Verbundvorhaben

Final report for Verbmobil Teilprojekt 4.4 English Synthesis

Simon King

IKP Universität Bonn

Report 195
Januar 1997

Januar 1997

Simon King
Institut für Kommunikationsforschung und Phonetik
Universität Bonn
Poppelsdorfer Allee 47
53115 Bonn

Tel.: (0228) 7356 - 13
Fax: (0228) 7356 - 39
e-mail: {ski}@ikp.uni-bonn.de

Gehört zum Antragsabschnitt: 4.4

Die vorliegende Arbeit wurde im Rahmen des Verbundvorhabens Verbmobil vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) unter dem Förderkennzeichen 01 IV 101 D 08 gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei dem Autor.

Contents

List of Figures

Introduction

This is the final report for work carried out by Simon King from January 1996 to October 1996 for the Verbmobil project, Teilprojekt 4.4 (English synthesis).

What does this document cover ?

It describes the algorithms for unit selection and prosody generation. The flexibility of the solution is shown, as are any assumptions, simplifications and limitations.

What does this document NOT cover ?

This document does not describe the software written to implement the chosen algorithms or how to use the software; that is given in [7]. The inventory design and recording is described separately in [6].

Acknowledgements

I would like to thank Thomas Portele for his support, Paul Taylor at CSTR for his help and advice, and Florian Höfer for his hard work.

Contact Information

The author is now at The Centre for Speech Technology Research, and can be contacted by email : `Simon.King@ed.ac.uk` or at :

Simon King,
C.S.T.R.,
University of Edinburgh,
80, South Bridge,
Edinburgh EH1 1HN
Scotland, G.B.

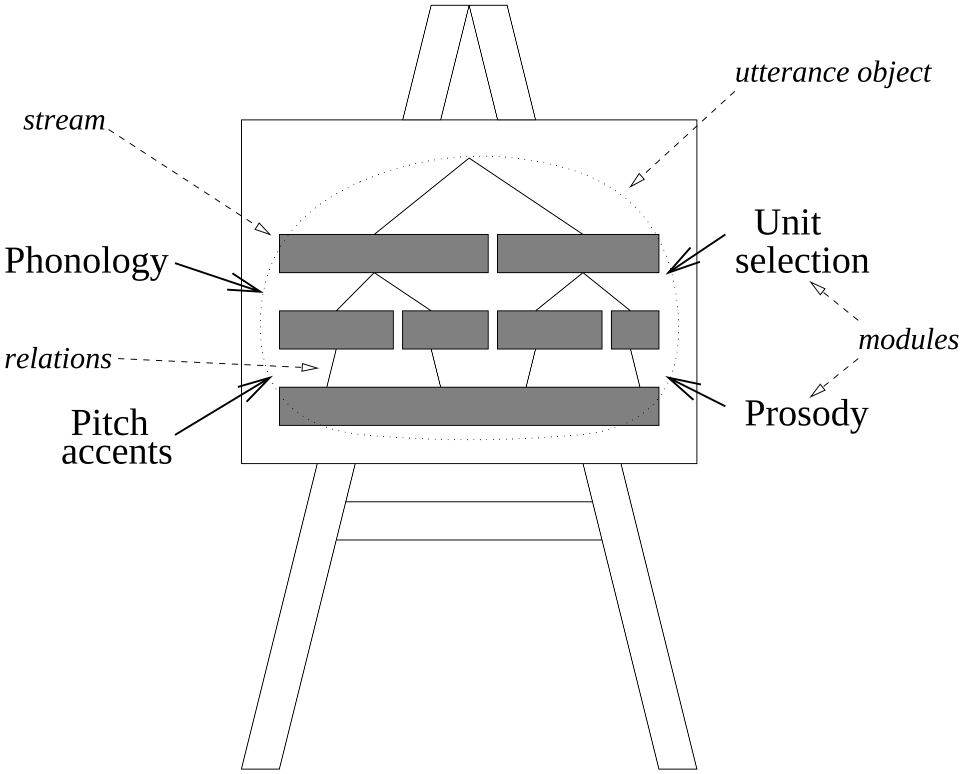

Figure 1: The blackboard architecture

1 Preliminaries

1.1 Research

This work draws heavily from a number of sources. The architecture is similar to that in [3], partly because the software implementation uses [12]. The reduced word class mapping and word accent labels come directly from [5]. The assignment of pitch accents is similar to that in [2]; realisation of pitch accents using a *tilt* representation is taken from work by Taylor [11]. The unit inventory structure, described in detail in [6], comes from [9], although the selection algorithm is new.

No synthesizer would be complete without reference to [1], from which the phoneme duration rules are taken (and slightly modified). There are of course many more sophisticated duration systems, but, in the available time, the simplicity of the Klatt rules was preferred.

1.2 A framework for the algorithms

From the start, it was decided to design and implement the algorithms in as flexible a framework as possible. A *blackboard architecture* (figure 1) provides such a framework.All algorithms operate on a common object — the *utterance*

Main algorithm

repeat

read a single utterance(sentence) from the input

do within-word phonology

construct phoneme stream from word pronunciations

assign *word accents* according to word class

do cross-word phonology

find syllable nuclei

from *word accents*, assign *syllable accents*

according to sentence modality

do further phonology (vowel reduction)

determine phoneme durations and boundary pause durations

select units from inventory

generate F0

create baseline

realise syllable accents as pitch accents

complete connections between accents

output

until no more input

Figure 2: The main algorithm

— a concept also used in [3].

The *utterance* contains a set of *streams* which are sequences of, for example, phonemes or syllables. The items in the streams have *relations* to items in other streams; these *relations* show, for example, which phonemes belong to which syllable. This concept translates directly into a software implementation, described in [7].

1.3 Division of the task

The blackboard architecture allows each subtask to be treated independently. The top level algorithm is shown in figure 2.

Each step modifies the *utterance*, perhaps creating a new stream or modifying relations between streams. The descriptions of the various steps follow.

2 Input processing

The input format is *EI* (Erweiterte Informationen, [8]), which is simply marked up text. Each word has a pronunciation, with syllable boundaries and lexical stress marked; for example:

```
{Transcription:fraI[31]|dI[12]} {WordClass:N} friday
```

Early versions of the lexicon used by the text generation module did not have syllable boundaries marked, and a simple syllabification algorithm was used. However, this has been removed in the final version. Other information provided in the input and used by this module is : sentence modality, phrase boundaries, word class, focus and (in a possible future version) extra information such as given/new, phrasal verb, local/global focus and so on. As input is read, word, syllable and phoneme streams are created, and relations between them made.

2.1 Finding syllable nuclei

Since syllable boundaries are marked in the word pronunciations, each syllable nucleus is *assumed* to be the first vowel or syllabic consonant in each syllable. The syllable nuclei will be used in both the prosodic (section 4) and unit selection (section 5) algorithms. Subsequent vowels or syllabic consonants in the same syllable are not considered as potential nuclei.

3 Phonology

Phonological phenomena are modelled in several parts. The first two, within- and cross-word phonology, are concerned with phenomena expressed as *rules*. The other two parts, vowel reduction and plosive suppression, are treated as special cases, and are *"hard wired"* into the code.

3.1 Rules

3.1.1 Within-word phonology

Within-word phenomena (those only dependent on contextual influences within the same word) are applied directly to word pronunciations. An example of such an effect is the conversion of `@ l` to the syllabic consonant `=l` when following a dental.

3.1.2 Cross-word phonology

Cross-word effects must be applied after phrase boundaries have been determined, since cross word effects are assumed not to occur across pauses. Effects such as devoicing word final `z` when followed by `s` are covered by the cross-word rules.

Appendix B gives a full list of all within- and cross-word phonological rules used.

3.2 Special cases

3.2.1 Vowel reduction

Short, unrounded vowels which are fairly central (that is, they are not *both* back and low, or *both* front and high) and occur in words marked "cliticize" (see section 4.1.1) are reduced to schwa. This is a major simplification, but modelling of more subtle effects, such as partial reduction (to a vowel *closer* to schwa) is far from straightforward in a concatenative synthesis system. As a result of this choice, the frequency of reductions is limited, and therefore there is a tendency to produce over-articulated speech. This is acceptable in this application, where intelligibility is more important than naturalness.

3.2.2 Plosive suppression

Unvoiced plosives which are followed by another unvoiced plosive in the same clause do not get released.

4 Prosody

ACCENT ASSIGNMENT

for each *word* **in** *sentence* **do**
    map word class label to one of a reduced set
    apply Hirschberg word accent assignment algorithm
    use IFT rule to assign word accents, given Hirschberg label
    for each *syllable* **in** *word* **do**
        assign syllable accent from accents assigned to this word
        realise syllable accent using its tilt parameter definition

---

Figure 3: The accent assignment algorithm

4.1 Accents

The algorithm for accent assignment is given in figure 3 and an example is shown in figure 4.

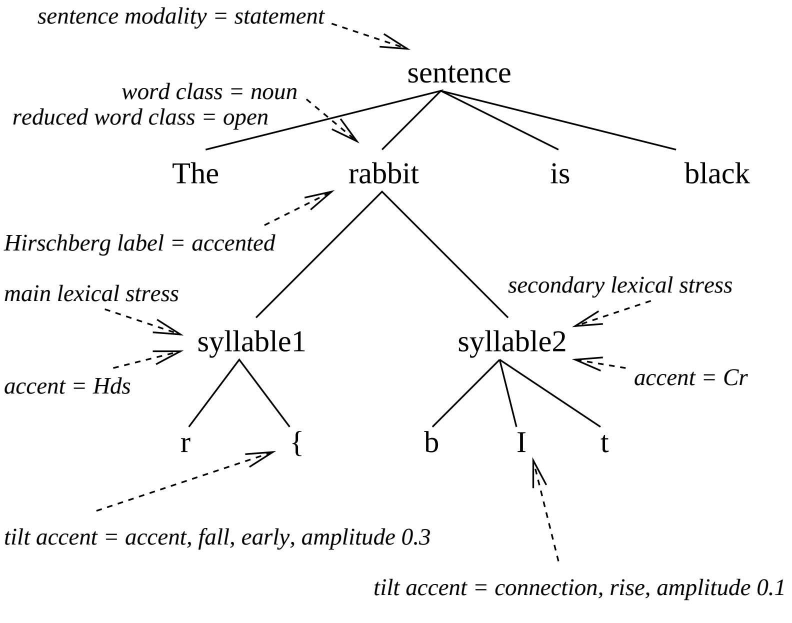

Figure 4: Example of accent assignment

4.1.1 Word-level

At the word level, the word–class labels are mapped to the reduced set {*closed–cliticized,closed–deaccent,closed–accent,open*} . The simple 1-to-1 mapping follows the scheme in [5] and is given in appendix C.

Hirschbergs algorithm [5] is used to give each word a label from the set { *cliticize, deaccent, accent, emphatic* } using the reduced word class label and further information such as focus, whether a verb is phrasal and so on[1]. This is the stage at which any extra information available in the input is used to guide accent type and placement.

4.1.2 Syllable level

The Hirschberg word labels are now used to make syllable accent assignments. This process is dependent on the *Illocutionary Force Type* of the phrase (or sentence). There are a set of rules giving the word accent to syllable accent mapping for each of the three *IFT*s used : *Statement*, *YN–question* and *Wh–question*. Figure 5 shows one such rule.

Statement:	**START**	main=Hds, secondary=Cr
	accent	main=Hds, secondary=Cr
	emphatic	main=Hemph, secondary=Cr
	PhraseTAIL	last=CrTail, main=Hds, secondary=Cr
	TAIL	last_stressed=HdsTail, last=Cf, main=Hds, secondary=Cr

Figure 5: One of the *IFT* rules

Syllable stress is treated as a scalar ranging from 0 to 1. A simple conversion to the boolean values Klatt calls *stress* and *2-stress* is acheived thus:

$$\begin{aligned} \textit{scalar stress} &> 0 & \Rightarrow & \;\; \textit{2-stress} \\ \textit{scalar stress} &> 0.5 & \Rightarrow & \;\; \textit{stress} \end{aligned}$$

where *stress* and *2-stress* are used *only* in the Klatt duration rules.

4.2 Boundary pauses

Boundary pause durations are calculated from the `BorderProminence` tag values according to the formula :

$$\textit{duration} = \textit{border prominence} \times 50ms$$

This is an *ad hoc* attempt to linearise the relationship between prominence and duration which works reasonably well at normal speech rates, but has limitations.

[1]only focus is currently fully implemented because the text-generation module does not yet give any further information

In particular, the phenomenon of shorter pauses disappearing at higher speech rates is not modelled. This is not considered a great restriction because high speech rates are not anticipated to be of much use in this application. Border prominence values of 1 and 9 correspond to pauses of 50 and 450ms respectively.

Of course, the question of what a `BorderProminence` value of, say, 4 *means* depends on the text generation module. Roughly speaking, a value of 1 might be used after noun phrases, 2 and 3 to indicate punctuation, 4 or 5 for breath pauses and 9 for a sentence break.

4.3 Duration

Segment durations are simply calculated using the first 10 Klatt duration rules [1, pages 95–96]. The minimum and inherent durations required by these rules are given in a resource file; their values were also taken from [1], but modified slightly for British English.

Klatts rule 8 lengthens “emphasized” vowels. This rule was slightly modified (figure 6) to vary the degree of lengthening depending on the degree of stress (on a scale $0 \rightarrow 1$) marked on the syllable of the vowel. Pause duration is also handled by the Klatt routine, using the formula in section 4.2.

RULE 8

if this segment is a vowel or a syllabic consonant **and** this syllable is accented
then lengthen by a factor $1 + 0.5 \times \textit{syllable stress}$

Figure 6: Modified Klatt rule 8

4.4 F0

4.4.1 Framework

The observed phenomenon of *downdrift* — F0 being lower at the end of a phrase than at the start — is acheived mainly through *downstep* — pitch accents ending at a lower F0 than they started — and only partially through *declination* — the gradual lowering of F0 through a phrase, due to decreasing air flow/pressure.

The chosen route to constructing the F0 contour was to determine a baseline, place accents on or about the baseline, and make connections between the accents. Downdrift was thus initially modelled by the downward sloping baseline but acheived mostly by pitch accents (typically downstepping) placed on the baseline. Connections are made between the end of one accent and the start of the next. The pitch accents and connections then form the target F0 contour.

The choice of this framework for constructing the F0 contour was motivated by [4], [11], [10] and [2].

4.4.2 Baseline

The treatment of the baseline is very simple. Start and finish values for F0 are set for the entire sentence, and a simple, linear, declining baseline is constructed. Phrase initial reset positions are set at phrase boundaries, but their sizes are calculated *after* the pitch accents have been determined (see section 4.4.3).

4.4.3 Pitch accents

Syllable accent		Tilt definition of H_{ds}		Parametric description
		+/− rise		rise amplitude
		+/− fall		fall amplitude
Name	→	+/− late	→	rise duration
e.g. H_{ds}		+/− early		fall duration
		scalar amplitude		peak position
		vertical placement		vertical placement

Figure 7: Pitch accent representations

Pitch accents are realised in two steps : parametric and discrete. In the first step, the *tilt accent definition* of the accent assigned to each (accented) syllable is used to compute a parametric representation of each accent (figure 7). The syllable nucleus duration is used in calculating the parametric representation.

Some scaling is applied to the parametric representations. If the total downstep due to *downstepping* accents is greater then the declination for the phrase, the accent amplitudes are scaled down, within limits.

The parametric representations are now used to generate discrete F0 targets. For each phrase, a new declining baseline is calculated whose initial F0 is the global baseline F0 at that time, plus a phrase-initial reset. The reset is taken as the total amount of downstep in the accents in the phrase, limited to some pre-determined amount. The phrase-final F0 is the global baseline F0 at that time.

4.4.4 Resets

The size of phrase-initial resets is determined after accents have been constructed. This is because the size of the reset depends on the amount of downstepping which takes place in the phrase. Some scaling of the accent amplitudes may be necessary is the case of phrases with many downstepping accents, since there is a limit on the size of phrase-initial resets.

Phrase reset algorithm

for each *phrase*

Compute all accents for this phrase

Add up the total downstep in the accents

Set phrase reset to this value*

if phrase reset is too large **then**

limit phrase reset amplitude

scale down accent amplitudes**

Construct linear downward sloping baseline for this phrase

* phrase resets are further scaled down according to their distance from the start of the sentence. This models the observation that reset amplitudes (and accent amplitudes) decrease through an utterance.

** the allowable amount of scaling is also limited

Figure 8: Phrase reset algorithm

4.4.5 Connections

The pitch accents are simply connected by linear sections of pitch contour. Connections starting or ending at phrase boundaries have one end point on the baseline, with the restriction that phrase-initial connections have non-positive slope.

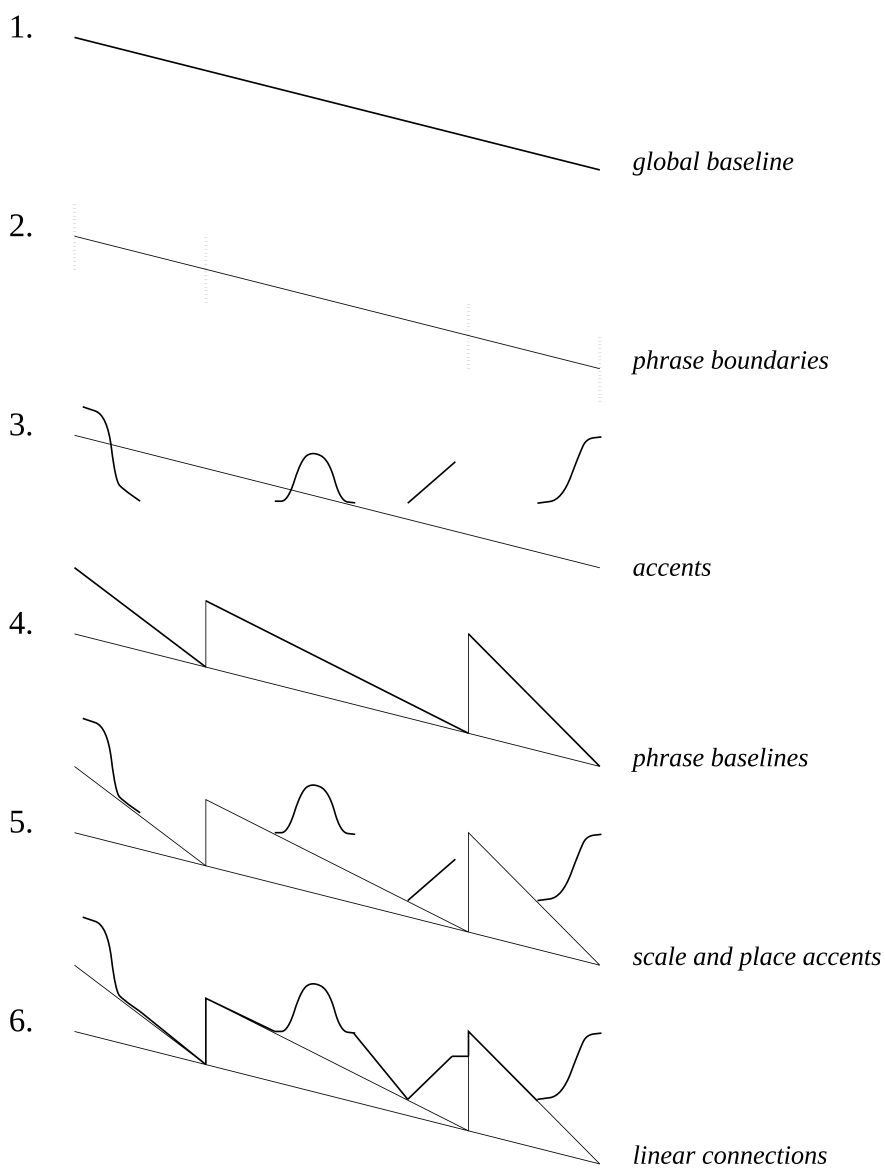

Figure 9: Pitch accent realisation

5 Unit selection

The HADIFIX [9] inventory structure means that the choice of unit sequence for a given phoneme sequence is neither unique nor trivial (as would be the case for a diphone system).

5.1 Requirements

The key features required of the selection algorithm are listed in figure 10.

The most important of these requirements (figure 10, point 1) means that a unit sequence should be found even if there is only one unit in the inventory

1. a unit sequence is found for any and all phoneme sequences
2. the units selected contain the required phonemes in contexts like, or as similar as possible to, those in the phoneme sequence
3. the units are joined in such a way as to minimise the (perceived) discontinuity

Figure 10: Desirable features of a unit selection algorithm

containing each phoneme, however poorly it fits the context. This motivated the choice of a *scoring* system for unit selection, in which all candidate units are given a score, and the best scoring candidate is chosen.

Figure 10, point 2, is satisfied by devising a scoring system which gives appropriate weight to each of the desirable features listed in figure 11.

Figure 11, points 1 to 5, are expressed directly in the rules, and 6 is expressed indirectly by penalising units with more phonemes. It is important to note that the score is used to differentiate between units, rather than give an *absolute* measure of "goodness of fit".

1. exactly matching right context
2. broadly matching right context
3. exactly matching left context
4. broadly matching left context
5. appropriate unit type
6. no poorly matching context, right or left

Figure 11: Desirable phonetic contexts for the selected unit

5.2 Rule system

The scoring system is expressed as a simple decision tree, and is given in full in appendix A. The system appears complicated because of the need to deal with special cases (phrase–initial and –final phonemes) and the fact that points awarded for matches at the second phoneme (left or right) are conditional on a match at the first (adjacent) phoneme.

r	{	$\boxed{\text{b}}$	I	t	*phoneme sequence*
					candidate units
	{	b			{b
		b	I		bI
		b	I	s	bIs
		b	I	n	bIn

Figure 12: Unit selection

5.3 Example

As an example, consider the case in figure 12. The phoneme `b` is neither phrase intitial or final. The candidate unit `{ b` has one exact match to the left, and so scores 2.98 points. The unit `b I` has one exact match to the right and scores 9.98 points; the unit `b I s` has an additional context match at the second phoneme to the right and so scores 11.47 points. The unit `b I n` scores similarly to `b I` but is penalised for the extra phoneme `n` and so scores 9.97 points. The chosen unit is `b I s`.

5.4 Multiple alignments

If a unit contains the central phoneme more than once, multiple alignments are possible. The scoring system is used to score each possible alignment, and the highest scoring one is chosen.

5.5 Unit concatenation

Each unit pair is considered in turn and the type and position of the join is decided. The details of how and why the two types of join – hard and soft – are made are given in [6]. The pseudo unit for silence always has hard concatenation joins on both the left and right since the inventory contains no specific units containing silence.

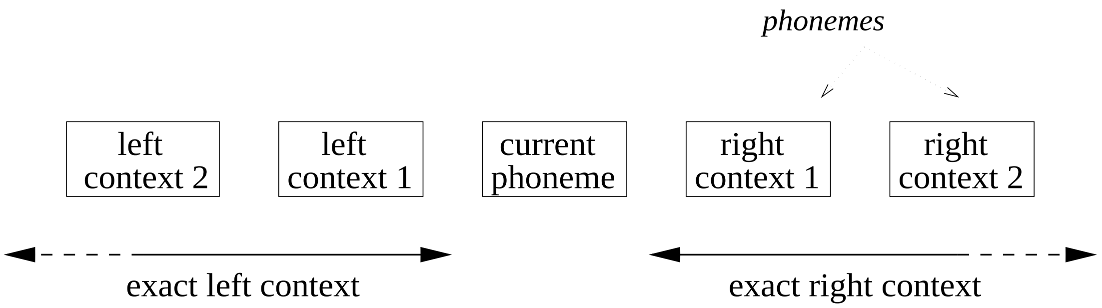

Figure 13: Phonetic context

A Scoring system

Definitions

Figure 13 shows the surrounding phonemes used in comparing units. The exact left (or right) context is the number of exactly matching phonemes (up to a maximum of 3) to the left (or right). Broad context refers to particular positions, as shown in the figure. In the algorithm given below, the notation is:

elc	exact left context
$blc1$	broad left context at 1st phoneme to the left
$blc2$	broad left context at 2nd phoneme to the left
$brc1$	broad right context at 1st phoneme to the right
	..... etc.

where the broad contexts (e.g. $brc2$) are scores based on the contextual similarity of the phoneme at that position in the unit to the phoneme at that position in the utterance (see [6]).

For this algorithm *phrase initial* means *breath phrase initial*, that is, preceding phoneme is silence; likewise for *phrase final*.

Scoring system

if *phrase initial* **and** *phrase final* **then**
 if *any* left context **or** *any* right context **then**
  return -100
 else
  return 100
else if *phrase initial* **then**
 if *any* left context **then**
  return -100
 else
  if unit type = *initial demisyllable* **then**
   $score = score + 30$
  if $erc > 1$

```
            score = score + 15
        else if erc > 0
            score = score + 10
            if brc2 > 0
                score = score + 3 + brc2
                if brc3 > 0
                    score = score + brc3
        else
            if brc1 > 0
                score = score + 4 + brc1
                if brc2 > 0
                    score = score + 1 + brc2
                    if brc3 > 0
                        score = score + brc3
            else if arc > 0
                score = score + 1
else if phrase final then
    if any left context then
        return −100
    else
        if unit type = final demisyllable or suffix then
            score = score + 30
        if elc > 2
            score = score + 25
        else if elc > 1
            score = score + 15
                if blc3 > 0
                    score = score + blc3
        else if elc > 0
            score = score + 10
            if blc2 > 0
                score = score + 3 + blc2
                if blc3 > 0
                    score = score + blc3
        else
            if blc1 > 0
                score = score + 4 + blc1
                if blc2 > 0
                    score = score + 1 + blc2
                    if blc3 > 0
                        score = score + blc3
            else if alc > 0
                score = score + 1
else
    if erc > 2
        score = score + 25
    else if erc > 1
```

```
        score = score + 15
        if brc3 > 0
            score = score + brc3
    else if erc > 0
        score = score + 10
        if brc2 > 0
            score = score + 1 + brc2
            if brc3 > 0
                score = score + brc3
    else
        if brc1 > 0
            score = score + 4 + brc1
            if brc2 > 0
                score = score + brc2
                if brc3 > 0
                    score = score + brc3


    if elc > 2
        score = score + 4 + elc
    else if elc > 1
        score = score + 3 + elc
        if blc3 > 0
            score = score + blc3
    else if elc > 0
        score = score + 4 + elc
        if blc2 > 0
            score = score + blc2
            if blc3 > 0
                score = score + blc3
    else
        if blc1 > 0
            score = score + 1 + blc1
            if blc2 > 0
                score = score + blc2/10
                if blc3 > 0
                    score = score + blc2/20


score = score − number of phones in unit/100
return score
```

B Phonological rules

B.1 Within-word

```
; rules have the form :
;rule name      (oldphonemes leftcontext_rightcontext newphonemes)
@l1             (@l n_ =1)
@l2             (@l t_ =1)
@l3             (@l d_ =1)
@l4             (@l s_ =1)
@l5             (@l z_ =1)

@n1             (@l n_ =1)
@n2             (@l t_ =1)
@n3             (@l d_ =1)
@n4             (@l s_ =1)
@n5             (@l z_ =1)
```

B.2 Cross-word

```
;rule name      (oldphonemes newphonemes)
; difficult across-word sequences
; these rules produce over-articulation
; -------------------------------------
kt1             (kt|t kt#|t)
St1             (St|t St#|t)
Sd1             (Sd|t Sd#|t)
dk1             (d|k d#|k)
tk1             (t|k t#|k)
;pt1            (p|t p#|t)

; /r/ insertion
; -------------
A:r1            (A:|@ A:r|@)
A:r2            (A:|{ A:r|{)

{r1             ({|@ {r|@)
{r2             ({|{ {r|{)

@r1             (@|@ @r|@)
@r2             (@|{ @r|{)

O:r1            (O:|@ O:r|@)
O:r2            (O:|{ O:r|{)
```

```
; devoicing z
; -----------
z1              (z|s s|s)

; testing
tmp             (d|b d#|b)
```

C Word class mapping

```
; assign word classes to one of four broad categories :
; closed_cliticized closed_deaccented closed_accented open

; - cliticized means the stressed syllable of
;   the word can have its vowel reduced
; -  closed_* correspond to function words
; -  open     correspond to content words

DEFAULT open
SILENCE closed_deaccented
DET closed_cliticized
NUM open
NUMADJ closed_accented
ORD open
ADJ open
RELPRON closed_deaccented
V open
COORD closed_deaccented
PREP closed_deaccented
N open
NOUN open
NAME open

;such as "which"
WH-DET closed_accented

;personal pronoun
PPRON closed_accented

;demonstrative pronoun
DPRON closeD_accented

;conjunction
CONJ closed_deaccented
ADV open
PREFIX open
```

```
;possessive pronoun
POSS closed_deaccented

;interrogative pronoun
IPRON closed_accented

PARTICLE closed_accented
NEG open
VINF1 open
VINF2 open
VINF3 open

;interjection
INTERJ closed_deaccented

;auxiliary verb
AUX closed_deaccented

;modal verb
MODAL closed_deaccented

MINUTE_WORD open
HOUR_PREP_WORD closed_deaccented
```

References

[1] Jonathan Allen, M. Sharon Hunnicut, and Dennis Klatt. *From text to speech : The MITalk system.* Cambridge University Press, 1987.

[2] Alan Black and Paul Taylor. Assigning intonation elements and prosodic phrasing for English speech synthesis from high level linguistic input. In *Proc. ICSLP '94*, Yokohama, Japan, 1994.

[3] Alan Black and Paul Taylor. CHATR : a Generic Speech Synthesis System. In *Proc. COLING94*, Kyoto, Japan, 1994.

[4] Alan Black and Paul Taylor. A framework for generating prosody from high level linguistic descriptions. In *Spring meeting of the Acoustical Society of Japan*, 1994.

[5] Julia Hirschberg. Using discourse content to guide pitch accent decisions in synthetic speech. In G. Bailly and C. Benoit, editors, *Talking Machines*, pages 367–376. North-Holland, 1992.

[6] Simon King. Inventory design for Verbmobil Teilprojekt 4.4. Technical report, IKP, Universität Bonn, October 1996.

[7] Simon King. *Users Manual for Verbmobil Teilprojekt 4.4*. IKP, Universität Bonn, October 1996.

[8] Stefan Merten. Erweiterte Informationen in Sprachsynthesesystemen. Technical Report Verbmobil-Memo 112, DFKI Kaiserslautern, September 1996.

[9] Thomas Portele, Florian Höfer, and Wolfgang Hess. A mixed inventory structure for German concatenative synthesis. In *Progress in Speech Synthesis*. Springer, To be published. Also as Verbmobil report 149.

[10] Paul Taylor. The Rise/Fall/Connection model of intonation. *Speech Communication*, 15, 1994.

[11] Paul Taylor and Alan Black. Synthesizing conversational intonation from a linguistically rich input. In *Proc. ESCA Workshop on Speech Synthesis*, Mohowk, N.Y., 1994. ESCA.

[12] Paul Taylor, Simon King, and Alan Black. CSTR Speech Tools. Available at `http://www.cstr.ed.ac.uk/`, 1996/7. email {`pault,simonk,awb`}`@cstr.ed.ac.uk`.