segmentation)
        (CF (category ?segment liquid)
          =(CFand (CFcomb
            (CFunknown)
            (CFweight ?*evidence* ?CF1pr-0-500-1000-6000)
            (CFweight ?*evidence* ?CF1pr-0-500-4000-6000))
          ?CF1pr-0-500-1000-6000
          ?CF1pr-0-500-4000-6000))))))

(defrule get-mid-liquid-region-candidates
  (declare (saliency ?*top-down*))
  (goal (mid-liquid-region-candidate ?from&?? ?to&??))
  (explicit (not (mid-liquid-region-candidate ?from ?to)))
  =>
  (bind ?mid-liquid-region-candidates
    (get-mid-liquid-region-candidates))
  (for region in ?mid-liquid-region-candidates
    do (assert =(seq$ (cons 'mid-liquid-region-candidate region))))))
```

LM17: >hatazaki>sre>seg-liquid>seg-liquid-left. art. 8

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/31/89 17:00:13

Queued at: 3/28/89 11:11:33

```
;;;-*- Mode: ART; Package: art-user; Base:10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-mid-liquid-left
  ""
  (declare (salience ?*left-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment liquid)
  (CF (category ?segment liquid) ?CF-category&:mightbe-valid)
  (not (applied ?segment sg-liquid-left))
  (not (start-time ?segment ?))
  =>
  (assert (applied ?segment sg-liquid-left))
  (conflicting-alternatives
   (hypothesize
    (assert (left-context ?segment vowel))
    (assert (CF (left-context ?segment vowel) =(CFunknown))))))
```

```
;;;-----
```

```

(defrule sg-mid-liquid-left-vowel-1
  "Find time the preceding vowel starts."
  (declare (salience (+ ?*left-segmentation* 1)))
  (segment-status ?segment segmentation)
  (category ?segment liquid)
  (left-context ?segment vowel)
  ?x<-(CF (left-context ?segment vowel) ?CFleft-vowel)
  (not (applied ?segment sg-mid-liquid-left-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))

;;; search segmenetation position of left vowel
(power-decrease (?pos&~NONE ?value)
  before ?to 50 1000 4000 (5 3)
  ?*vowel-end-min-change-before-liquid*)
(CF (mid-liquid-left-power-change ?value) ?CFmid-liquid-left-power-change&: mightbe-valid)

;;; check charasterictics of left vowel
(power-strength =(- ?pos 20) ?pos 0 1000 ?vpw-0-1000)
(CF (vowel-by-liquid 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength =(- ?pos 20) ?pos 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-liquid 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
(power-strength =(- ?pos 20) ?pos 5000 6000 ?vpw-5000-6000)
(CF (liquid-vowel-power-ratio-0-1000-5000-6000 =(- ?vpw-0-1000 ?vpw-5000-6000))
  ?CFvpr-0-1000-5000-6000&: mightbe-valid)
=>
(assert (applied ?segment sg-mid-liquid-left-vowel-1))
(hypothesize
  (assert (prop ?segment (mid-liquid-start-time ?pos)))
  (retract ?x)
  (assert (CF (left-context ?segment vowel)
    =(CFand
      (CFcomb ?CFleft-vowel
        (CFweight ?*strong-evidence* ?CFmid-liquid-left-power-change)
        (CFweight ?*weak-evidence* ?CFvpw-0-1000)
        (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
        (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
      ?CFmid-liquid-left-power-change
    )))))

(defrule sg-mid-liquid-left-vowel-done-1
  ""
  (declare (salience ?*left-segmentation-end*))
  (segment-status ?segment segmentation)
  (category ?segment liquid)
  (CF (category ?segment liquid) ?CF-category&: mightbe-valid)
  (left-context ?segment vowel)
  (CF (left-context ?segment vowel) ?CF-context&: mightbe-valid)
  (not (applied ?segment sg-mid-liquid-left-vowel-done-1))
  (prop ?segment (mid-liquid-start-time ?pos))
=>
(assert (applied ?segment sg-mid-liquid-left-vowel-done-1))
(hypothesize
  (assert (start-time ?segment ?pos))
  (assert (CF (start-time ?segment ?pos) =(CFvalue ?CF-context))))

```

LM17: >hatazaki >sre>seg-liquid>seg-liquid-right.art. 9

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/02/89 15:52:34

Queued at: 3/28/89 11:12:14

```
;;;-*- Mode: ART; Package: art-user; Base: 10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-mid-liquid-right
  "Hypothesizes right phoneme."
  (declare (salience ?*right-segmentation*))
  (viewpoint ?v
    (segment-status ?segment segmentation)
    (category ?segment liquid)
    (CF (category ?segment liquid) ?CF-category&:mightbe-valid))
  ;;;;
  (viewpoint ?
    (exists (start-time ?segment ?)))
  ;;;;
  (not (applied ?segment sg-mid-liquid-right))
  (not (end-time ?segment ?))
  =>
  (at ?v
    (assert (applied ?segment sg-mid-liquid-right))
    (conflicting-alternatives
      (hypothesize
        (assert (right-context ?segment vowel))
        (assert (CF (right-context ?segment vowel) =(CFunknown)))))))
```

```
;;;-----
```

```

(defrule sg-mid-liquid-right-vowel-1
  "Find time the following vowel starts."
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment liquid)
  (right-context ?segment vowel)
  ?x<-(CF (right-context ?segment vowel) ?CFright-vowel)
  (not (applied ?segment sg-mid-liquid-right-vowel-1))
  (prop ?segment (candidate-loc ?from ?to))

;; search segmenetation position of right vowel
(power-increase (?pos&-NONE ?value)
  after ?from 50 1000 4000 (5 3)
  ?*vowel-start-min-change-after-liquid*)
(CF (mid-liquid-right-power-change ?value) ?CFmid-liquid-right-power-change&: mightbe-valid)

;; check characteristics of right vowel
(power-strength ?pos =(+ ?pos 20) 0 1000 ?vpw-0-1000)
(CF (vowel-by-liquid 0-1000-power ?vpw-0-1000) ?CFvpw-0-1000&: mightbe-valid)
(power-strength ?pos =(+ ?pos 20) 1000 6000 ?vpw-1000-6000)
(CF (vowel-by-liquid 1000-6000-power ?vpw-1000-6000) ?CFvpw-1000-6000&: mightbe-valid)
(power-strength ?pos =(+ ?pos 20) 5000 6000 ?vpw-5000-6000)
(CF (liquid-vowel-power-ratio-0-1000-5000-6000 ?vpw-0-1000 ?vpw-5000-6000)
  ?CFvpr-0-1000-5000-6000&: mightbe-valid)
=>
(assert (applied ?segment sg-mid-liquid-right-vowel-1))
(hypothesize
  (assert (prop ?segment (mid-liquid-end-time ?pos)))
  (retract ?x)
  (assert (CF (right-context ?segment vowel)
    =(CFand
      (CFcomb ?CFright-vowel
        (CFweight ?*strong-evidence* ?CFmid-liquid-right-power-change)
        (CFweight ?*weak-evidence* ?CFvpw-0-1000)
        (CFweight ?*weak-evidence* ?CFvpw-1000-6000)
        (CFweight ?*evidence* ?CFvpr-0-1000-5000-6000))
      ?CFmid-liquid-right-power-change
    )))))

(defrule sg-mid-liquid-right-vowel-done-1
  ""
  (declare (salience ?*right-segmentation*))
  (segment-status ?segment segmentation)
  (category ?segment liquid)
  (CF (category ?segment liquid) ?CF-category&: mightbe-valid)
  (right-context ?segment vowel)
  (CF (right-context ?segment vowel) ?CF-context&: mightbe-valid)
  (not (applied ?segment sg-mid-liquid-right-vowel-done-1))
  (prop ?segment (mid-liquid-end-time ?pos))
  =>
  (assert (applied ?segment sg-mid-liquid-right-vowel-done-1))
  (hypothesize
    (assert (end-time ?segment ?pos))
    (assert (CF (end-time ?segment ?pos) =(CFvalue ?CF-context))))))

```

LM17:>hatazaki>sre>seg-liquid>seg-liquid-end.art.6

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 1/20/89 14:25:02

Queued at: 3/28/89 11:13:00



```
;;;-*- Mode:ART; Package:art-user; Base:10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
;;; check liquid duration time
(defrule sg-mid-liquid-duration-time
  ""
  (declare (salience (+ ?*end-segmentation* 1)))
  (not (applied ?segment sg-mid-liquid-duration-time))
  (join
    (category ?segment liquid)
    ?x<-(CF (category ?segment ?liquid) ?CFcategory))
  (join
    (start-time ?segment ?start)
    (explicit (CF (start-time ?segment ?start) ?CFstart)))
  (join
    (end-time ?segment ?end)
    (explicit (CF (end-time ?segment ?end) ?CFend)))
  (CF (mid-liquid-segmenting-duration-time =(- ?end ?start)) ?CFduration)
  (power-decrease (?s-pos&~NONE ?s-value)
    before ?to 50 1000 4000 (5 3)
    ?*vowel-end-min-change-before-liquid*)
  (power-increase (?e-pos&~NONE ?e-value)
    after ?from 50 1000 4000 (5 3)
    ?*vowel-start-min-change-after-liquid*)
  (CF (mid-liquid-inc-dec-value =(+ ?s-value ?e-value)) ?CFinc-dec-value)
  =>
  (retract ?x)
  (assert (applied ?segment sg-mid-liquid-duration-time))
  (assert (CF (category ?segment liquid)
    =(CFand (CFcomb ?CFcategory
      (CFweight ?*strong-evidence* ?CFduration)
      (CFweight ?*strong-evidence* ?CFinc-dec-value))
    ?CFduration
    ))))
```

LM17: >hatazaki>sre>rule>seg-end. art. 16

For: Kaichiro Hatazaki

Printed on: LGP8-J ON ATR-LM01

Number of copies: 1

Data created at: 2/02/89 18:46:30

Queued at: 3/28/89 10:25:42

```
;;-*- Mode:ART; Package:art-user; Base:10; Default-character-style: (:FIX :ROMAN :NORMAL) -*-
```

```
(defrule sg-end-1
  ""
  ;
  ;
  ; ;; [hatazaki 121] Checks that there is another boundaries which has larger CF value.
  ; ;;
  ; (declare (salience ?*end-segmentation*))
  ; ?x <- (segment-status ?segment segmentation)
  ; ;;(not (applied ?segment sg-end-1))
  ; (viewpoint ?v
  ;   (start-time ?segment ?start)
  ;   (explicit (CF (start-time ?segment ?start) ?CFstart&:(?CFstart >= 0.2)))
  ;   (end-time ?segment ?end)
  ;   (explicit (CF (end-time ?segment ?end) ?CFend&:(?CFend >= 0.2))))
  ; (viewpoint ?
  ;   (not (exists (start-time ?segment ?another-start)
  ;                 (explicit (CF (start-time ?segment ?another-start)
  ;                               ?CFanother-start&:(?CFanother-start > ?CFstart))))))
  ; (viewpoint ?
  ;   (not (exists (end-time ?segment ?another-end)
  ;                 (explicit (CF (end-time ?segment ?another-end)
  ;                               ?CFanother-end&:(?CFanother-end > ?CFend))))))
  ; =>
  ; (at ?v
  ;   (assert (applied ?segment sg-end-1))
  ;   (retract ?x)
  ;   (assert (segment-status ?segment segmentation-ended))))
```

```
(defrule sg-end-1
  ""
  ;;
  ;; [hatazaki 143] CF of (segment-result ?segment (?start ?end)),
  ;;   =(CFcomb ?CFstart ?CFend) -->
  ;;   =(CFcomb (CFweight ?*evidence* ?CFcategory)
  ;;            (CFweight ?*evidence* ?CFstart)
  ;;            (CFweight ?*evidence* ?CFend))
  ;;
  ;; komori 1988.10.28 (FRI) insert CF into segmentation
  ;;
  ; (declare (salience ?*end-segmentation*))
  ; ;;(not (applied ?segment sg-end-1))
  ; (join
  ;   (category ?segment ?category)
  ;   (CF (category ?segment ?category) ?CFcategory&: mightbe-valid))
  ; (join
  ;   (start-time ?segment ?start)
  ;   (explicit (CF (start-time ?segment ?start) ?CFstart&: mightbe-valid)))
  ; (join
  ;   (end-time ?segment ?end)
  ;   (explicit (CF (end-time ?segment ?end) ?CFend&: mightbe-valid)))
  ; (test (> ?end ?start))
  ; (not (segment-result ?segment ?))
  ; =>
  ; (assert (applied ?segment sg-end-1))
  ; (assert (segment-result ?segment (?start ?end)))
  ;;
  ;; [hatazaki 168]
  ;;   =(CFcomb (CFunknown)
  ;;            (CFweight ?*strong-evidence* ?CFcategory)
  ;;            (CFweight ?*evidence* ?CFstart)
  ;;            (CFweight ?*evidence* ?CFend))
  ;; <--
  ;;   =(CFcomb (CFunknown)
  ;;            (CFweight ?*evidence* ?CFcategory)
  ;;            (CFweight ?*weak-evidence* ?CFstart)
  ;;            (CFweight ?*weak-evidence* ?CFend))
  ;;
  ; (assert (CF (segment-result ?segment (?start ?end))
  ;            =(CFcomb (CFunknown)
  ;                      (CFweight ?*strong-evidence* ?CFcategory)
  ;                      (CFweight ?*evidence* ?CFstart)
  ;                      (CFweight ?*evidence* ?CFend))))))
```

```
(defrule sg-end-2
  ""
```

```
;;
```

```

;; maximum CF segmentation
;;
(declare (salience (- ?*end-segmentation* 1)))
(viewpoint ?v
  (segment-result ?segment ?)
  (CF (segment-result ?segment ?) ?CF)
  ?x <- (segment-status ?segment segmentation)
  (not (applied ?segment sg-end-2)))

(viewpoint ?
  (not (exists (segment-result ?segment ?)
    (explicit (CF (segment-result ?segment ?)
      ?CFanother&:(CFgreaterp ?CFanother ?CF))))))

=>
(at ?v
  (assert (applied ?segment sg-end-2))
  (retract ?x)
  (assert (segment-status ?segment segmentation-ended)))

(defrelation conflicting-segments (?segment1 ?segment2))

(defrule sg-end-3
  "Select best result."
  ;;
  ;;[hatazaki 178] Rule sg-end-3;
  ;; Uses "(viewpoint .." not to merge the viewpoints,
  ;; and asserts "(conflicting-segments ..." in each viewpoint.
  ;; conflicting-segments is not a symmetric relation.
  ;; OLD ... (defrelation conflicting-segments (?segment1 ?segment2) symmetric)
  ;;
  ;;
  ;;[hatazaki 149] Newly defined to select the best result among more than one results for
  ;; same location.
  ;;
  (declare (salience ?*finish-segmentation*))

  (viewpoint ?v1
    ;[hatazaki 178]
    (segment-status ?segment1 segmentation-ended)
    (join
      (segment-result ?segment1 (?start1 ?end1))
      (CF (segment-result ?segment1 (?start1 ?end1)) ?CFsegment1)))

  (viewpoint ?v2
    ;[hatazaki 178]
    (segment-status ?segment2&~?segment1 segmentation-ended)
    (not (conflicting-segments ?segment2 ?segment1))
    (join
      (segment-result ?segment2 (?start2 ?end2))
      (CF (segment-result ?segment2 (?start2 ?end2)) ?CFsegment2)))

  ;;(not (conflicting-segments ?segment2 ?segment1))

  (test (or (and (< ?start1 (/ (+ ?start2 ?end2) 2))
    (< (/ (+ ?start2 ?end2) 2) ?end1))
    (and (< ?start2 (/ (+ ?start1 ?end1) 2))
    (< (/ (+ ?start1 ?end1) 2) ?end2))))

  =>
  ;;(assert (conflicting-segments ?segment1 ?segment2))
  (at ?v1
    ;[hatazaki 178]
    (assert (conflicting-segments ?segment1 ?segment2)))
  (at ?v2
    ;[hatazaki 178]
    (assert (conflicting-segments ?segment2 ?segment1))))

(defrule sg-end-4
  ""
  (declare (salience (- ?*finish-segmentation* 1)))

  (viewpoint ?v
    ?x <- (segment-status ?segment1 segmentation-ended)
    (CF (segment-result ?segment1 (?start1 ?end1)) ?CFsegment1))

  (viewpoint ?
    (not (exists (conflicting-segments ?segment2 ?segment1)
      (CF (segment-result ?segment2 (?start2 ?end2))
        ?CFsegment2&:(CFgreaterp ?CFsegment2 ?CFsegment1))))))

```

```
=>  
(at ?v  
  (retract ?x)  
  (assert (segment-status ?segment1 segmentation-result)))  
)
```