Proceedings of the Institute of Acoustics

WELSH LETTER-TO-SOUND RULES FOR TEXT-TO-SPEECH SYNTHESIS

Briony Williams

Centre for Speech Technology Research, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN.

1 INTRODUCTION

In developing a text-to-speech (TTS) synthesiser for the Welsh language, it is necessary to include a set of letter-to-sound rules, to convert words from an orthographic representation to a phonemic transcription. Letter-to-sound (LTS) rules have been part of TTS systems for English for many years (e.g. [1], [2]), including those systems that carry out a morphological analysis of the input words. This paper describes a set of LTS rules that have been written for Welsh, together with some examples of those rules.

1.1 The Role of Letter-to-Sound Rules

In a TTS system for unrestricted input text, the most powerful strategy is to carry out morphological analysis on all input words that can be analysed, and to pass the remainder to a set of LTS rules. This approach has the twin advantages of productivity (in that neologisms based on existing morphemes can be analysed) and comprehensiveness (in that even words not found in the morpheme dictionary can, to some extent, be handled). In the case of English, it is accepted that unaided LTS rules will not be completely accurate, but this is held to be preferable to the lack of any output at all. In the same way, LTS rules for Welsh will also be used for those words where no entries exist in the morpheme dictionary.

In addition, LTS rules for Welsh are of use during the development of a TTS system for Welsh. No machine-readable dictionary of Welsh exists that incorporates pronunciation. Therefore it will be necessary to generate phonemic forms for all entries by running LTS rules over the orthographic forms of the headwords in the dictionary (with subsequent manual checking). It is clear that LTS rules have an important part to play in a TTS system for Welsh.

1.2 Particular Features of Welsh

One striking feature of Welsh (at least for English speakers) is the fact that, on the whole, there is a much closer mapping between grapheme and phoneme than is the case in English. This situation could be expected to lead to a higher accuracy for Welsh LTS rules than for English rules, and also a lower number of Welsh rules than English ones.

In the case of consonant graphemes, such is indeed the case, as there is almost a one-to-one mapping between grapheme and phoneme for consonants (even the consonant digraphs are invariant, e.g. 'ff' for /f/). In the case of stress assignment, the situation in Welsh is very much simpler than in English: in Welsh polysyllables, the penult is stressed. Of the few exceptions to this rule, the majority are stressed on the final syllable (ultima). The small number of words stressed on the antepenult are largely loan-words from English. In many cases, irregular stress is indicated orthographically by an accent mark (e.g. *gwacáu*, input at keyboard as 'gwaca/u', 'to empty', /g w a k ai*/, where stress is shown by an asterisk in the phonemic transcription). Where irregular stress is marked orthographically

in this way, it can be handled by the LTS rules. Words with irregular stress that is not orthographically marked will have to be included in the lexicon as exceptions (e.g. *parhau*, 'to continue', /p a r h ai*/).

In the case of vowels, Welsh has phonological vowel length, determined by the type and number of the following consonants [3]. The contrast between long and short vowels exists only in stressed syllables, as all vowels are phonologically short in unstressed syllables. Phonological vowel length is realised phonetically in terms of both vowel quality and duration, in that, except for /a/, phonologically short vowels have lower vowel height [4]. Phonological vowel length must therefore also must be included in the LTS rules.

One vowel in particular (orthographic 'y'), as well as having long and short forms (/ii/ and /i/) like other monophthongs, may also be realised as schwa (/@/) in non-final syllables of polysyllables. It may also represent the first, vocalic, part of a diphthong ('yw', /iu/), or the second, consonantal, part of a diphthong ('wy', /ui/). This means that the LTS rules must do a certain amount of parsing within each word. In addition, Welsh permits *epenthetic* vowels, these being vowels which are pronounced but which are not shown in the orthography. Such vowels occur in certain phonological contexts: for example, *cefn*, 'back', is pronounced /k ee* v e n/ (with stress on the first vowel).

The greatest difficulties, however, are posed by the graphemes 'i' and 'w'. The grapheme 'i' may represent either a vowel (which can be long /ii/ or short /i/) or the palatal glide /j/. The grapheme 'w' may represent either a consonant (/w/), a monophthong (long /uu/ or short /u/), the first half of a diphthong ('wy', /ui/, with vocalic /u/ and consonantal /i/), or a labialisation marker on an alveolar consonant (/l, n, r/) after /g/ or start-of-word (e.g. *gwraig*, 'woman', is pronounced with a labialised /r/ as /g rw ai g/). The grapheme 'w' is by far the most difficult grapheme for the LTS rules to handle, and accounts for a great number of the existing rules, due to the many ways in which it can be interpreted.

1.3 Rule Software Used

The rule software used (known as 'phon') was originally written by Greg Lee of Hawaii University (e-mail address: lee@uhunix.uhcc.hawaii.edu), who used it to realise a set of LTS rules for English. The software allows the linguist to write critically-ordered context-sensitive rewrite rules in a form familiar to linguists. The software converts these rules into a program header file, and compiles a program (written in the C programming language) which runs the rules. This approach combines the advantages of user-friendliness in writing the rules with speed of running the rules. A few modifications were made to the program by the present author, in order to allow for Welsh-specific contexts. For example, 'D' was introduced to refer to 'one and only one consonant grapheme or digraph that can condition phonological length in the preceding vowel', i.e. one of 'b, d, g, f, dd, ff, th, ch'. All the letter-to-sound rules for Welsh were also written by the present author. The rules were developed and run on a Sun 4 workstation, but it is also possible to run them on an IBM-compatible PC.

1.4 Welsh Accent Used

Welsh exists as a constellation of local accents and dialects, together with a non-local attempted spoken standard form known as *Cymraeg Byw*, ('Living Welsh'), and a more formal literary language that is not used in everyday speech. The many local accents precipitate out into two broad groupings: North Welsh accents and South Welsh accents. North Welsh accents contain a greater number of distinct vowel phonemes than do South Welsh accents (in which the high central unrounded vowel has fallen together with the high front unrounded vowel). There are also differences of vocabulary, and

minor differences of syntax. The LTS rules that have been developed for Welsh are based on a South Welsh accent. It would, however, be a minor matter to edit them so that they output North Welsh transcriptions instead (ithe high central unrounded vowel would easily be differentiated from 'i', as they are orthographically distinct).

1.5 Transcription Conventions

A Machine-Readable Phonetic Alphabet for Welsh (MRPAW) was designed, along the lines of the Machine-Readable Phonetic Alphabet for English in use at the Centre for Speech Technology Research (see section 6 of [6]). In this type of transcription system, each phoneme is represented by one or more alphabetic characters, with a space between each phoneme. As a South Welsh accent had been chosen, there were 32 consonants (including the three labialised consonants /lw, nw, rw/, and also the phonemes /z/ and /jh/ that are used only in loan-words from English). There were also 19 vowels, making a total of 51 phonemes in all.

For the input of orthographic forms, it was decided to assume no special character capability on the part of the computer terminal that might be used for the eventual TTS system. Therefore, acute accent was input as a forward slash (/) after the relevant vowel, grave accent as a backward slash (\) after the relevant vowel, and circumflex (^) and diaeresis (") as the appropriate symbols after the relevant vowel. Some examples follow:

(1) *siôl*, 'skull' input as 'sio\l', output by LTS rules as /sh o" l/
(2) *siôl*, 'shawl' input as 'sio^l', output by LTS rules as /sh oo" l/
(3) *sïo*, 'to hiss' input as 'si"o', output by LTS rules as /s ii" o/
(4) *monópoli*, 'monopoly' input as 'mono/poli', output by LTS rules as /m o n o" p o l i/

1.6 LTS Rule Strategy

The LTS rules implemented by Lee for English (using his 'phon' software) were based on those in [7], and only one pass through each input word was required. For Welsh, however, three passes are required through each word, with a different set of rules for each pass. The first pass adds any epenthetic vowels to the input word. This has to be done at the outset so that the phonological syllable count will be faithfully represented in the data received by the second set of rules, which assign stress according to syllable position in the word. The second pass implements rules which locate the stressed syllable, identify the vowels, and differentiate between the vocalic and consonantal forms of orthographic 'w' and 'i'. The third pass implements the grapheme-to-phoneme rules proper, which also determine phonological vowel length and differentiate between the two variant pronunciations of orthographic 'y'.

Section 2 of this paper describes the rules implemented in the first pass. Section 3 describes the rules of the second pass (which form the most complicated set), and section 4 describes the rules of the third pass. Examples will be given for each rule cited.

2 EPENTHETIC VOWELS

In Welsh, an epenthetic vowel occurs after one of /b, d, g, p, t, k, f, v, dh, s, x, th/ when followed by one of /l, n, r/, or between /m/ and /l/, at the end of a word (which may form the first element of a compound word). The vowel to be inserted is identical to that of the preceding syllable. If the preceding syllable contains a diphthong, then the epenthetic vowel takes on the quality of the second half of the diphthong. Some examples follow:

(5) *llyfr*, `book', pronounced /lh i* v i r/
(6) *llwybr*, `footpath', pronounced /lh ui* b i r/
(7) *aml*, `frequent', pronounced /a* m a l/
(8) *copr*, `copper', pronounced /k o* p o r/

The following is a example of the rules for epenthetic vowels, which comprise the rules implemented during the first pass through the input word. Since the rule formalism does not permit the specification of an output symbol from a null input, it is necessary to supply rules for all possible consonants that can precede an epenthetic vowel. Since the quality of the epenthetic vowel depends on that of the preceding vowel, it is also necessary, for each consonant, to specify all possible preceding vowels, both monophthongs and diphthongs. These restrictions lead to the large number of rules required for a relatively simple task, namely 170 rules (including rules for puctuation characters).

(9) e[f]R# = fe

This rule applies to the grapheme `f' when preceded by `e' and followed by one of `l', `n', `r' (represented by the single `R' in the rule), which in turn is followed by a non-alphanumeric character (represented by `#' in the rule), which would signal end-of-word. If the rule applies, the sequence `fe' is output in the place of the input `f'. Thus, for instance, input *cefn*, `back', would be output as *cefen*. It should be noted that, at this stage, the representation of the data is still in orthographic form.

3 SYLLABIC PARSING

The output of the first set of rules forms the input to the second set of rules. These locate the stressed syllable (one per word, except in the case of hyphenated compounds), and locate the vowels. They also differentiate between the various realisations of orthographic 'w' and 'i'. There are 700 rules altogether (including rules for punctuation characters)

3.1 Stress Determination and Vowel Location

The rules fall naturally into several `blocks' of rules, each dealing with the same kind of context. As the rules are critically ordered, the rules with the most specific contexts appear first, while those with the least (or no) context appear last, and serve as default rules. Where the algorithm finds a rule with a context matching the input, that rule will be applied, and no further rules will be applied to the particular target segment. This makes it possible to leave context unspecified in the later rules.

In the first block of rules, vowels that are orthographically marked as stressed are output as capitalised graphemes (capitalisation being used as a marker of stress in the output forms). Such a syllable may form the ultima, the antepenult, or a preantepenultimate syllable of a polysyllabic word, or may be a monosyllabic word. The stressed vowel may be orthographically shown by either an acute accent (input as `/') or a circumflex (if the vowel is long). An example rule follows:

(10) [a/u] = AU. eg. *nesáu* = *nesa/u*, /n e s ai*/, `to approach'

In the above rule, the capitalisation of the output (which is still in orthographic form) indicates stress, while the dot after the output sequence indicates the presence of a vowel. There is no context specification in this rule.

The next block of rules refers to monosyllabic words. The rules assume that all monosyllables are stressed, since stressless monosyllables, forming a limited set of function words, would already be included in the dictionary. The rules allow for vowel contractions, such as word-final "....a'u", pronounced /ai/ (e.g. *dyma'u* from *dyma + eu*, `here are their....'). Included here are rules for penult vowels where the ultima vowel is `w' (phonemically /u/), since such a pattern would appear as a single syllable to the rules.

The third block of rules handles unstressed penults before ultimas that are orthographically marked as stressed, or unstressed preantepenults before stress-marked antepenults. The fourth block of rules, similarly, handles unstressed penults after antepenults that are orthographically marked stressed. The fifth block of rules handles stressed penults (the default case in polysyllables), where no orthographic marking of stress is present. An example rule follows:

(11) [iw]QCVC# = IW. eg. *diwmod*, /d iu" r n o d/, `day'

This rule states that input `iw' is output as `IW.' (ie. a stressed vowel) when it is followed by at least one consonant (represented by `Q'), plus zero or more further consonants (`C'), plus one and only one vowel (`V'), plus a non-alphabetic character (`#'), which may signal either end-of-word or, if it is a hyphen, the end of the first part of a compound word.

The sixth block of rules handles the remaining vowel cases, ie. vowels in unstressed ultimas and antepenults (both of which are normally unstressed), and unstressed penults (normally stressed). The seventh block of rules allows consonant graphemes to pass through unchanged, while a final block does the same for punctuation marks.

3.2 Orthographic 'i'

Orthographic `i' can be realised as either a vowel or a palatal glide. An example follows:

(12) #C[iai]C# = iAI. eg. *iaith*, /j ai" th/, `language'

The rule in (12) states that the sequence `iai' becomes `i AI.' (ie. consonantal `i' followed by the stressed diphthong `ai') when preceded by a non-alphabetic character (`#') -- ie. start-of-word -- plus zero or more consonants (`C'), and when also followed by zero or more consonants (`C') and a non-alphabetic character (`#') -- ie. end of-word. The lack of a dot after the first `i' in the output will indicate to the next set of rules (see section 4 below) that this is not a vowel.

3.3 Orthographic 'w'

Orthographic `w' is the subject of many of the rules in most blocks of rules. Since `w' cannot be classed a priori as either a consonant or a vowel before the operation of the rules, it is necessary to have separate sub-blocks of rules to deal with the special cases it presents. An example follows:

(13) [w]SGHG# = W. eg. *bwled*, /b u" l e d/, `bullet'
(14) [w]CVC# = W. [not actually one of the rules used]

The rule in (13) states that `w' becomes `W.' (ie. identified as a vowel by the following dot, and as stressed by capitalisation) when followed by at least one non-`w' consonant (represented by `S'), plus zero or more non-`w' consonants (`G'), plus one and only one (non-`w') vowel (`H'), plus zero or more non-`w' consonants (`G'), plus a non-alphabetic character (`#'): this is the pattern found in *bwled*. The

simpler rule in (14), while seemingly adequate to cover the same case, would not in fact do so. This is because it is necessary to specify that at least one consonant must intervene, in order to cut out words such as *dwyn*, /d ui* n/, 'to bear', where the 'w' forms part of a diphthong, and words such as *chwarae*, /x w aa* r ai/, 'to play', where the 'w' is a consonant. The rule in (14) is also inadequate because it could not cover cases such as *cwrw*, /k uu* r u/, 'beer', where the second vowel is the grapheme 'w' which has not yet been assigned consonantal or vocalic status (which is why it cannot with certainty be assigned to the 'V').

Complications also arise with the two orthographic sequences 'wyw' and 'ywy', as in *gwyw*, /g w iu*/, 'withered', and *tywyllwch*, /t @ w @* lh u x/, 'darkness' (where /@/ is the vowel schwa and /lh/ is a voiceless alveolar lateral fricative). The following are two of the rules for these sequences:

(15) K[wyw]C# = wYW.
(16) [ywy]SGwG# = y.wY.

The rule in (15) states that the sequence 'wyw' becomes 'wYW.' (ie. consonantal 'w' followed by the stressed diphthong 'yw') when preceded by one of 'c, g, ngh' ('K') and followed by zero or more consonants plus a non-alphabetic character ('#'). This rule would apply to *gwyw* to produce (part of) 'gwYW.' The rule in (16) states that the sequence 'ywy' becomes 'y.wY.' (ie. unstressed vowel 'y' followed by consonant 'w' and stressed vowel 'y') when followed by one and only one non-'w' consonant ('S'), plus zero or more non-'w' consonants ('G'), plus 'w', plus zero or more non-'w' consonants ('G'), plus a non-alphabetic character ('#'). This rule would apply to *tywyllwch* to produce (part of) 'ty.wY.llw.ch'.

(17) G[wyw]H = WY.w eg. *dwywaith*, /d ui* w ai th/, 'twice'

The rule in (17) states that the sequence 'wyw' becomes 'WY.w' (ie. a stressed realisation of the diphthong 'wy' plus consonantal 'w') when preceded by zero or more non-'w' consonants ('G') and followed by one and only one (non-'w') vowel. This rule is ordered after that in (15), and so does not deal with cases with initial 'c', 'g' or 'ngh'.

It is clear that, at least in the case of the grapheme 'w', what the rules are doing is a simple parse of the input segment string into syllable-sized units, so that the consonantal or vocalic nature of 'w' can be determined. Context-sensitive rewrite rules can handle this type of process only somewhat clumsily, as they are primarily suited to rewriting (using distinct input and output alphabets, in most cases), rather than to the process of grouping or parsing (where the nature of the units of representation does not change). However, this slight loss in expressive power is more than compensated for by the speed of execution of such rules. An alternative strategy might have been to undertake syllable parsing using two-level rules and a lexicon consisting of permissible syllable templates. This is the type of approach adopted in several algorithms for morphological decomposition, such as [8], where the lexicon consistes of permissible morphemes and the two-level rules encode 'graphotactic' spelling changes at morpheme boundaries (eg. for English, the occurrence of 'ie' for underlying 'y' before the inflectional suffix 's' in *tries*). However, such a strategy would have placed a greater computational load on the overall system (in terms of memory size and processing speed) for the sake of only a small benefit (as only some of the rules carry out a parsing type of function). Therefore it was decided to retain the same rule type throughout the letter-to-sound rules.

4 GRAPHEME-TO-PHONEME CONVERSION

The output of the second set of rules forms the input to the third (and final) set of rules. These rules number 225 (including the punctuation rules) and are less complex than the second set, though more complex than the first. The rules for consonants are very straightforward, while those for vowels must handle vowel length and the variant realisations of 'i', 'y' and 'w'. The output of the second set of rules distinguishes vowels by using a following dot, and distinguishes stressed vowels by using capitalisation.

4.1 Phonological vowel length

In Welsh, phonological vowel length for monophthongs is determined by the type and number of the following consonants [3]. Where a vowel is followed by one of orthographic 'b, d, g, f, dd, ff, th, ch' (/b, d, g, v, dh, f, th, x/), or no consonant at all, before the next vowel (or end-of-word) then the vowel is long if it is stressed (assuming a South Welsh accent). Where the vowel is followed by any other consonant, or by two or more of any consonant, then it is short (even if stressed). The following are examples of vowel length rules:

(18)	C[E].Dwy. = ee*	eg. *dedwydd*, /d ee* d ui dh/, 'happy'
(19)	C[E].DwV = e*	eg. *edwi*, /e* d w i/, 'to fade'
(20)	C[E].Dw = ee*	eg. *bedw*, /b ee* d u/, 'birch', *credwr*, /k r ee* d u r/, 'believer'
(21)	C[E].DiV = e*	eg. *brechiad*, /b r e* x j a d/, 'vaccination'
(22)	C[E].DV = ee*	eg. *brechu*, /b r ee* x i/, 'to vaccinate'
(23)	C[E].V = ee*	eg. *lleol*, /lh ee* o l/, 'local'
(24)	C[E].D# = ee*	eg. *llech*, /lh ee* x/, 'slate'
(25)	C[E].P# = ee*	eg. *lles*, /lh ee* s/, 'benefit'
(26)	C[E].# = ee*	eg. *lle*, /lh ee*/, 'place'
(27)	[E].C = e*	eg. *llen*, /lh e* n/, 'sheet'
(28)	[e]. = e	eg. *llefaru*, /lh e v a* r i/, 'to speak'

These (critically-ordered) rules use 'D' to refer to the vowel lengthening consonants given above, and 'P' to refer to the additional consonants 's' and 'll' (/s/ and /lh/) which lengthen a preceding monophthong only when word-final, in South Welsh accents [3]. In (18), (20), (22), (23), (24) and (25), the single vowel-lengthening consonant is followed by a vowel or end-of-word, and so the preceding vowel (which is stressed) is long. In (19), (21) and (27), the vowel-lengthening consonant is followed by a second consonant (which may be a glide) and so the preceding vowel (even though stressed) is short. In (26), the stressed vowel is followed by end-of-word, and so is long, while in (28), the input form of the vowel shows it to be unstressed, and so it is short whatever the context.

4.2 Orthographic 'i'

The second set of rules has distinguished between the vocalic and consonantal realisations of the grapheme 'i' by adding a dot after the former. The third set of rules then converts these to the appropriate phonemes: /i/ or /ii/ (stressed or unstressed) for the vocalic form, and /j/ for the consonantal form. An exception occurs where 's' precedes, in which case the 'si' digraph takes precedence and becomes /sh/.

4.3 Orthographic 'y'

Where it is a monophthong, orthographic 'y' becomes schwa (stressed or unstressed) when in non-final syllables of polysyllables, and /i/ or /ii/ (stressed or unstressed) in monosyllables or final syllables of

polysyllables. Where it forms part of a diphthong (orthographic `oy, wy, yw, ey'), orthographic `y' is a high front vocoid and never schwa.

4.4 Orthographic 'w'

A final complication in the treatment of `w' is the case of labialised consonants, in words such as *gwlad*, /g lw aa* d/, `country'. In such cases, there is strong labialisation on the consonant following the /g/ (which may be /l/, /n/ or /r/). A labialised consonant may also follow word-initial /ng/ (voiced velar nasal) or be word-initial itself. The latter two contexts are the result of two different types of consonantal mutation (nasal mutation and soft mutation) applying to the root form with initial /g/. One of the rules dealing with labialised consonants is as follows:

(29) #g[wr] = rw eg. *gwraig*, /g rw ai* g/, `woman'

Since the second set of rules output a `w' with no following dot, it is clear from the input form to these rules that the `w' in question is not a vowel. A contrasting case is the following:

(30) [w] = u eg. *damhegwr*, /d a m h ee* g u r/, `allegorist'

The rule in (30) is the final one for input `w', and so needs no context specification, not even a following dot. It contrasts with the case in (29) in that, in (30), the input `w' functions as a vowel even though it also is flanked by the graphemes `g' and `r'.

5 ACKNOWLEDGEMENTS

The author gratefully acknowledges the support of a BP Research Fellowship, awarded by the Royal Society of Edinburgh and funded by British Petroleum.

6 REFERENCES

[1] J ALLEN, M S HUNNICUTT & D KLATT, *From text to speech: The MITalk system*, 1987. Cambridge: CUP.

[2] J M McALLISTER, `CSTR Text-to-Phoneme Project Status Report', October 1988. CSTR internal Report.

[3] G M AWBERY, `Phonotactic Constraints in Welsh', in [5].

[4] G E JONES, `The Distinctive Vowels and Consonants of Welsh', in [5].

[5] M J BALL & G E JONES, *Welsh Phonology*, (1984). Cardiff: University of Wales Press.

[6] J HARRINGTON, J LAVER & D CUTTING, `Word-structure reduction rules in automatic, continuous speech recognition. In: *Proceedings of the Institute of Acoustics*, vol. 8, part 7 (1986):451-459.

[7] H S ELOVITZ, R W JOHNSON, S McHUGH & J E SHORE (1976), `Automatic translation of English text to phonetics by means of letter-to-sound rules'. US Naval Research Laboratory Report 7948.

[8] G D RITCHIE, S G PULMAN, A W BLACK & G J RUSSELL (1987) `A Computational Framework for Lexical Description', *Computational Linguistics*, vol. 13: 290-307.