Documentation and User Guide to Unisyn Lexicon and Post-Lexical Rules

Susan Fitt
Centre for Speech Technology Research
University of Edinburgh

Contents

1. Notes

1.1. Files

Lexicon

unilex	Contains accent-independent transcriptions

Core scripts (perl)

get-exceptions.pl	Merges exceptions with lexicon
read-text.pl	Basic text processor, using output of get-exceptions to convert text to pronunciation string, for input to post-lexical rules
post-lex-rules.pl	Post-lexical rules and conversions
map-unique.pl	Removes redundancy in keysymbol output

Output utilities (perl)

transform-text.pl	Calls other scripts to transform plain text to accent-specific, non-redundant transcriptions
make-lex.pl	Calls other scripts to make an accent-specific, non-redundant lexicon (treats all entries as single-word phrases)
output-fest.pl	Simplifies keysymbol output (pronunciation string only, not lexicon) for input to Festival synthesiser
output-sam.pl	Transforms accent-specific keysymbol output (pronunciation strings or lexicon) to SAMPA
output-ipa.pl	Transforms SAMPA output (pronunciation strings or lexicon) to IPA in html-format unicode

Modules (perl)

Readtext.pm	Reads lexicon ready for making strings, and converts text to pronunciation

Transcription checking utilities (perl)

check-trans.pl	Checks format of keysymbol lexica
check-phon.pl	Checks phonotactics of keysymbol lexica or pronunciation strings

Supporting files

uni_towns	Contains town hierarchies used by transformation scripts
uni_exceptions	Lists regional pronunciation exceptions
uni_positions	Lists keysymbols and some of their features, used by various scripts
uni_scores	Contains scores which determine post-lexical rules and symbol mappings
uni_rules	Contains phonotactic rules used by checking programs
uni_sam	Contains scores specifying mappings from accent-specific keysymbols to SAMPA
uni_freqs	Lists word frequencies

Perl scripts are written for version 5.6.0, and can be run on UNIX or MS-DOS.

1.2. Annotation conventions

NORTH	Keywords are given in small capitals
BAN**A**NA	Bold is used to highlight word segments
|@r|	Keysymbol transcriptions in ASCII use straight brackets
/ɾ/	Phonemic transcriptions in IPA use forward slashes
[aɪ]	Phonetic transcriptions in IPA use square brackets
unilex	Existing lexicons, scripts and supporting files are in bold italics
towncode	In examples, variables, arguments etc. defined or selected by the user may be highlighted in green.

Keysymbol, phonemic and phonetic brackets are only used where they are needed for clarity. Note that square bracket pairs and single forward slashes, used for IPA phones and phonemes, can also form part of keysymbol transcriptions.

2. Overview

This documentation provides a user-guide for the Unisyn accent-independent keyword lexicon, which has been produced to facilitate the synthesis of regional accents of English. The keyword lexicon is used to compose transcriptions of running text, which are then put through various rules to generate accent-specific transcriptions.

2.1. The Accent-Independent Lexicon

The accent-independent keyword lexicon, ***unilex***, is based on Wells (1982), who classifies the vowels of English into keyvowels, which describe some of the phonemic variation in regional Englishes. For example, in some accents the vowels of FOOT and STRUT differ, while in other accents they are the same. These two words are "keywords". So, when we wish to describe the vowels of other words such as 'put' or 'putt', rather than describing them in phonemic or phonetic terms we can say that they belong to the set of words defined by the keywords FOOT or STRUT; i.e. a word in the FOOT set will have the same vowel as the word 'foot', while a vowel in the STRUT set will have the same vowel as 'strut'. 'Put', then, belongs to the FOOT set, while 'putt' belongs to the STRUT set.

This keyword system is the basis of the lexicon. We use "keysymbols" based on keywords to transcribe the lexicon; unlike phonemes or phones, which describe only one accent, keysymbols map many accents at once and so enable a single lexicon to encode the characteristics of a number of accents. A keysymbol can be seen as a kind of meta-phoneme. Well's keyword set is not extensive enough for our purposes, as by his own admission the keywords do not cover all the vowel variation of English; for example, his NURSE keyword encompasses a number of distinct vowels in Scottish English, so to differentiate these and other vowels in the dictionary we have added more keywords. Furthermore, Wells's keywords only cover vowels. While the vast majority of English accent differences occur in the vowels, there are some consonantal differences that a comprehensive lexicon needs to include.

We have aimed to include in the lexicon all the features necessary for describing regional pronunciation at the segmental level. (Some regional features, such as intonation or duration, are beyond the scope of a lexicon.) So, in addition to the higher-level segmental information (such as the FOOT-STRUT distinction) which is normally covered in pronunciation lexica, we have included morphological and boundary information, which is necessary to derive predictable lower-level features such as allophones. Our transcription system also includes systematic typographical markers which extend the basic keysymbol set, and allows us to cover many pronunciation variants which would otherwise have to be listed as exceptions, such as 'tomato' with /ɑː/ in the UK and /e/ in the US.

The transcription system and other features of the lexicon will be described in more detail in Section 3.

2.2. Accent-Specific Rules

To produce pronunciations in any given accent, the words in the base lexicon undergo accent-specific rules. Firstly, the accent-independent lexicon ***unilex*** is checked against a small exceptions list, ***uni_exceptions***, using ***get-exceptions.pl***. The resulting lexicon is then used to compose transcriptions of running text, using ***read-text.pl*** or some other text processing script. The resulting transcription is run through ***post-lex-rules.pl*** to produce an accent-specific transcription which includes allophones and other rule-based features. This transcription still uses keysymbols rather than accent-specific segments; it can be run through a separate mapping (***map-unique.pl***) to reduce the symbol set and retain only distinctive symbols for that accent. If desired, ***output-sam.pl*** or ***output-ipa.pl*** can then be used to display the output as SAMPA or IPA respectively.

The scripts and rules are described fully in Section 4, and a summary of how to achieve various tasks using the scripts is given in Section 6.

2.3. Documentation

The rest of this document describes the system in more detail. Section 3 describes the lexicon. Section 4 contains a description of the processing used to derive pronunciations in each accent, including examples of the post-lexical rules. In Section 5 there are some brief notes on the supported accents. Finally, Section 6 gives a summary of how to use the lexicon and rules.

References, parts-of-speech and keysymbol tables are listed in the appendices.

3. Description of Lexicon

This section describes the structure and content of the lexicon ***unilex***.

The lexicon uses a one-line format. A colon is used as a field divider to enable simple searching on particular fields; the field divider is used even when the field is empty, so that other fields may be easily located. Examples are:

```
economically:1:RB: { ~ ii . k @ . n * o . m == i k }.> l >> iy > :{econom==ic}>al>>ly>:1194
economically:2:RB: { ~ e . k @ . n * o . m == i k }.> l >> iy > :{econom==ic}>al>>ly>:1194
rocking-horse::NN: { r * o k }.> i ng >.{ h ~ or r s } :{rock}>ing}>-{horse}:6
```

The fields are: orthography; identifier (optional); part-of-speech; pronunciation; enriched orthography; and frequency. These are discussed below.

Each different orthography has a different entry, so 'synthesise' and 'synthesize' are separate entries. Homographs are given different entries if the transcription differs. This may be a pronunciation difference due to category or semantics (e.g. 'record' as a noun and as a verb), or personal choice (included for a small number of entries such as 'economically'), or it may be a morphological difference.

3.1. Orthography

Orthography is given in lower-case for all words. Only single words are listed, not phrases, though the words may be compounds such as 'hotrod' or 'rocking-horse'. As well as the letters [a-z], this field may contain hyphens, apostrophes or full stops (e.g. 'dot.com').

3.2. Identifier

This field can be used to hold semantic information, distinguishing for example between the homographs 'row' (a boat) and 'row' (to argue). It can also be used to mark free variants, such as the variants of 'economically' shown above, which are not defined by accent or style.

All orthographies which have more than one entry have been numbered, with the entry believed to be the most frequent as number 1. So, for example, 'does' (verb) is listed as number 1, so that in the absence of any other information it will be chosen in preference to 'does' (plural noun) which is number 2.

3.3. Part-of-Speech

The parts of speech used are the Penn Treebank categories, obtained partly by automatic parsing and partly by hand-editing. These are included as potentially useful information, although at present they are only used in a few cases, for example Read-text.pm destresses 'have/had/has' before a past-tense verb, but leaves it stressed elsewhere, as in the RP phrases:

```
i have gone:     #{ ai }#.#{ h @ v }#.#{ g * o n }#
i have a dream:  #{ ai }#.#{ h * a v }#.#{ @ }#.#{ d r * ii m }#
```

These modifications are simplistic, but aid naturalness.

The tags are taken from the 1993 version (Marcus et al. 1993), and are shown in Appendix II. Different categories are separated by forward slashes, for example

```
horses::NNS/VBZ: { h * or r s }.> I7 z > :{horse}>s>:20865
```

Linked categories have a vertical line, for example the plural possessive

horses'::NNS|POS: { h * or r s }.> I7 z > :{horse}>es'>:

3.4. Pronunciation

The pronunciation field is the core of the lexicon. In general, unpredictable information is included in the lexicon, while predictable information is generated by rule. So, pre-consonantal /r/, for instance, uses the same keysymbol in the base lexicon as word-initial /r/, and is deleted by rule for non-rhotic accents. However, some predictable information is included in the lexicon to improve the patterning of keysymbols and aid rule-writing. For example, |aer| in FIRE is a predictable variant of |ae| (TIED) before |r|, and could be derived by rule, but to improve consistency of transcriptions and rules for the pre-r vowels, this symbol is transcribed in the lexicon.

The keysymbols can be grouped in terms of their linguistic function (consonants, vowels etc.) and their place in the lexicon/rule system (pre-rule symbols, global symbols etc.).

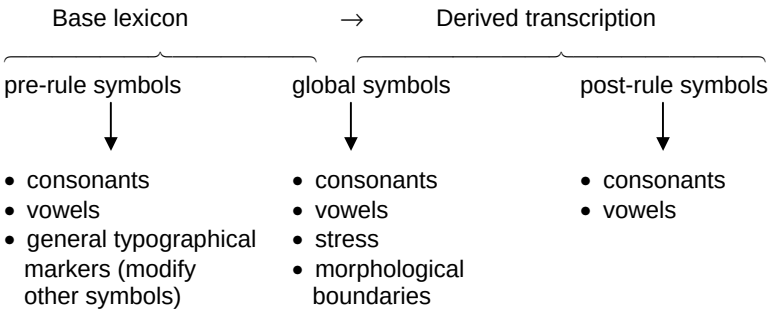

Figure 1: Symbol types

The base lexicon contains pre-rule symbols and global symbols; the output of post-lexical processing contains global symbols and post-rule symbols.

A complete list of keysymbols is given in Appendix III. Details of the symbol types are discussed below, with full notes on the most complex group, the global and post-rule vowel symbols. For details of the rules which effect the changes between symbols, see Section 4.4.

3.4.1. Pre-rule symbols

Pre-rule symbols appear only in the base lexicon, and extend the global symbol set by using typographical features (numbers, capitals, square brackets, forward slashes and round brackets). They are converted by the post-lexical rules to global symbols, which may then undergo further rules to become post-rule symbols or different global symbols.

(1) Pre-rule vowels and consonants

Pre-rule vowel symbols are subdivisions of other vowel symbols; for instance, |ah2| ('d**a**nce') is a subdivision of |ah|, and maps to either |ah| (BATH) or |a| (TRAP) depending on the accent. |I2| is a type of |I| (reducible |i|) and converts to |i| or |@| depending on the accent.

The symbol |ii2| ('mill**i**on') becomes either a vowel (|ii|) or a consonant (|y|).

Appendix III shows these and other pre-rule symbols along with the global symbols they are mapped onto.

(2) General typographical markers

The pre-rule typographical markers are listed with examples in Appendix III, and consist of brackets, numbers, forward slashes and capitalisation. They can be used in conjunction with any appropriate vowel or consonant keysymbol, or even with stress and boundary markers, and represent general patterns across accents which are not covered by the global keysymbol set. For example, square brackets represent deletions in certain accent groups. So, |[h]| represents an |h| which is retained in the UK and deleted in the US (e.g. 'herb') while |[h1]| represents an |h| deleted in the UK and retained in the US. These conventions provide a number of advantages: they extend the available symbol set; they aid transparency of transcriptions; and they reduce the exceptions list. For example, by transcribing 'tomato' in the base lexicon as

tomato::NN: { t @ . m * aa/ee . t ou } :{tomato}:2723

we avoid the need to make it an exception. This speeds up the implementation of the exceptions list, and also keeps as much information as possible in the base lexicon. So, where these typographical conventions can be used, this solution is preferable to an entry in the exceptions list.

3.4.2. Global and post-rule symbols

All accent-dependent transcriptions output by the rules should contain only global and post-rule symbols. Global symbols can occur in both the base lexicon and the output lexicon. Post-rule symbols are not found in the base lexicon, and are produced by the post-lexical rules. The post-rule symbols may be applied differently in different accents, for example the environment which conditions dark 'l', |lw|, varies according to accent. In some cases, a single post-rule symbol has different derivations in different accents, for example the tap, |t^|, is an allophone of |t| and |d| in US accents, but an allophone of |r| in Scottish accents.

Here we briefly discuss consonants, and then look at vowels in more detail.

(1) Global and post-rule consonants

These are listed in Appendix III. As is often noted, the regional accents of English have little systemic variation in their consonants. In the global keysymbol group, the only differences are in the use of |x| (LO**CH**), |ll| (**LL**ANDUDNO) and |hw| (**WH**ICH); the first two are minor both in geographical spread and incidence in the lexicon. |hw| is more widespread, though in many areas where it is used it is dying out.

There is more variation in the post-rule consonant symbols, shown in red in Appendix III; these represent allophones. For example, the use of taps and glottal stops varies by accent.

There is also some regional variation in the rules to which global consonants are subjected, for instance |s y| → |sh| in 'issue', or |h| → |0| in 'hat'.

(2) Global and post-rule vowels

The vowels contain most of the accent differences in English. As with consonants, the symbol set is not designed to describe phonetic differences (though these are illustrated in the table), but systemic differences. The global vowel keysymbols are based on Wells's keywords, extended and adjusted where necessary to cover the accents implemented so far.

Like the consonants, some vowel symbols have different allophones, or are subjected to different rules, in different accents. Due to the importance of vowels, the individual global vowel keywords and the post-rule vowels are briefly discussed here. They are ordered as in the symbol tables in Appendix III.

DR**E**SS

This keyword is straightforward and undergoes no rules or splits, in the accents studied so far,

M**A**ZDA

This is motivated by the correspondence between TRAP and PALM in UK and US accents; it is used mainly in foreign words. For more on this, see Lindsey (1990).

TR**A**P

This keyword is fairly straightforward, and the resulting vowel is a front vowel for every accent that contains more than one non-rhotic 'a' vowel. The keysymbol is used as input for the raised-a rule in US accents, with the environment in which the rule applies varying by accent.

B**A**TH

This keyword is used for the different pronunciations of 'bath' in northern and southern UK accents. Like |a|, it is an input for raised-a in US accents. This set has a split, denoted by the pre-rule keysymbol ah2, which is used for example in Australia.

BAN**A**NA

This symbol is in effect the reverse of the MAZDA symbol, with the PALM and TRAP vowels used in the UK and US respectively. It is used as input for raised-a. Following Wells (1982, Vol. 2, p. 356) 'half' and 'can't' have been transcribed with the BANANA vowel.

P**A**LM

This vowel is a long, back vowel everywhere that a distinction is made between non-rhotic 'a' vowels.

ST**A**RT

This is a rhotic 'a' vowel. In non-rhotic areas it merges with the PALM vowel, but in rhotic areas it may have a different vowel quality. For example, in Scotland it is often further back than the PALM vowel. It could be produced as an allophone of PALM for accents which need this distinction. It has been retained as a separate keysymbol for symmetry with other pre-rhotic vowels. It also enables us to use the typographical convention VR → vr/@r in words such as

mallard::NN: { m * a . l AR r d } :{mallard}:224

ann

The post-rule symbol used in ann is the result of the raised-a rule in US accents.

g**oa**l

This post-rule symbol is used for the pre-/l/ allophone of GOAT.

G**OA**T

The quality of this vowel varies by accent, and is sometimes a diphthong and sometimes a monophthong. Some accents have an allophone before |l|; some of these accents distinguish between 'holy' and 'wholly', and some do not.

KN**OW**

This is the same as the GOAT vowel in most accents; like WASTE/WAIST, the two vowels are distinguished in some accents such as Welsh English.

ADI**O**S

This keyword is used for the correspondence between LOT and GOAT in UK and US accents, and is used mainly for foreign words and names. (See Lindsey (1990)).

L**O**T

This set of words is the same as PALM in US accents, but is distinct in UK accents.

CL**O**TH

This set of words merges with LOT in most UK accents, but with THOUGHT in US accents.

TH**OU**GHT

This is an open back vowel everywhere.

N**O**RTH

This is a pre-rhotic vowel. In non-rhotic accents it merges with THOUGHT.

F**O**RCE

This vowel also merges with THOUGHT in non-rhotic accents. FORCE is distinguished from NORTH in some (mostly rhotic) accents, such as Scotland, but others are losing this distinction and merging the two sets.

FL**EE**CE

This is a high front vowel; in some areas it is diphthongised. It is one of the vowels lengthened by the Scottish Vowel Length Rule. We have used a rule to reduce FLEECE to a short vowel of the same quality, either HAPPY or harriet, in certain unstressed environments.

HAPP**Y**

This vowel is mainly found morpheme-finally, and only occurs in unstressed positions. In some accents it is similar to FLEECE, and in others it is similar to KIT. In RP it is traditionally similar to KIT, but is changing to a higher vowel. It is not shown in the table as merging with KIT or FLEECE; since it occurs in different environments it is difficult to be sure whether it is phonetically the same vowel or not. Before a vowel within

the same word, and generally elsewhere, the HAPPY vowel is pronounced as the FLEECE rather than KIT, regardless of accent.

KIT

This is a high front short vowel. In New Zealand it is not distinct from COMMA.

harriet

This post-rule vowel is used for a short fleece vowel, in accents where the HAPPY vowel is the same as KIT.

agreed

This post-rule vowel is the result of the Scottish Vowel Length Rule applied to the FLEECE vowel.

LETTER

This vowel merges with COMMA in non-rhotic accents, but may be distinctly r-coloured in rhotic accents. Like START, it could be produced as an allophone of comma, but is retained as a global symbol to simplify rule-writing.

COMMA

This vowel is generally a schwa.

STRUT

This vowel is distinct in most accents, but in some it merges with other vowels; for example, in Northern England it merges with FOOT, and in Wales it merges with COMMA.

FOOT

This is a short back high vowel. In Scotland it is not distinct from GOOSE – this is one of the most robust vowel markers of a Scottish speaker.

GOOSE

This is a high back vowel. Like FLEECE, it is diphthongised in some areas. A rule reduces the length of this vowel in certain environments to the LOUISE vowel.

BLEW

In most areas this is not a distinct vowel, but is the same as GOOSE. However, in some regions it is a separate diphthong. It has a subdivision, iu3, which merges with GOOSE in some |iu| regions, but is retained as the BLEW vowel in others.

brewed

This post-rule vowel is the result of the Scottish Vowel Length Rule applied to GOOSE, FOOT or BLEW.

louise

This post-rule vowel is the result of the GOOSE shortening rule.

ghoul

This vowel is an allophone of GOOSE before /l/, for accents in which the quality of the two is noticeably different.

WAIST

This vowel began as Wells's FACE keyword, but was split into WAIST and WASTE to reflect a distinction in some conservative accents. In some of these areas this distinction is dying out.

WASTE

See WAIST.

PRICE

This is a diphthong of varying quality.

TIED

The keyword TIED splits from the keyword PRICE in Scottish accents. It is of marginal phonemic status, and generally follows the same environments as are used for the Scottish Vowel Length Rule, though in some cases the distinction has sociolinguistic connotations.

FIRE

This is a pre-rhotic vowel. In non-rhotic accents this vowel usually has a schwa off-glide, and in many accent it is distinct from the TIED vowel (c.f. 'high-risk', with the TIED vowel, and 'hiring', with the FIRE vowel). Like start, use of fire as a separate symbol from tied enables us to use the typographical convention VR, for example

ayrshire::NNP: { * eir r =.= sh AER1 r } :{ayr==shire}:100

t**i**me

This is an allophone used in Southern US, a monophthongal variant of PRICE (or TIED) which occurs in environments other than before a voiceless consonant in the same syllable.

CH**OI**CE

This is a diphthong everywhere.

C**OI**R

This is a pre-rhotic vowel corresponding to CHOICE. It is very rare, and in many accents is equivalent to a sequence such as |oi| + |@r| or |@|.

M**OU**TH

This is a back diphthong. Some words in the MOUTH set are pronounced /ʉ/ in broad Scottish accents; this split has not at present been incorporated.

H**OU**R

This set consists of pre-rhotic MOUTH vowels.

t**i**me

Like l**ou**d, this is an allophone used in Southern US, a monophthongal variant of HOUR.

ID**EA**

This set of words has the same diphthong as NEARING in RP, and a sequence of /i/ + /ə/ in General American and some other accents. Unlike NEARING, it is not a pre-rhotic vowel. For |i@| before syllable-final |l|, the contrast with |ii| + |l| is debatable (c.f. 'real', transcribed by most RP dictionaries as IPA /rɪəl/, the equivalent of keysymbols |r * i@ l|, and 'reel', generally transcribed as /ril/, the equivalent of |r * ii l |).

N**EA**RING

NEARING has been chosen as a keyword rather than Wells's NEAR as it is only in the _rV environment that the |ir| keysymbol is retained for all accents in our system. In Northern UK accents, for instance, final |ir| in 'near' is transformed into the sequence |ii @r|.

n**ear**

This post-rule vowel is used for the output of the Scottish Vowel Length Rule applied to the vowel in NEARING.

b**ear**d

This is used for the Australian NEARING vowel before consonants.

N**UR**SE

This is a long central vowel. The original NURSE set in Wells has been split into NURSE and PERT; a further split is possible, of words which contain /ɾ/ in Scottish accents. This split is less common and has not been made at present.

P**E**RT

The PERT set has been split from Wells's original NURSE set, and consists of words which have /eˠ/ in Scottish accents.

SQU**A**RING

This vowel is generally a diphthong in non-rhotic accents and a monophthong in rhotic accents. For New Zealand English we have merged it with NEARING (hence the choice of SQUARING rather than SQUARE as the keyword).

J**U**RY

Like NEARING, Wells's CURE keyword has been replace by JURY as |ur| in 'cure' is often replaced by allophonic variants. In some accents and some environments, |ur| is merging with |or| and/or |oo|.

c**ur**e

This is for the output of the Scottish Vowel Length Rule applied to JURY.

curious

This vowel represents the Welsh sound [ɪu] before /r/. Phonetically it is the same as the BLEW vowel, but a pre-rhotic variant is required for rule specifications.

3.4.3. Stress

The conventions followed for stress are:

i. All lexical entries are given one primary stress. There are a handful of exceptions which have no primary stress, for example 'sh', which is transcribed |sh|, or 'hmm', which is transcribed |h m|. The primary stress is shown immediately before the vowel, and is transcribed for the word in isolation; stress location may of course change due to emphasis or context, often shifting to a previous secondary stress for adjectives preceding nouns:

```
absentee::JJ: { ~ a b . s == n t }.> * ii > :{abs==ent}>ee>:132
landlord::NN: { l * a n d }.{ l - or r d } :{land}{lord}:4567
    → absentee landlord: #{ ~ a b . s == n t }.> - ii >#.#{ l * a n d }.{ l - or r d }#
```

Some stress shift is included in Readtext.pm.

ii. All schwas are unstressed. There is just one exception to this, 'huh', transcribed |h * @|.

iii. Compounds with more than two syllables generally contain one primary and at least one secondary stress, e.g.

```
absent-minded::JJ: { ~ a b . s == n t }.{ m * ai n d }.> I7 d > ::
```

iv. Lexical roots in compounds are given at least tertiary stress; this does not normally note a stressed pronunciation, but is used to block certain rules, for example vowel reduction, glottalisation or tapping, e.g.

```
armpit::NN: { * ar r m }.{ p - i t } ::
```

Tertiary stress is not generally used on function words such as 'and', but includes verbal particles, for example, 'lived-in'.

3.4.4. Boundary markers

An important feature of the lexicon is the boundary markers, which show syllable divisions and morphological breakdown.

(1) Syllables

The most basic boundary marker is the syllable marker. In general, syllable boundaries are aligned with morphemes, rather than by general phonetic principles, since these can be applied by rule if desired. In a few instances, however, phonetic considerations overrule the morphological principles, as will be shown below. Syllable divisions follow these rules:

i. Syllable boundaries are aligned with free roots and affixes, for example:

```
amalgamated::VBN/VBD/JJ: { @ . m * a l . g @ m }.> ~ ee t >.> I7 d > :{amalgam}>ate>>ed>:786
```

This facilitates matching of transcriptions across different words.

ii. The above is overruled if it results in inadmissible syllable structures, and is also subject to phonetic considerations in consonant clusters, e.g.

```
puppetry::NN: { p * uh . p I2 . t }> r iy > :{puppet}>ry>:2
```

Note that 'bedroom' syllabifies as 'be-droom' rather than the expected 'bed-room' (c.f. 'outrage', which syllabifies as 'out-rage'). '-ion' endings are syllabified as |sh }.> n| rather than |}.> sh n >| as they often violate phonotactic constraints, e.g.

```
election::NN: { i . l * e k . sh $}> n > :{elect}>ion>:15705
```

There is some accent variation in syllable boundaries, for example 'petrol' is generally syllabified as 'pe-trol', but some Scottish speakers (in an as yet unimplemented variety) say 'pet-rol'. Syllable boundaries affect some rules, and are sometimes transcribed differently for different varieties for this reason, for example 'jesuit':

jesuit::NNP: { jh * e (z ./. z) y iu3 . I t } :{jesuit}:208

The string |. z y|, used for US English, is subject to a rule which converts it to |. zh|, while the syllable boundary in |z . y|, used for UK English, blocks the rule.

iii. Syllable boundaries are not affected by bound roots, e.g.

ambulate::VB/VBP: { * a m . b y UU . l == ee t } :{ambul==ate}:2

iv. When no other considerations apply, maximal onset is used.

v. If there is a conflict between these rules when applied to different accents, the simplest transcription is chosen, e.g.

osmosis::NN: { o z/s . m * ou . s == i s } :{osmos==is}:311

rather than

osmosis::NN: { o (z ./. s) m * ou . s == i s } :{osmos==is}:311

(2) Morphological boundaries

There are a number of levels of markers. Curly brackets {} surround free morphemes, left angle brackets << are used to enclose prefixes, right angle brackets >> are used for suffixes, and equals signs == join bound morphemes or relatively unproductive affixes, for example 'ade' in 'blockade'. These symbols are used both in the pronunciation field and on the enriched orthography. The dollar sign $ notes a derived pronunciation which varies from the root in any way other than stress location, and is not shown in the enriched orthography field.

In our previous examples,

economically:1:RB: { ~ ii . k @ . n * o . m == i k }.> l >> iy > :{econom==ic}>al>>ly>:1194
rocking-horse::NN: { r * o k }.> i ng >.{ h ~ or r s } :{rock}>ing}>-{horse}:6

some of these are demonstrated. 'Economic' is a free morpheme surrounded by {}. There is an internal boundary == (c.f. 'economist' etc.), and 'economically' consists of the morpheme 'economic' with the addition of suffixes. The compound 'rocking-horse' contains two free morphemes; we can also see where the internal word-boundary is, as this occurs at the junction of the opposing brackets >.{. Other possible compound brackets, with, or sometimes without, syllable boundaries, are }.{, >.{ and >.<. We can tell from this information that the word breaks down into 'rocking+horse' rather than 'rock' + 'ing-horse'.

Note that in our system there is no combination of bound morpheme + free root/suffix/affix, only bound morpheme + bound morpheme. Thus, the same element may sometimes be, say, a prefix, and sometimes a bound morpheme, for example <anti<{climax} and {anti==cline}. This solution is not ideal, as we cannot directly relate the two 'anti' elements, but it has been chosen as a simple system that enables easy identification of free units; in these two examples, {climax} and {anticline} are the minimal free units in the two words. Free unit boundaries are far more important than bound unit boundaries in determining pronunciation, such as the Scottish Vowel Length Rule, or the difference in Belfast between 'matter' (with a dental 't') and 'fatter' (see Kaisse and Hargus 1994).

These markers serve a number of functions, which can be divided into lexicon development and pronunciation. Equals signs == and dollar signs $ are used primarily in lexicon development, to aid consistency of transcription across different words, and to help identify roots.

At the level of pronunciation, morpheme boundaries can trigger or block post-lexical rules. For example, one of the environmental conditions for the Scottish vowel length rule is a free morpheme boundary, c.f.:

greed::NN: { g r * ii d } :{greed}:962
→ greed::NN: { g r * ii d } :{greed}:962

agreed::VBD/VBN/JJ: { @ . g r * ii }> d > :{agree}>d>:32464
morpheme boundary triggers rule
→ agreed::VBD/VBN: { @ . g r * ii; }> d > :{agree}>d>:32464

Like equals and dollar signs, these boundaries also facilitate matching of morphemes across different entries, which aids consistency, and they are used in identifying and substituting exceptions. For example, 'organise' is an exception in Scottish English due to its stress pattern. The morpheme 'organise' occurs in 30 entries in the lexicon, but only needs to be listed once in the exceptions list; other related entries can easily be identified and transformed as necessary.

It should be noted that the morphemic structure of English is complex and a fully comprehensive morphemic annotation has not been attempted. For example, the notation used does not provide a complete bracketing for complex derivations such as <un<{health}>y>; this has not be been disambiguated between 'un' +

'healthy' and the nonexistent 'unhealth' + 'y'. There are many further issues that we cannot address here: for instance

- backformations (e.g. 'chat' derived from 'chatter' – how should we annotate these?)
- speaker knowledge (how many speakers would link 'native', 'nation' and 'pre-natal', and if they do not, is there any merit in annotating these?)
- annotation of a common derivation (e.g. 'premonition') and an uncommon root ('premonish')
- location of boundaries and leftover parts
- blends (can we link 'workaholic' with both 'work' and 'alcoholic'?)
- linking of dissimilar forms (is it possible to link 'coalesce' with 'coalition'?)
- the status of bound/free morphemes ('berry' in 'mulberry' is treated here as a bound morpheme since it co-occurs with another bound morpheme, but as a suffix in 'strawberry' since 'straw is a free root)

A final comment is needed for proper nouns. Where these are divorced from their origin as ordinary words, they are broken down as bound morphemes rather than free morphemes, e.g.:

```
hatfield::NNP: { h * a t =.= f ii l d } :{hat==field}:1430
```

It is assumed that rules which are triggered or blocked by boundaries will not apply to such words; if any future data contradicts this, the transcriptions can be altered accordingly.

3.5. Enriched Orthography

This field mirrors the morpheme breakdown shown in the pronunciation field. It is useful for matching morphemes across words. The enriched orthography was generated automatically by comparing the pronunciation with the spelling and with entries consisting of free roots, and then inserting boundaries in appropriate places. The spelling was modified to match the free roots, for example:

```
bat::NN/VB/VBP: { b * a t } :{bat}:2572
batting::VBG/NN/JJ: { b * a t }.> i ng > :{bat}>ing>:410
```

A certain amount of orthographic adjustment was also made to affixes, so that for example 's' in 'cats' and 'es' in 'churches' are both shown as the same morpheme. Some hand-editing has been done to this field as the automatic algorithm inevitably produces errors, for example linking 'mobilise' to 'mobil' rather than 'mobile'.

3.6. Frequency

The frequency field is taken from a composite of a number of on-line sources of word-frequency. It includes frequencies from the British National Corpus and Maptask, and frequencies derived from Time articles and on-line texts such as Gutenberg. They were weighted to give more importance to sources of spoken speech, and also to increase the numeric frequency of smaller corpuses. The results are in the file ***uni_freqs***.

It is used in, for example, selecting words for diphone recording, where more frequent words are preferred to ensure familiarity. Unfortunately there is currently no method for distinguishing between homographs by frequency. Furthermore, it should be noted that the frequency field, as it was obtained from simple word lists, is not particularly reliable. Like any frequency rating, it is highly dependent on context, so that proper names, for instance, may have an unusually high frequency if they were topical at the time the data was collected. Also, the lists also contain some abbreviations or other usages that are not generally covered by the lexicon (these are left for pre-processors). For example, during recordings it was noticed that the uncommon word

```
gi::NN: { g * ii } :{gi}:2421
```

had been given a high frequency. On returning to the original frequency data it emerged that this was due to use of the term 'G.I.', entailing the following amendment to the dictionary:

```
gi:1,us-soldier:NN: { jh - ii }.{ * ae } :{g}{i}:2421
gi:2,judo-costume:NN: { g * ii } :{gi}:2421
```

Improvements could be made to the frequency field, for example as well as attempting to differentiate between homographs, parts of speech within one entry could be given frequencies, and the frequency of each morpheme could be explicitly annotated.

4. From Text to Accent-Specific Pronunciations

4.1. Overview of Processing Stages

There are various stages needed to produce accent-specific pronunciation strings from the base lexicon, and these have been implemented as perl scripts. The scripts have been tested for perl 5.6.0 under the linux version of UNIX, and MS-DOS under Windows 98. Perl will need to be installed, and in MS-DOS the commands may need be prefixed with 'perl'.

i. Read exceptions into the entries from the base lexicon ***unilex***.

get-exceptions.pl **–a towncode –f** ***unilex*** **> intermediatelexicon**

This creates an accent-specific lexicon containing local exceptions, and is discussed below.

ii. (optional). Concatenate pronunciations from intermediatelexicon to form running text. The form of the script may vary depending on the type of text. At this stage there could also be pre-processing of text (e.g. numbers, dates and so on), selection of pronunciation variants by part-of-speech or semantics, and prosodic features such as stress-shift. These have not been included in the project. A simple text processor is:

read-text.pl **–a towncode –l intermediatelexicon –f sentence-file > fileofstrings**
read-text.pl **–a towncode –l intermediatelexicon –i "sentence strings" > fileofstrings**

This reads in the intermediate lexicon. It then breaks the text file down into sentences, converts it to lower case, and uses ***Readtext.pm*** to replace the orthographic strings with pronunciation strings. This performs simple text processing. For example, numbers are converted to text and some acronyms are converted to letters. A simple selection of stressed or unstressed variants, whereby a word with two entries in the lexicon, for stressed and unstressed, uses the stressed form clause-finally and the unstressed form elsewhere, so we get:

we are there: → #{ w * ii }#.#{ @r r }#.#{ dh * eir }#
we are: → #{ w * ii }#.#{ * ar r }#

These forms are so common that a simple algorithm such as this greatly improves naturalness. The text processing applied here is fairly unsophisticated and is by no means comprehensive.

iii. Apply post-lexical pronunciation processing.

post-lex-rules.pl **[–d] –a towncode –f intermediatelexicon/fileofpronunciationstrings**
post-lex-rules.pl **[–d] –a towncode –i "lexiconline/pronunciationstring"**

This is the core of the pronunciation processing and contains numerous post-lexical rules, which are discussed later.

iv. If desired, the strings produced can be simplified. One possible simplification is removal of morpheme boundaries, which at this stage have performed their functions. Somewhat less straightforward is removal of redundancy. The output string is coded in global keysymbols, and as such contains more symbols than any one accent will use; for some applications such as recording diphones it might be desirable to remove the redundancy in each accent to produce a string containing only the distinctions made in that accent. This also makes it easy to check the distinctions and equivalences which the lexicon and the rules produce for a given accent. The script

map-unique.pl **[–d] –a towncode –f lexiconfile/fileofpronunciationstrings**
map-unique.pl **[–d] –a towncode –i "lexiconline/pronunciationstring"**

reduces the keysymbol redundancy. The output of this program reflects the distinctions and groupings shown in the keysymbol-IPA tables earlier. Some features of ***map-unique.pl*** are discussed below.

v. An example of how these four stages can be combined is:

transform-text.pl **–[uw]** ***unilex***

This utilises ***get-exceptions.pl***, ***Readtext.pm post-lex-rules.pl*** and ***map-unique.pl***, to transform plain text into accent-specific pronunciation strings. It asks for input from the command line or files, and can produce standard output or file output. Accents can be switched within the program, so it is easy to compare the same text in different accents. The following example shows the same text in RP (code rpx) and GA (General American, code gam).

```
Please type input

-a rpx edinburgh is a lovely city
#{ * e . d i m . b r @ }#.#{ i z }#.#{ @ }#.#{ l * uh v }.> l iy >#.#{ s * i . t iy }#

Please type input

-a gam edinburgh is a lovely city
#{ * e . t^ i m . b ~ @@r . r ou }#.#{ i z }#.#{ @ }#.#{ l * uh v }.> lw ii >#.#{ s * i . t^ ii }#
```

These two commands can be combined using

```
-a rpx,gam edinburgh is a lovely city

rpx:  #{ * e . d i m . b r @ }#.#{ i z }#.#{ @ }#.#{ l * uh v }.> l iy >#.#{ s * i . t iy }#
gam: #{ * e . t^ i m . b ~ @@r . r ou }#.#{ i z }#.#{ @ }#.#{ l * uh v }.> lw ii >#.#{ s * i . t^ ii }#
```

A new accent switch requires loading of a new intermediate lexicon, but subsequent switches within the session are more rapid. Options are available for SAMPA output or IPA (html file).

vi. An example of an accent-specific lexicon producer is:

make-lex.pl* –a towncode [-sp] –f *unilex

This calls ***get-exceptions.pl***, ***post-lex-rules.pl*** and ***map-unique.pl*** to produce an accent-specific keysymbol lexicon (note that this is useful for illustrating rule output, but should not be used as the basis for concatenating words; also, the script is slow on large files). Options are available for SAMPA or IPA (html) output.

4.2. Regional Hierarchy

4.2.1. Town descriptions and rule settings

The transformations applied to each town are determined by the place of the town in a regional hierarchy. This is used so that each level will inherit the features of the previous level, unless they are overridden by new features introduced at that level. This removes redundancy from the rule specifications, making it easier to maintain and avoid errors in the rule applications, and making it easier to add new towns as they will automatically take on the features of appropriate higher levels in the specifications.

To encode this, each supported town has a three-letter towncode. For example, Edinburgh has the code "edi". This is linked to an entry in the file ***uni_towns***:

```
$towns{edi}{CNY} = "UK";
$towns{edi}{REG} = "SCOTLAND";
$towns{edi}{TWN} = "EDINBURGH";
$towns{edi}{PER} = "0";
$towns{edi}{NOTES} = "";
```

There are four levels in the regional hierarchy: country; region; town; and person. Settings for wide geographical areas are overruled by settings for local areas, so that if a rule is defined for {CNY} (country), it can be overruled by a definition for {REG} (region) or {TWN} (town). {PER} (person) is the lowest level and overrules all the others

Settings for the rules and mappings are contained in the file ***uni_scores***. This hierarchy is used to specify whether or not given rules apply to the base transcription, and in some cases how the rules apply, for example

```
$rules{do_non_rhotic}{CNY}{UK} = 2;
$rules{do_non_rhotic}{REG}{SCOTLAND} = 0;
$rules{do_non_rhotic}{REG}{S_US} = 1;
```

This states that the non-rhotic rule does applies to the UK with a rule-score of 2, but is overridden (re-set to 0) for Scotland. The rule also applies to Southern US but with a rule-score of 1, i.e. the rule for Southern US is slightly different from the rule for the UK.

Note that there are default settings for all the rules, so the introduction of a new country would take on these defaults until they are overridden, producing a possible (though not necessarily observed) pronunciation. If the addition of a new branch in the hierarchy necessitates new keysymbols, these must be added to the file ***uni_positions***; for checking transcriptions, they must also be added to the file ***uni_rules***.

4.2.2. Inter-speaker variation

Some accent features, such non-rhoticity, are obligatory for a given speaker (inter-speaker variation) while other accent features vary according to the social status of the speaker, the situation, and so on (intra-speaker variation). It is not possible within the current framework to produce intra-speaker variation. So, for example, although speakers may vary the amount of h-dropping they use over the course of a few sentences, in our framework h-dropping must occur completely or not at all.

We can, though, produce inter-speaker variation to tailor pronunciations to a given speaker. This uses a mix of the available transformations and can be done in two ways: by adjusting the scores given in ***uni_scores*** for a listed accent; or, preferably, by creating a new person within the nearest accent in the ***uni_towns*** hierarchy and giving it rule scores in ***uni_scores***.

For example, our Leeds accent uses h-dropping, and full vowels in initial syllables.

```
#FULL INITIAL VOWELS
    $convert{fullinit4}{REG}{N_ENG} = 2;    # full vowels

# H-DROPPING
    $rule{do_h_drop}{REG}{N_ENG} = 1;        # h -> ∅
```

We could produce an accent with full initial vowels but no h-dropping by changing the N_ENG score for h-dropping to 0, or simply omitting this line. It is preferable, though, to maintain the existing accents and add a new Leeds speaker to ***uni_towns***:

```
$towns{lds1}{CNY} = "UK";
$towns{lds1}{REG} = "N_ENG";
$towns{lds1}{TWN} = "LEEDS";
$towns{lds1}{PER} = "LDS1";
$towns{lds1}{NOTES} = "Variant with no h-dropping";
```

It would then take on a score of 2 for the conversion fullinit4 (full initial vowels), and we could set the h-dropping score to 0:

```
H-DROPPING

# H-DROPPING
    $rule{do_h_drop}{REG}{N_ENG} = 1;     # h -> ∅
    $rule{do_h_drop}{PER}{LDS1} = 0;      # h -> h
```

As our new Leeds speaker takes on the existing Leeds settings, we only need specify rules that we wish to alter. We can use the new person specification to determine what happens to exceptions lists, the post-lexical-rules and the unique symbol mappings.

4.3. Exceptions

The first stage in processing is the replacement of entries with exceptions. This must be done before the post-lexical rules as the rules apply to the exceptions.

Exceptions are contained in the file ***uni_exceptions***, and follow the format

```
{REG} = "SCOTLAND" % magazine::NN: { m ~ a . g @ . z * ii n } % { m * a . g @ . z ~ ii n } :{magazine}:
{REG} = "SCOTLAND" % organise::VB/VBP: { * or r . g @ . n == ae z } % { ~ or r . g @ . n == * ae z }
:{organ==ise}:
{TWN} = "GLASGOW" % j::NNP/NN/LS: { jh * ee } % { jh * ae } :{j}:
```

The first part indicates the area in which this is an exception, taking the area definitions used in ***uni_towns*** e.g. {REG} ("region") Scotland. After the first percentage sign comes the entry with pronunciation as given in the

base lexicon, and after the second percentage sign is the exceptional pronunciation. Following that is the enriched orthography. The enriched orthography field is important as it, combined with the original pronunciation, enables the program to match related forms. For example, the entry above for 'magazine' will also match 'magazines', and 'organise' will match 'organising' and 're-organise'. Related forms can sometimes be linked by using the stem for a match, though care must be taken that only the desired forms are matched. For example, the stem

{REG} = "S_ENG" % theory::NN: { th * ii @r . r == % { th * ir . r == :{theor==:

will match 'theory', 'theorem', 'theoretical', etc.. Derived forms are sometimes included in the list if the derivation itself is the exception, e.g.

{REG} = "S_US" % insurance::NN: { i n =.= sh * ur r }.> A5 n s > % { * i n =.= sh ur r }.> A5 n s > :{in==sure}>ance>:21854

The program outputs an intermediate lexicon containing exceptions for the specified accent. This lexicon can then be used in the construction of running text.

4.4. Post-lexical Rules

After inserting exceptions, we can run the post-lexical rules on either the intermediate lexicon or strings of pronunciations produced from the intermediate lexicon.

***post-lex-rules.pl* [–d] –a towncode –f intermediatelexicon/file_of_strings**
***post-lex-rules.pl* [–d] –a towncode –i "lexiconline/strings"**

Note that a number of rules apply across word-boundaries, so we cannot apply the rules to a lexicon and then concatenate the results to form pronunciations for running text.

The ***post-lex-rules.pl*** program takes as an argument the towncode as described above for exceptions. The –d option shows which rules have been applied to transform input to output. The –f option is for file input, and automatically detects whether the file consists of a lexicon or pronunciation strings. The –i option is for standard input of lexicon lines or concatenated pronunciations.

The program contains two types of rules: conversions, which are mostly context-independent and mostly apply to single segments, and rules, which are more complex. Conversions apply first and generally change input symbols to global symbols. The file ***uni_scores*** contains specifications of which conversions apply to which accents, by giving assigning scores for each town for each conversion. The conversions have a default score of 1, since for input symbols a conversion of some kind must take place. For rules, on the other hand, the default score is 0, which indicates that the rule is not applied. These defaults apply if the town, region and country is otherwise unspecified, so that if a new accent is added, it will take on the defaults.

The main motivation for the separation into conversions and rules is that the conversions remove the pre-rule symbols, leaving only global and post-rule symbols. The conversions are applied first and are ordered; the rules follow these and are also ordered.

It should be noted that the split into conversions and rules is not equivalent to a linguistic division such as that between optional and obligatory rules, or between phonemes and allophones. Some conversions relate to obligatory features of an accent, for example the realisation of |[h]| in 'herb' is dictated by accent, while other conversions may vary with speaker choice, for example some uses of syllabic consonants. Allophones, which are obligatory, do appear as rules, for example dark |l|, but so do some speaker variables such as '–ing' → |i n|.

4.4.1. Conversions

As noted above, conversions must apply first. Generally, they are context-free and apply to single (pre-rule) segments, though there are some exceptions to this. For example, the conversion $convert{conv_basic} deals with many of the general pre-rule markers listed in the table earlier as typographical pre-rule conventions. The specification for this in the ***uni_scores*** file is:

```
#BASIC TYPES
#based on us and uk pronunciations
#1(default) for uk-types

    $convertorder{b} = conv_basic;
```

```
$convert{conv_basic}{REG}{IRELAND} = 2; # like US in syllabics
$convert{conv_basic}{REG}{WALES} = 2; # like US in syllabics
$convert{conv_basic}{CNY}{AUS} = 2; # like US in syllabics
$convert{conv_basic}{CNY}{NZ} = 2; # like US in syllabics
$convert{conv_basic}{CNY}{US} = 5;
```

The $convertorder line specifies the order of conversions. This conversion will, for example, convert |[h]| in 'herb' to |h| for accents which score 1, and to null for accents which score 5. For potentially full initial vowels, in for example 'computer', we have:

```
#FULL INITIAL VOWELS
#for full vowels in prefixes etc., marked as caps4.  E0 included here too
#1(default) is /i/ for E0, schwa for others

$convertorder{n} = conv_fullinit4;

$convert{fullinit4}{REG}{N_ENG} = 2;     # full vowels
$convert{conv_fullinit4}{CNY}{AUS} = 3;     # schwa for all
$convert{conv_fullinit4}{CNY}{NZ} = 3;     # schwa for all
```

So, all accents which default to 1 will convert O4 to schwa, while those specified as 2 will convert it to the global keysymbol |o|.

A brief description of the conversions, in order of application, is given below. Examples show the different paths a transcription may take. Conversions changing output are given in red; examples in which the output is unchanged are given in black.

conv_ii2

This converts the |ii2| pre-rule keysymbol in words like 'million' to either |y| or |ii|, and makes any necessary syllable boundary adjustments using a separate function, 'correct_bounds'. This has to be an early rule as the classification as vowel or consonant affects other rule inputs. Scores are 1 or 2.

million	{ m * i . l ii2 @ n }	→	{ m * i . l y @ n }
		↘	{ m * i . l ii @ n }

conv_y

This is used for Irish English, which is similar to US English in use of |[y]|. Scores are 1 (|[y]| is left to be dealt with under the general conventions in conv_basic) or 2 (conversion applies). This conversion must therefore precede conv_basic.

new	{ n [y] * iu }	→	{ n [y] * iu }
		↘	{ n * iu }

This is a fairly crude implementation, as the rule is somewhat variable, and appears to be affected by, amongst other things, frequency. Some other (unimplemented) UK accents also use this conversion.

conv_basic

This converts most of the general pre-rule markers. Scores are 1 (general UK-type), 2 (Ireland, Cardiff, Australia and New Zealand) and 5 (US). In the following examples, scores below 5 undergo the first conversion illustrated, while scores of 5 or above undergo the second.

[x]	→	x	herb	{ [h] * er r b }	→	{ h * er r b }
	↘	∅			↘	{ * er r b }
[x1]	→	∅	figure	{ f * i . g [y1] @r r }	→	{ f * i . g @r r}
	↘	x			↘	{ f * i . g y @r r }
V	→	v	cosmos	{ k * o z . m O s }	→	{ k * o z . m o s }
	↘	@			↘	{ k * o z . m @ s }
V1	→	@	anchovy	{ * a n . ch OU1 . v iy }	→	{ * a n . ch @ . v iy }
	↘	v			↘	{ * a n . ch ou . v iy }
VR	→	vr	mallard	{ m * a . l AR r d }	→	{ m * a . l ar r d }
	↘	@r			↘	{ m * a . l @r r d }
VR1	→	@r	caernarvon	{ k AR1 r . n * ar r . v n }	→	{ k @r r . n * ar r . v n }
	↘	vr			↘	{ k ar r . n * ar r . v n }

x/y	→	x	adolf	{ * a/ee . d o l f }	→	{ * a . d o l f }
	↘	y			↘	{ * ee . d o l f }
(x x/y y)	→	x x	schedule	{ (sh/s k) * e . d y uu l }	→	{ sh * e . d y uu l }
	↘	y y			↘	{ s k * e . d y uu l }

Type 2 accents follow US conventions for syllabic consonants, so that |[@]| is deleted rather than converted to |@| like other UK accents:

norton	{ n * or r . t [@] n }	→	{ n * or r . t n }

Note that this is not a strict rule in each accent; some other UK accents may use syllabic consonants, while some speakers of type 2 accents may use |@|; we have, however, chosen these settings as typical of the accents. This group of accents is similar to the grouping in do_syllab below.

Boundaries are corrected using correct_bounds.

conv_ah2

This changes |ah2| to |ah| (BATH) or |a| (TRAP). It represents a split in the BATH set for accents such as Australian English, and includes 'aunt', demand', and 'plant'. Scores are 1 (conversion to |ah|) or 2 (conversion to |a|).

demand	{ d i . m * ah2 n d }	→	{ d i . m * ah n d }
		↘	{ d i . m * a n d }

conv_fullfinal5

The set of conversions named '5' deal with full vowels in final syllables in Welsh. This one converts |V5| vowels to schwa for score 1 or 3, or |v| (full vowel) in Welsh accents (score 2):

cruel	{ k r * uu . E5 l }	→	{ k r * uu @ l }
		↘	{ k r * uu . e l }

[V50] is deleted for score 1 and 3 , and for score 2 it is converted to |v|:

level	{ l * e . v [E50] l }	→	{ l * e . v l }
		↘	{ l * e . v e l }

V05 becomes an |i| for score 1, and for score 2 it becomes a full vowel, and for 3 it becomes schwa.

fastest	{ f * a s t }.> E05 s t >	→	{ f * a s t }.> i s t >
		↘	{ f * a s t }.> e s t >
		↘	{ f * a s t }.> @ s t >

|[V5]| follows 3 paths: for a score of 1 or 3, and a conv_basic score of 2 or 5, |[V5]| is deleted to give syllabic consonants. For other conv_basic scores, it becomes a schwa. For a conv_fullfinal5 score of 2, it becomes a full vowel.

president	{ p r * e . z i . d == [E5] n t }	→	{ p r * e . z i . d == n t }
		↘	{ p r * e . z i . d == @ n t }
		↘	{ p r * e . z i . d == e n t }

Boundaries are corrected.

conv_iu

This deals with |iu3| diphthongs, a subdivision of |iu|, and also converts |y uu|, |y u|, |y ur| and |ur| following |sh| to |iu| or the |iur| allophone. Scores are 1, which changes |iu3| to |uu| and 2, which changes it to |iu|.

blue	{ b l * iu3 }	→	{ b l * uu }
		↘	{ b l * iu }
argue	{ * ar r . g y uu }	→	{ * ar r . g y uu }
		↘	{ * ar r . g y iu }
pure	{ p y * ur r }	→	{ p y * ur r }
		↘	{ p * iur r }

conv_fullinit4

This deals with full vowels in initial syllables, transcribed |V4|. Scores are 1, 2 or 3, converting |V4| to schwa or a full vowel. E0 is also covered here, and is converted to |i|, |e| or schwa. Necessary syllable boundary adjustments are made by correct_bounds.

computer	{ k O4 m =.= p y * uu t }.> @r r >	→	{ k @ m =.= p y * uu t }.> @r r >
		↘	{ k o m =.= p y * uu t }.> @r r >
expect	{ E0 k . s p * e k t }	→	{ i k . s p * e k t }
		↘	{ e k . s p * e k t }
		↘	{ @ k . s p * e k t }

conv_i_schwa_2

This and the following two conversions simplify the patterning of schwa/i vowels across different accents. Scores are 1 or 2, and map the pre-rule vowels onto |@| or |i| according to accent group. As the conversion is context-free, it could be done by the mapping program, but these vowels are all classed as pre-rule symbols and so are removed by the post-lexical rules. (The gaps in the numberings of these vowels is historical, and dates from a time in the lexicon development when more i/@ combinations were in use.) The default setting of 1 for conv_i_schwa_2 gives |i|.

notice	{ n * ou . t I2 s }	→	{ n * ou . t i s }
		↘	{ n * ou . t @ s }

conv_i_schwa_6

See above. The default here gives schwa.

curate	{ k y * ur . r == I6 t }	→	{ k y * ur . r == @ t }
		↘	{ k y * ur . r == i t }

conv_i_schwa_7

See above. The default here gives |i|.

chases	{ ch * ee s }.> I7 z >	→	{ ch * ee s }.> i z >
		↘	{ ch * ee s }.> @ z >

conv_longschwa

This deals with two subdivisions of |@@r|. It covers a few words of mainly foreign origin, with vowels in non-pre-rhotic environments, such as 'peugeot' and 'goebbels'. These map to |@@r|, |uu| or |ou|. Scores are 1 or 2. Note that the notation |@@r| has been used despite these vowels not being pre-r, since in non-rhotic accents the vowels tend to be pronounced as |@@r|.

chartreuse	{ sh ar r . t r * @@r2 z }	→	{ sh ar r . t r * @@r z }
		↘	{ sh ar r . t r * uu z }
goebbels	{ g * @@r3 . b l z }	→	{ g * @@r . b l z }
		↘	{ g * ou . b l z }

conv_ll

This maps medial |ll| in non-Welsh accents to |th l|. Scores are 1 (apply) and 2 (do not apply):

llanelli	{ ll a n =.= * e . ll i }	→	{ ll a n =.= * e th . l i }
		↘	{ ll a n =.= * e . ll i }

conv_eir_ir

This merges the vowels |eir| and |ir in New Zealand. It is done as an (early) conversion rather than a (late) merger so that they undergo the same rules subsequently. Scores are 1 (do not merge) and 2 (merge)

llanelli { ll a n =.= * e . ll i } → { ll a n =.= * e th . l i }
↘ { ll a n =.= * e . ll i }

conv_general

This conversion applies in the same way to all accents, so is not listed in uni_scores. It makes syllabic consonants explicit:

bottle { b * o . t l } → { b * o . t l! }

This simplifies the specification of later rules.

4.4.2. Rules

The rules follow the conversions, and are given an ordering defined in the file ***uni_scores***. They are mostly context-sensitive. The rules use classes of segments and boundaries which have been defined using the parameters in the file ***uni_positions***. For example, the consonant |p| has the entry:

```
Cons:p  [+ini][+fin][-syl][-vce][+lab][+stp][0bas]
```

It is specified as a consonant, may occur initially and finally, is a voiceless labial stop, and is a global keysymbol. These features are used in ***post-lex-rules.pl*** to create useful classes. The use of classes allows for a range of input, and means that new keysymbols should not require the editing of existing rules, and rules inserted later should not affect existing rules specified in this way. A new vowel, for instance, would be added to the class of vowels and would be accepted wherever a rule required a vowel. For a few rules the features do not provide a useful set; the keysymbols for these must be explicitly specified in the rule, for instance, the Scottish Vowel Length Rule as applied here only affects |ii|, |uu|, and other keysymbols referring to the same phonemes in Scottish English.

An example of a rule is non-rhoticity. A default of 0 means that the rule is not applied; other values take on other specifications. The $ruleorder line explicitly determines the place this rule takes in rule ordering.

```
#NON-RHOTIC
#removes rhotic r in non-rhotic accents
#1 for non-rhotic non-linking, linking, 2 for non-rhotic linking, 3 for non-rhotic linking with intrusive /r/

    $ruleorder{ap} = do_non_rhotic;

    $rules{do_non_rhotic}{CNY}{UK} = 3;          #let's make uk intrusive /r/
    $rules{do_non_rhotic}{TWN}{RP} = 2;          #let's not make rp intrusive /r/
    $rules{do_non_rhotic}{REG}{SCOTLAND} = 0;
    $rules{do_non_rhotic}{REG}{IRELAND} = 0;
    $rules{do_non_rhotic}{CNY}{AUS} = 2;
    $rules{do_non_rhotic}{CNY}{NZ} = 2;
    $rules{do_non_rhotic}{PER}{NYC1} = 3;        #nyc1 has intrusive /r/
    $rules{do_non_rhotic}{REG}{S_US} = 1;
```

So, the US is rhotic except for Southern regions and the non-rhotic version of New York. The UK is non-rhotic except for Scotland and Ireland (and of course other areas which do not appear in this example). As before, {CNY} is overridden by {REG} which is overridden by {TWN} and then by {PER}. As an illustration, the non-rhotic rule includes the following transformations for non-rhotic linking accents:

```
#remove pre-consonantal rhotic |r| in linking accents, regardless of boundary

    s/ r  ((($b)  )*($c)) / $1 /;
```

where $b denotes a boundary of any kind, optionally present, and $c denotes a consonant, and

```
#remove utterance-final |r| in linking accents

    s/ r  ($b_utt) / $1 /;
```

where $b_utt is an utterance boundary.

The rules are listed below. Note that examples only show the effect of the rule in question and not of subsequent rules, so that e.g. 'poor' in the rule "do_ur_or" is shown as a rhotic pronunciation although the final pronunciation is non-rhotic. Examples showing that a rule applies are given in red; examples in which the rule does not apply are given in black. The word boundary symbol, #, is only illustrated in multi-word strings, though it is present in all transcriptions.

do_h_drop

This deletes |h| in some accents, e.g.

hot { h * o t } → { * o t }

Deletion is universal for these accents; ts presence or absence affects subsequent rules. Scores used are 0 and 1.

do_ur_or

This attempts to cover the merger in some accents of |ur| and |or| or |our|, by which 'poor' rhymes with 'law' or 'adore'. It applies to |ur r| before a consonant or word/compound boundary, e.g.

poor { p * ur r } → { p * or r }
poorly { p * ur r }.> l iy > → { p * or r }.> l iy >

poor { p * ur r } → { p * our r }
poorly { p * ur r }.> l iy > → { p * our r }.> l iy >

Scores used are 0, 1 (merge |ur| with |or|), and 2 (merging |ur| with |our|).

do_ur_ir

This rule deals with monophthongisation of |ur| and |ir| in certain environments in certain accents. Scores are 0, 1, 2, 3 and 4. Type 1 is for Leeds and New Zealand and has a monophthong in VrV, e.g. 'beery', with a diphthong elsewhere, e.g. 'beer'.

beer { b * ir r } → { b * ii @r r }
beery { b * ir r }.> iy > → { b * ir r }.> iy > (where |ir| is a monophthong)

Type 2, for Wales, is as type 1 except that |ir| in monosyllabic roots, such as 'beer', is generally |@@r| or |y @@r|; also, the allophone |iur| is included in the rule.

beer { b * ir r } → { b y * @@r r }
beery { b * ir r }.> iy > → { b y * @@r r }.> iy >
serious { s * ir . r ii @ s } → { s * ir . r ii @ s }

Type 3, for Australia, only applies to |ir| and has different diphthongs finally and pre-consonantally (i.e. before |r C|), e.g. in 'beer' and 'beard'.

beer { b * ir r } → { b * ii @r r }
beery { b * ir r }.> iy > → { b * ir r }.> iy > (where |ir| is a monophthong)
beard { b * ir r d } → { b * iir r d } (where |iir| is a diphthongal allophone)

Type 4, for Southern US, only applies to |ir| and transforms this to |ii @| finally and pre-consonantally (i.e. before |r C|), e.g. in 'beer' and 'beard'.

beer { b * ir r } → { b * ii @r r }
beery { b * ir r }.> iy > → { b * ir r }.> iy > (where |ir| is a monophthong)
beard { b * ir r d } → { b * ii @r r d }

do_y_insert

This inserts a |y| before initial |er| and |eir| in Welsh, for example

earl { * er r l } → { y * er r l }

Scores are 0 and 1.

do_non_rhotic

The dictionary contains rhotic |r|, and this rule removes it for non-rhotic accents. There are two basic types of non-rhotic accents: non-linking (like Southern U.S., in which 'far' is pronounced the same before either a vowel or a consonant, i.e. without an |r|); and linking (like RP, in which the |r| in 'far away' is pronounced). Non-linking is listed as type 1:

far	{ f * ar r }	→	{ f * ar }
far away	#{ f * ar r }#.#{ @ . w * ei }#	→	#{ f * ar }#.#{ @ . w * ei }#
safari	{ s @ . f * ar . r iy}	→	{ s @ . f * ar . r iy }

Linking accents are listed as type 2:

far	{ f * ar r }	→	{ f * ar }
far away	#{ f * ar r}#.#{ @ . w * ei }#	→	#{ f * ar r }#.#{ @ . w * ei }#
safari	{ s @ . f * ar . r iy}	→	{ s @ . f * ar . r iy }

We have also implemented a third type of non-rhoticity, type 3, with both linking-r and intrusive-r. This inserts |r| after certain vowels (|aa|, |oo|, |@|, |ah|, |oa|, |@@r|, |@@r2|, |@@r3|, and |i@|) when they precede another vowel. So, for example, 'drawing' becomes |d r * oo r . i ng|, and 'idea of' becomes |#{ ae . d * i@ r }#.#{ @ v }#|. (Pre-rhotic vowels are output as this simplifies the rule specification; these are in any case non-distinctive in the relevant accents, and will be collapsed by the mapping program.)

far	{ f * ar r }	→	{ f * ar }
far away	#{ f * ar r}#.#{ @ . w * ei }#	→	#{ f * ar r }#.#{ @ . w * ei }#
safari	{ s @ . f * ar . r iy}	→	{ s @ . f * ar . r iy }
ma and pa	#{ m * aa }#.#{ @ n d }#.# { p * aa }#	→	#{ m * aa r }#.#{ @ n d }#.# { p * aa }#

Note that this rule does not apply to 'the' and 'to', but if the correct pre-vocalic selection has been made they will not fall into the environment specification. For instance 'the owl' should be |#{ dh ii }#.#{ * ow l }#| in the relevant accents, rather than |#{ dh @ }#.#{ * ow l }#|, and the h-dropping rule should convert 'the horse',|# { dh @ }#.# { h * or r s }#|, to |# { dh ii }#.# { * or r s }#|.

do_sy_zy

This rule converts |. s y| and |. z y| in unstressed syllables to |. sh| and |. zh|, e.g. 'tissue'. The syllable boundary is included in the specification so that the rule can be blocked; it does not apply to transcriptions marked |s . y|, e.g. 'exudate'. Scores are 0, 1, 2 and 3.

All accents using this rule transform |s y| and |z y| before |@| and |@r|:

grecian	{ g r * ii . s == y @ n }	→	{ g r * ii . sh == @ n }

Accents other than those with score 2 transform them before other unstressed vowels as well:

casual	{ k * a . z y uu @ l }	→	{ k * a . zh uu @ l }

Those with score 3 also transform |s ii @| and |z ii @|. (An intervening syllable boundary has been added in ***uni_exceptions*** to block the rule for some words.)

hessian	{ h * e . s ii @ n }	→	{ h * e . sh n}

do_ty_dy

This is similar to the above rule and converts |.t y| and |. d y| to |. ch| and |. jh|. Scores are 0, 1 and 2. 1 and 2 convert all non-initial |t y| and |d y| in unstressed syllables, and word-medial syllable initial |t y|, |d y| and |s t y|:

culture	{ k * uh l . t y @r r }	→	{ k * uh l . ch @r r }
mature	{ m @ . t y * ur r }	→	{ m @ . ch * ur r }

Score 2 applies in Cardiff and makes the further transformations of |t iu| and |d iu|, and applies word-initially also:

tune	{ t * iu n }	→	{ ch * uu n }

do_class

This makes a split in the |ah| vowels for Cardiff, transforming it to |aa| before a fricative:

pass	{ p * ah s }	→	{ p * aa s}

Some words are bled from this rule in ***uni_exceptions***. Scores are 0 and 1.

do_scots_long

There are a number of formulations of this rule in the literature; we are following Scobbie et al. 1999 and only applying it to |ii| and |uu|, and their variants |ir| and |ur|/|iu|/|u|, in stressed position (which includes tertiary stress for compounds). |u| is included as this is equivalent to |uu| in Scottish English, but there do not seem to be any examples of |u| in environments which trigger the rule. Scores can be 0 or 1.

The rule applies at the boundary of a free morpheme, e.g.

agree	{ @ . g r * ii }	→	{ @ . g r * ii; }
agreed	{ @ . g r * ii }> d >	→	{ @ . g r * ii; }> d >
greed	{ g r * ii d }	→	{ g r * ii d }

It also applies before the voiced fricatives |z|, |v|, |dh| and |zh| when the consonants are free-morpheme final, e.g.

freeze	{ f r * ii z }	→	{ f r * ii; z }

and in hiatus, e.g.

leo	{ l * ii . ou }	→	{ l * ii; . ou }

We have applied this rule globally in environments where it may apply, though it actually varies lexically by speaker; this applies also to the comparable PRICE/TIED distinction.

do_tapped_r

|r| is realised as a tap in some environments; this rule implements the tap allophone. Scores are currently 0, 1, 2 and 3. For 1, the environment is intervocalic, regardless of boundaries and stress, and word-initial; a score of 2 has the additional environments of the start of a free unit, and in an initial cluster following |f|, |v|, |p|, |b|, |d| and |th|:

hurry	{ h * uh . r iy }	→	{ h * uh . t^ iy }
three	{ th r * ii; }	→	{ th t^ ii; }
far	{ f * ar r }	→	{ f * ar r }
far away	#{ f * ar r }#.#{ @ . w * ei }#	→	#{ f * ar t^ }#.#{ @ . w * ei }#

A score of 3 gives a tap intervocalically only.

do_dark_l

Some accents differentiate between dark and light |l|. This rule specifies the enviroments and introduces the allophone |lw|. Scores are 0, 1, 2, 3, and 4 Type 1 includes RP, and uses dark |l| word or utterance-finally, or in a syllable coda where it precedes a consonant in the same or following syllable.

feel	{ f * ii l }	→	{ f * ii lw }
feel that	#{ f * ii l }#.#{ dh * a t }#	→	#{ f * ii lw }#.#{ dh * a t }#
feel itchy	#{ f * ii l }#.#{ * i ch }.> iy >#	→	#{ f * ii l }#.#{ * i ch }.> iy >#
feeling	{ f * ii l }.> i ng >	→	{ f * ii l }.> i ng >
liu	{ l y * uu }	→	{ l y * uu }
light	{ l * ai t }	→	{ l * ai t }

Type 2 is used in the US, and has clear |l| where it both follows a word boundary or vowel and precedes a stressed vowel. It is dark if it precedes a consonant, a word-boundary or an unstressed vowel.

feel	{ f * ii l }	→	{ f * ii lw }
feel that	#{ f * ii l }#.#{ dh * a t }#.	→	#{ f * ii lw }#.#{ dh * a t }#
feel itchy	#{ f * ii l }#.#{ * i ch }.> iy >#	→	#{ f * ii lw }#.#{ * i ch }.> iy >#
feeling	{ f * ii l }.> i ng >	→	{ f * ii lw }.> i ng >
liu	{ l y * uu }	→	{ lw y * uu }
light	{ l * ai t }	→	{ l * ai t }

A third type, type 3, is used in the Southern US, and is clear between vowels, or after |ai|, |ae|, |oi| or |y|. It is dark word-finally (preceding a pause or consonant), and dark pre-consonantally except before |y|.

feel	{ f * ii l }	→	{ f * ii lw }
feel that	#{ f * ii l }#.#{ dh * a t }#	→	#{ f * ii lw }#.#{ dh * a t }#
feel itchy	#{ f * ii l }#.#{ * i ch }.> iy >#	→	#{ f * ii l }#.#{ * i ch }.> iy >#
feeling	{ f * ii l }.> i ng >	→	{ f * ii l }.> i ng >
liu	{ l y * uu }	→	{ l y * uu }

light	{ l * ai t }	→	{ l * ai t }
file	{ f * ai l }	→	{ f * ai l }

A score of 4 is like 1 but retains the consonant |ll|.

do_ing

This rule changes |i ng| to |i n| or syllabic |n!| in certain environments. It has scores 0, 1 or 2. To undergo this rule, |i ng| must precede a free boundary (word or free morpheme), and be unstressed. Since lexical units in compounds are given at least tertiary stress, this is a simple way of blocking application of the rule to such words as 'earring'. For score 1, all are converted to |i n|, while for score 2, |i ng| is converted to syllabic |n!| following |t|, |?|, |d|, |s|, |z|, |f|, |v|, |th| or |dh|, and |i n| elsewhere:

thinking	{ th * i ng k }.> i ng >	→	{ th * i ng k }.> i n >
shaving	{ sh * ei v }.> i ng >	→	{ sh * ei v }.> n! >
pudding	{ p * u . d i ng }	→	{ p * u . d n! }
earring	{ * ir r }.{ r - i ng }	→	{ * ir r }.{ r - i ng }

do_t_r

Final |t| following the short vowels |u|, |uh|, |e|, |o| and |i| is transformed to |r|, e.g. in 'shut up'. Scores are 0 and 1.

shut up	#{ sh * uh t }#.#{ * uh p }#.	→	#{ sh * uh r }#.#{ * u p }#

do_vless_plural

Some Welsh accents have a voiceless fricative in plurals where a voiced one would normally be expected, such as 'baths', and a voiceless plural |s| after vowels:

baths	{ b * ah dh $}> z >	→	{ b * ah th }> s >
tomatoes	{ t @ . m * aa . t ou }> z >	→	{ t @ . m * aa . t ou }> s >

Note that in the currently rule system we cannot access category to distinguish plurals from verb forms. Scores are 0 and 1.

do_us_eh

This is for the raised [æ] used in the US (Wells 1982, Vol 3, p. 478). Various levels of application can be set; those implemented are 0, 1, 2, 3, 4, 5 and 6; 0, 1, 3 and 4 are currently used. It applies to |a|, |ah| and |oa|, all of which are realised in the US as TRAP-type vowels, and results in the allophone |eh|. The environment specifies monosyllabic free morphemes (some multisyllabic words are transformed in ***uni_exceptions*** for certain accents). Type 1 applies before a morpheme-final |m| or |n|:

ham	{ h * a m }	→	{ h * eh m }

Type 2 includes the above, and adds the environment before a morpheme-final |f|, |th|, |s|:

half	{ h * ah f }	→	{ h * eh f }

Type 3 includes both of the above, and also morpheme-final |d|, |b|, |g|, |sh|:

bad	{ b * a d }	→	{ b * eh d }

Type 4 includes all the above, and morpheme-final |v|, |z|:

jazz	{ jh * a z }	→	{ jh * eh z }

Type 5 includes all the above, and adds in morpheme-final |p|, |t|, |k|:

hat	{ h * a t }	→	{ h * eh t }

Type 6 includes all the above, and adds in |l| or |lw|:

pal	{ p * a l }	→	{ p * eh l }

In some areas the use of this rule in certain environments is somewhat variable. In some areas there are exceptions which could be result in analysis as a phonemic split rather than allophonic variation.

do_ou_l

Some accents, including many UK ones, differentiate between 'holey' and 'holy'; this rule introduces an allophone |oul| for |ou| or |ouw| followed by |l| or |lw| in the same syllable, and/or an |uul| allophone for |uu|. Variants include type 1, which differentiates 'holey' and 'holy' (e.g. North of England), type 2, which converts both to |oul| (a pre-l allophone not dependent on syllable or morphological structure, e.g. Australia), and type 3, in which both convert to |oul| and also |uh| is neutralised with |o|, and |e| with |a|, in the environment before |l| + consonant. Type 0 (rule doesn't apply):

holey	{ h * ou l }.> iy >	→	{ h * ou l }.> iy >
holy	{ h * ou . l iy }	→	{ h * ou . l iy }

Type 1 ('holey' and 'holy' differentiated):

holey	{ h * ou l }.> iy >	→	{ h * oul l }.> iy >
holy	{ h * ou . l iy }	→	{ h * ou . l iy }

Also, 'food' does not equal 'fool':

fool	{ f * uu l }	→	{ f * uul l }
food	{ f * uu d }	→	{ f * uu d }

Type 2 ('holey' and 'holy' both converted):

holey	{ h * ou l }.> iy >	→	{ h * oul l }.> iy >
holy	{ h * ou . l iy }	→	{ h * oul . l iy }

Type 3 (both converted, and also some neutralisation before |l| + C):

holey	{ h * ou l }.> iy >	→	{ h * oul l }.> iy >
holy	{ h * ou . l iy }	→	{ h * oul . l iy }
gulf	{ g * uh l f }	→	{ g * o l f }
golf	{ g o l f }	→	{ g * o l f }
elf	{ * e l f }	→	{ * a l f }
alf	{ * a l f }	→	{ * a l f }

'Food' does not equal 'fool':

fool	{ f * uu l }	→	{ f * uul l }
food	{ f * uu d }	→	{ f * uu d }

The neutralisation in type 3 applies for some people in all environments before |l|; we have not currently included this.

do_i_reduction

Unstressed |i| is not differentiated from |@| in General Australian, except before a velar; this rule collapses them in non-pre-velar environments. Scores can be 0 or 1.

colin	{ k * oou . l i n }	→	{ k * oou . l @ n }
colic	{ k * o . l i k }	→	{ k * o . l i k }

do_syllab

This rule adds extra syllabic consonants. Scores can be 0 or 1. |@ n|, and |@r n| in non-rhotic accents, are changed to a syllabic consonant after a stressed vowel and an alveolar consonant or any fricative:

pattern	{ p * a . t @r r n }	→	{ p * a . t n! }

They are also changed to syllabic after |s|, |z|, whether following a stressed vowel or not, e.g. in Australian:

sincere	{ s i n . s * ii @r }	→	{ s n! . s * ii @r }

This rule is not wholly determined by accent, since speakers may use it variably, but accent is a useful predictor of the likelihood of syllabic consonants.

do_glottal_stop

This rule introduces a glottal stop allophone for |t|. So far only scores 0, 1 and 2have been used, but there are several other possible formulations. 1 is implemented for a speaker from Aberdeen: glottal stops are used:

before syllabics; before medial unstressed vowels, syllable-finally before consonants, after a vowel and word-finally. In final clusters they are used after |n|, |l|, |r| and syllabic consonants. They are not used word-initially, in syllable-initial clusters, or before stressed vowels within the same word. In the examples, |t| follows a vowel, |n|, |l|, |r| or syllabic consonants, and also:

precedes a word or compound boundary

get	{ g * e t }	→	{ g * e ?}
getaway	{ g * e t }.{ @ . w ~ ei }	→	{ g * e ? }.{ @ . w ~ ei }

|t| is syllable-final before an unstressed vowel or consonant:

atlas	{ * a t . l @ s }	→	{ * a ? . l @ s }

|t| is syllable-initial (but not free-morpheme initial) and precedes an unstressed weak vowel (|ii|, |i|, |iy|, |ii;|, |@|, |@r|, |@@r|, |ou|):

twenty	{ t w * e n . t iy }	→	{ t w * e n . ? iy }

precedes a syllabic consonant:

bottle	{ b * o . t l! }	→	{ b * o . ? l! }
putting	{ p * u t }.> i ng >	→	{ p * u ? }.> i ng >

is in a final cluster:

suits	{ s * uu t }> s >	→	{ s * uu ? }> s >

Examples where the rule doesn't apply include

act	{ a k t }	(|t| follows a consonant other than those specified)
attain	{ @ . t * ei n }	(|t| precedes stressed vowel)
today	{ t @ . d * ei }	(|t| is word-initial)
satire	{ s * a . t aer r }	(|t| precedes non-weak vowel)

For score 2, glottal stops are used before a pause, a consonant, or syllabic |n!|.

do_taps

This rule changes |t| and |d| in certain environments to taps. Scores are 0, 1, 2, 3 and 4. U.S. tapping has the score 1, while tapping in Cardiff has the score 2. Both of these scores apply the rule to |t| or |d| following a vowel, |n|, or |r|, preceding a syllabic |l!| or an unstressed weak vowel |ii|, |i|, |@|, |iy|, |ou|, |@r|, or |@@r|. (The allophone |ii;| is specified in the set but only occurs in Scottish English.) Boundaries may occur after |t| or |d|; a word or compound boundary, or a boundary marking the start of a free morpheme, may not occur before it.

eating	{ * ii t }.> i ng >	→	{ * ii t^ }.> i ng >
party	{ p * ar r . t iy }	→	{ p * ar r . t^ iy }
attack	{ @ . t * a k }	→	{ @ . t * a k }
hypnotise	{ h * i p . n @ =.= t ae z }	→	{ h * i p . n @ =.= t ae z }
low terrain	#{ l * ou }#.#{ t @r . r * ei n }#	→	#{ l + ou }#.#{ t @r . r * ei n }#
extraterrestrial	< ~ e k . s t r @ <.{ t @r . r * e s . t r ii @ lw }	→	< ~ e k . s t r @ <.{ t @r . r * e s . t r ii @ lw }

Score 1 (US) further includes all intervocalic |t| or |d| preceding a word or compound boundary, regardless of stress or vowel identity. (This overlaps with the previous set of environments.)

not always	#{ n * o t }#.# { * oo lw . w ei z }#	→	#{ n * o t^ }#.#{ * oo lw . w ei z }#
great-uncle	{ g r – ee t }.{ * uh ng . k l! }	→	{ g r – ee t^ }.{ * uh ng . k l! }

Score 3 (Ireland) applies only to |t| when intervocalic and word final (or at a compound boundary):

eating	{ * ii t }.> i ng >	→	{ * ii t^ }.> i ng >
attack	{ @ . t * a k }	→	{ @ . t * a k }
low terrain	#{ l * ou }#.#{ t @r . r * ei n }#	→	#{ l * ou }#.#{ t @r . r * ei n }#
not always	#{ n * o t }#.#{ * oo lw . w ei z }#	→	#{ n * o t^ }#.#{ * oo lw . w ei z }#
great-uncle	{ g r – ee t }.{ * uh ng . k l! }	→	{ g r – ee t^ }.{ * uh ng . k l! }

Score 4 (Australian) uses a tap for intervocalic |t|, unless the |t| is free-morpheme-initial or precedes a stressed vowel within the same word.

tahiti	{ t @ . h * ii . t iy }	→	{ t @ . h * ii . t^ iy }
party	{ p * ar . t iy }	→	{ p * ar . t^ iy }
not always	#{ n * o t }#.#{ * oo lw . w ei z }#	→	#{ n * o t^ }#.#{ * oo lw . w ei z }#

hypnotise	{ h * i p . n @ =.= t ae z }	→	{ h * i p . n @ =.= t^ ae z }
low terrain	#{ l * ou }#.#{ t @r . r * ei n }#	→	#{ l * ou }#.#{ t @r . r * ei n }#
extraterrestrial	< ~ e k . s t r @ <.{ t @r . r * e s . t r ii @ lw }		
		→	< ~ e k . s t r @ <.{ t @r . r * e s . t r ii @ lw }
attack	{ @ . t * a k }	→	{ @ . t * a k }

do_hurry_furry

'hurry' and 'furry' and other similar words are not differentiated in U.S. English; this rule converts |uh| to |@@r| when it precedes |r| in the same word.

hurry	{ h * uh . r iy }	→	{ h * @@r . r iy }
furry	{ f * @@r r }.> iy >	→	{ f * @@r r }.> iy >

do_ng_ngg

Free-morpheme final |ng| is pronounced |ng g |:

long island	#{ l * au ng }#.#{ * ae. lw @ n d }#	→	#{ l * au ng }#.#{ * ae. lw @ n d }#

Scores are 0 and 1.

do_short_ii

|ii| in certain unstressed positions is transformed to |iy| (score 1):

harriet	{ h * a . r ii @ t}	→	{ h * a . r iy @ t }

or, for accents where |iy| is merged with |i|, into the allophone |ie| (score 2).

harriet	{ h * a . r ii @ t}	→	{ h * a . r ie @ t }

do_uw

This is similar to the above rule, but for |uu| and |iu|. Scores are 0 and 1.

casual	{ k * a . zh uu @ l }	→	{ k * a . zh uw @ l }

do_sus_velar

This is an allophone rule for Southern US. |i| becomes |ei| before a velar, and |e| becomes |ei| before |g|:

big	{ b * i g }	→	{ b * ei g }
furry	{ l * e g }	→	{ l * ei g }

Scores are 0 and 1.

do_sus_strut_nurse

Again, this is for Southern US; STRUT and NURSE merge before a labial consonant:

strut	{ s t r * uh t }	→	{ s t r * @@r t }
rub	{ r * uh b }	→	{ r * uh b }

Scores are 0 and 1.

do_aai_oow

Another Southern US vowel rule, this makes a monophthongal allophone for |ai| and |ow| except before a voiceless consonant in the same syllable.

time	{ t * ai m }	→	{ t * aai m }
night	{ n * ai t }	→	{ n * ai t }

loud	{ l * ow d }	→	{ l * oow d }
lout	{ l * ow t }	→	{ l * ow t

Scores are 0 and 1.

do_e_i

This merges |e| with |i| before a nasal, for Southern US:

pen	{ p * e n}	→	{ p * i n }
pin	{ p i n }	→	{ p * i n }
pet	{ p * e t }	→	{ p * e t }

Scores are 0 and 1.

do_sus_break

For Southern US, this gives a schwa offlide after the lax vowels /i/, /e/, /a/ (and the |a| counterparts |ah|, |ao|), when they occur before a labial:

web	{ w * e b}	→	{ w * e @ b }
pet	{ p * e t }	→	{ p * e t }
nib	{ n * i b}	→	{ n * i @ b }
nit	{ n * i t }	→	{ n * i t }

Scores are 0 and 1.

do_sus_weak

For Southern US, this reduces unstressed |uu|, |iu|, |ou| and |ouw| in the final syllable of a free root to schwa. Before a vowel |w| is also inserted:

follow	{ f * o . l ouw }	→	{ f * o . l @ }
following	{ f * o . l ouw }.> i n >	→	{ f * o . l @ }.> w i n >
argue	{ * ar . g y uu}	→	{ * ar . g y @}
arguing	{ * ar . g y uu }.> i n >	→	{ * ar . g y @ }.> w i n >

Scores are 0 and 1.

4.5. Mappings

4.5.1. Removing keysymbol redundancy

The script ***map-unique.pl*** is used to remove redundancy in the output of ***post-lex-rules.pl***. It uses the hierarchy in ***uni_towns*** and the scores set in ***uni_scores*** to determine which mappings to apply.

All mappings are context-free, and are specified in the form:

```
$map{ao_a}{CNY}{UK} = 1;          #mazda == trap
$map{ao_a}{CNY}{AUS} = 1;         #mazda == trap
$map{ao_o_aa}{CNY}{US} = 1;       #mazda == lot == palm
```

The symbol in the curly brackets specify equivalences, with the last symbol being retained. So, in the above examples, |ao| converts to |a| for UK and Australian accents, while |ao| and |o| convert to |aa| in US accents. Spaces can be included in order to transform to or from sequences of keysymbols, for example |i@| can be mapped to |ii @|:

```
$map{i@_ii @}{ALL} = 1;           #idea = free-er
```

Only mappings with a score of 1 are applied; mappings should be reset to 0 for a sub-level if they are not needed, e.g.

```
$map{our_or_oo}{CNY}{UK} = 1;          force == north == thought
$map{our_or_oo}{REG}{SCOTLAND} = 0;
```

If a mapping is very common, the default may be set to 1 for ALL, and reset to 0 for places where it does not apply:

```
$map{ouw_ou}{ALL} = 1;                 #know == no
$map{ouw_ou}{TWN}{ABERCRAVE} = 0;
```

The mapping script also removes the word-internal bound morpheme boundaries |==|, but does not remove word boundaries |#|, free morpheme boundaries |{}|, or prefix and suffix boundaries |<>|. Although these are no longer required to determine gross pronunciation differences, it may be desirable to take them into account when recording diphones.

For any accent, the keysymbol set in the output of ***map-unique.pl*** should consist only of distinctive sounds (phonemes or allophones); this has the implication that words which are output with identical keysymbols (though there may be boundary differences) should be pronounced the same, while words with different transcriptions should be pronounced differently. Examples are:

Edinburgh:	horse	#{ h * or r s }#	→	#{ h * or r s }#
	hoarse	#{ h * our r s }#	→	#{ h * our r s }#
	full	#{ f * u l }#	→	#{ f * uu l }#
	fool	#{ f * uu l }#	→	#{ f * uu l }#
RP	horse	#{ h * or s }#	→	#{ h * oo s }#
	hoarse	#{ h * our s }#	→	#{ h * oo s }#
	full	#{ f * u lw }#	→	#{ f * u lw }#
	fool	#{ f * uu lw }#	→	#{ f * uu lw }#

This makes it easier to check that the pronunciations are accurate, and also makes diphone recording easier.

4.5.2. SAMPA mapping

The script ***output-sam.pl*** can be used to make SAMPA output from the transcriptions resulting from ***map-unique.pl.*** In a similar manner to ***map-unique.pl***, it uses the hierarchy in ***uni_towns*** and the scores set in ***uni_sam*** to specify symbol mappings; however, in this case each mapping must be one-to-one, and many-to-one mappings are not allowed. Mappings take the form:

```
$sampa{oul_OU}{CNY}{AUS} = 1;
```

with the first (key)symbol as input and the second (SAMPA) symbol as output.

The keysymbols in the input have different meanings in different accents, so the SAMPA mappings must be specified for each accent. The SAMPA characters are used here to represent phones rather than phonemes, and so the output of ***output-sam.pl*** is accent-independent. The symbols used are from Wells (2000).

4.5.3. IPA mapping

The script ***output-ipa.pl*** takes input in SAMPA format and makes an html document which represents the transcriptions in unicode IPA, which can be read by web browsers if unicode fonts are installed. The output transcriptions are broad phonetic.

5. Regional Accents

The regional accents which are currently supported are as shown below.

Country	Region	Town	Person	Code
UK	S_ENG (Southern England)	RP	0	rpx
	N_ENG (Northern England)	LEEDS	0	lds
			LDS1	lds1
	SCOTLAND	EDINBURGH	0	edi
		ABERDEEN	ABD1	abd1
	WALES	CARDIFF	0	cdf
		ABERCRAVE	0	abc
	IRELAND	COUNTY_CLARE	CCL1	ccl1
AUS	GEN_AUS	GENERAL_AUSTRALIAN	0	gau
NZ	GEN_N_Z	GENERAL_NEW_ZEALAND	0	gnz
US	W_N_US (Western and Northern US)	GENERAL_AMERICAN	0	gam
	S_UK (Southern US)	SOUTH_CAROLINA	0	sca
	E_US (Eastern US)	NEW_YORK	0	nyc
			NYC1	nyc1

Here we give brief notes on features of these. For a summary of the rule scores in a particular accent, ***post-lex-rules.pl***, ***map-unique.pl***, or the sub-command-line of ***transform-text.pl*** can be used with the –d option.

The output transcriptions for these accents represent normal careful speech, i.e. assimilations arnd reductions such as 'next day' → [nɛks dɛɪ] are not generally included. Glottal stops, and other regional features which vary according to formality, are included at a level considered to be typical for relevant accents. Syllabic consonants are shown where they are obligatory (e.g. 'cattle'); for words where the use of syllabic consonants varies, such as 'button', the transcription shows a widespread pronunciation, but is not guaranteed to be the most common one.

We have retained symbol distinctions for pre-rhotic and non-pre-rhotic vowels in rhotic accents, e.g. the START and PALM vowels, or NORTH and THOUGHT; users may prefer to collapse these symbols in some cases.

5.1. British Accents

5.1.1. RP

RP forms the basic accent for the lexicon. All this really means is firstly that words whose pronunciation varies in unpredictable ways are generally listed in the base lexicon in their RP forms, and as exceptions for

other areas, and secondly that the default score of 1 for conversions gives output which follows RP conventions (this does not apply to the default of 0 for rule transformations).

We have followed conservative opinion (though not necessarily usage) for RP in the case of intrusive-r (e.g. in 'drawing', 'law-and-order'), and have not included it in the rule set. It has been included for all other non-rhotic UK accents.

The RP accent is a modern version, with [i] rather than the traditional [ɪ] for the HAPPY vowel, and [ɹ] rather than a tap for the consonant 'r'. We have omitted the rather old-fashiond [j] in words such as 'suit', which is output as [su:t] rather than [sju:t]. Words such as 'sure', on the other hand, are generally kept as [ʊə] rather than [ɔ:], though 'poor' has been listed as an exception to this.

We have included a length difference between |uu| and |uw| ([u:] and [ʊ]), and between |ii| and |iy| ([i:] and [i]), since in words such as 'mod**u**lar' and 'harr**i**et' we wish to indicate a short vowel of the same quality as the corresponding long vowel. We also differentiate between [ʊə] as in 'sure', stressed [u:ə] as in 'fuel', and unstressed [uə] as in 'visual' (|ur|, |uu @| and |uw @| respectively); likewise between [ɪə] as in 'career' and 'idea', stressed [i:ə] as in 'Korea', and unstressed [iə] as in 'axiom' (|i@|, |ii @| and |iy @|).

We have applied a rule which transforms |t| + |y| in certain environments to |ch|, and |d| + |y| to |jh|, so that 'mature', for example, is pronounced with [ʧ] rather than [tj]. A similar rule changes |s| + |y| to |sh|, ('tissue') and |z| + |y| to |zh|. (These rules are used in most of the accents listed here). Some RP speakers do not use these rules.

Many suffixes have pronunciations which vary between [ɪ] and [ə] depending on the speaker, for example '-ed', '-less' and so on. We have chosen a common realisation of each morpheme, and represent the morphemes consistently, so that '-ed' is always [ɪ], '-less' is always [ə], etc. Individual speakers may use different realisations from those shown in our transcriptions.

5.1.2. Leeds

Features of the Leeds accent which are different from RP include h-dropping, ing-reduction, full vowels in initial syllables, intrusive-r, use of the TRAP vowel in BATH words, and use of the same vowel in FOOT and STRUT. The HAPPY vowel is realised as a short [ɪ], unless it precedes another vowel. Dark and light /l/ are not distinguished.

We have included allophonic variation in the NEARING and JURY sets, which is determined by stress and position in the word. There is also a rule which distinguishes between /ou/ before /l/ in the same syllable and same morpheme, which becomes the /oul/ allophone, and /ou/ elsewhere; this permits distinctions between 'holy' and 'wholly'. A similar distinction is made for GOOSE and 'ghoul'. Another rule transforms final /t/ to /r/ when it precedes a vowel and follows a limited set of short vowels, for example in 'get off'.

Like the other accents, we are using a modern version of the Leeds accent, so some traditional dialect vowel distinctions are not included. For illustration of speaker differences, we have included lds1, a Leeds speaker with no h-dropping. (This speaker is not illustrated in the symbol table in Appendix III.)

5.1.3. Edinburgh

Edinburgh, like other Scottish accents, is rhotic, merges PALM and TRAP, and FOOT and GOOSE, while retaining the distinctions in WITCH-WHICH, NORTH-FORCE, and NURSE-PERT. Scottish English does not have light and dark |l|. While START may use the same vowel as PALM, many speakers have noticeable difference in quality, so we have retained a distinction in the transcriptions.

We have applied the ing-reduction rule, and a syllabic consonant rule which transforms some |@ n| sequences to syllabic |n|. There is also a rule converting |r| to a tap in certain environments, for instance between vowels, as in 'merry', and of course the Scottish vowel length rule. Our formulation of this applies only to [ʉ] and [ɨ]. The Edinburgh accent here also uses glottal stops in certain positions.

We have not included the [ʉ] pronunciation which may be used for some words in the MOUTH set, or dialect forms such as [sten] for 'stone', [hid] for 'head', and so on.

5.1.4. Aberdeen (1)

This Aberdeen implementation was set to describe the pronunciations of a particular speaker from Aberdeen. This speaker does not have a traditional Aberdonian accent, and so the settings are similar to those for Edinburgh, with a slight difference in the formulation of the tapped |r| rule. Examples of diphone synthesis from this speaker, with transcriptions, are available on the project web pages http://cstr.ed.ac.uk/projects/unisyn.html.

5.1.5. Cardiff

Cardiff is the largest city in South Wales. Some of its features, however, are like Southern English rather than Welsh English, for example it distinguishes between dark and light l. Descriptions of the phonetics of Cardiff English vary, suggesting that the realisations of phonemes vary across speakers.

This accent is non-rhotic. It merges STRUT with schwa, but in final closed syllables full vowels are generally used instead of schwa, in for example 'movement'. and 'ear' rhymes with 'yearn'. Like Leeds, there is allophonic variation for the NEARING and JURY vowels before |r| + vowel, and also h-dropping. The GOAT and 'goal' vowels are distinguished, as are GOOSE and 'ghoul'. The BATH vowel is generally equivalent to TRAP, but before a fricative the vowel is pronounced as PALM. |t| becomes a glottal stops before syllabic |n!|, or word-final before a pause or vowel, and it becomes a tap intervocallically. Intervocalic |r| is also a tap.

There is an extra diphthong, |iu|, (BLUE) in opposition to |uu| (GOOSE). Unless the rules are used in lexicon rather than string mode, our rule system is at present unable to make the distiction between words without orthographic 'y' such as 'use' (|* iu z|), and words with orthographic 'y' as in 'you', which is pronounced |y * uu| and should not be input to the conv_iu function. This illustrates an argument for reading orthographic information into the rule set (see the suggestions for further development in Section 7).

Welsh English uses the consonant |ll| as in LLANDUDNO, and |x| as in LOCH.

The FIRE vowel, |aer|, is broken into two syllables, as is the HOUR vowel, |owr|.

Some of the most noticeable features of Welsh English are rhythm and intonation; these are not covered by the transcription system here.

5.1.6. Abercrave

Abercrave is a small town but was added to illustrate some of the typically South Wales features which the larger city of Cardiff lacks. It has clear rather than dark |l|, and uses a distinction between GOAT and KNOW, and between WASTE and WAIST.

There is a devoicing rule which affects |th| in words like 'paths', and |z| in plurals (as for the conv_iu conversion, we only have access to the phonological transcription when in string mode and so are unable at present to use part of speech information to distinguish plurals from 3rd person verbs). Unlike Cardiff, The BATH vowel is always equivalent to TRAP, and there is a distinction between NORTH and FORCE. We have turned off |t y| and |d y| assimilation for Abercrave, and also |s y| and |z y|.

5.1.7. County Clare (1)

This accent description was based on an individual speaker, and so may not reflect the majority of inhabitants of this area. Examples of synthesised speech are available on the project web pages http://cstr.ed.ac.uk/projects/unisyn.html.

This accent is rhotic. It uses |x| as in LOCH, and |hw| as in WHICH. |t| is are realised as a tap word-finally before vowels. |s y| and |z y| assimilation are not used in as many environments as in most other accents. This speaker does not use 'ing'-reduction. NORTH and FORCE are distinguished, but NURSE and PERT are not. PALM is distinguished from TRAP mainly by length.

There are a number of realisational differences from, for instance, RP, which are not actually systemic differences; these will be evident using SAMPA or IPA output.

5.2. Australian Accents

5.2.1. General Australian

Australian English is similar in many ways to Southern British English accents, for example it is non-rhotic. One rule particular to General Australian is |i| reduction, where unstressed |i| is changed to schwa except before a velar. We also have a split |ah2| in the BATH set, |ah|, which reflects a split in Australian English. There is an allophone specific to Australian English, |iir|, which reflects a variant of NEARING. |t| is tapped between vowels.

We have implemented a rule transforming |ur| to the |or| of NORTH; this applies to some extent in many British accents too, but is more widespread in Australian. We have also applied the h-dropping and ing-reduction rules to this accent.

There are a number of phonetic differences between Australian English and Southern British English, which can be seen in the SAMPA or IPA output modes.

5.3. New Zealand Accents

5.3.1. General New Zealand

This is similar to General Australian. We have included tapped |t|, h-dropping and ing-reduction as in Australian.

There is no distinction between the KIT and COMMA vowels; this is done here by a symbol merger. Some people have vowel mergers in the front diphthongs, such as |ir| in 'beer' and |eir| in 'bear' (Gordon and Mclagan 1989), and we have included this merger (it is done as an early conversion in the rules, so the symbols may undergo the same rules).

5.4. U.S. Accents

5.4.1. General American

General American, combined with RP forms the primary basis for settings in the lexicon; for example, the square bracket notation for deletion of segments was based on differences between General American and RP. Mergers of the 'o'-vowels (e.g. THOUGHT, LOT) and the 'a'-vowels (e.g. TRAP, PALM) are noticably different in British English and American accents.

General American is, of course, a rhotic accent. |t| and |d| are tapped in certain environments. There are generally more syllabic consonants than in RP, and a tendency to schwa where RP has |i|, for example in 'hors**e**s'. Dark and light |l| are distinguished, but in different environments from RP. We have included assimilation of |s ii| and |z ii| to |sh| and |zh|, as in 'gla**ci**al'; we have also made a distinction between WITCH and WHICH, though this is not used by all speakers. CLOTH merges with THOUGHT, rather than with LOT as in British accents. There are fewer vowel transformation rules than for most other accents, but there is a rule which merges |uh| and |@@r| before an |r|, so that 'hurry' rhymes with 'furry'. There is also a rule for raised [æ̝] in words such as 'Ann'; this is in many accents across the US, but the environments where it applies vary by accent. There are a variety of possible mergers in sets such as 'Mary', 'merry' and 'marry'; we have not included these, but they could be added for individual speakers.

5.4.2. South Carolina

This is a non-rhotic accent with no r-linking. WITCH and WHICH are distinguished. Dark |l| and light |l| are used, but in a slightly different set of environments to General American. We have also applied 'ing'-reduction.

The vowel system in this area is very complex, and at this point the keyword listing breaks down somewhat, since some of the words we have chosen as keywords are affected by allophone rules; the symbols in Appendix III are therefore not always used in the keywords listed.

A rule transforms [ɪ] before velars, and [e] before [g], to [eɪ]. BATH-raising is applied (note that this affects BATH itself). FORTH and NORTH are distinct, but (similar to Australian English) in some environments FORCE and |ur| are merged. Southern US accents tend to favour |i| over schwa in many derivational endings. There is a wide range of vowel allophones resulting from Umlaut and Shading (see Wells Vol III, p. 534), which make the vowels either fronted or retracted. We have not included these two processes in our rule set due to the large number and continuous nature of the allophones resulting from each base vowel; Wells quotes Bailey as showing that there are up to eleven degrees of fronting of [ɪ] and [ʊ] due to these two processes; for synthesising an individual speaker, however, it might prove useful to add them in if they are used by the speaker. We have, however, included a rule of Breaking which gives the lax front vowels [ɪ], [ɛ] and [æ] a schwa offglide in certain environments, affecting amongst other things the word TRAP. Another rule merges |uh| with |@@r| except before a labial (thus affecting STRUT) while a further rule creates a long monophthongal allophone from the PRICE vowel. Yet another rule merges |i| with |e| before a nasal, and there is a rule reducing unstressed [oʊ] and [u] in morpheme-final syllables. Vowels plus a nasal consonant may become nasal vowels; for concatenative synthesis this should emerge from selection of suitable units.

5.4.3. New York

New York is often stated to be a non-rhotic accent, but in fact this varies, as since rhoticity seems to be on the increase we have made it rhotic. There is also a non-rhotic version enabled, code nyc1, which due to r-dropping has a few extra symbol mergers, such as |or| for NORTH with |oo| for THOUGHT.

WITCH and WHICH are not distinguished.

Like Welsh English, New York has an [ɪu] diphthong in words such as 'blew', though it is used in a smaller set of words; the word set where it is used varies to some extent by speaker, so we have only been able to show a representative selection. We have not included the well-known [ɜɪ] realisation of words such as 'bird', since this in decline.

The raised version of BATH is applied to a wider range of environments than in General American. (Note that it applies to BATH itself, so that after the rules are applied the word BATH does not accurately describe the word set for this accent.) There is also a rule specific to New York (amongst the accents here) by which |ng| at the end of a free morpheme is realised as |ng g|, for example in 'Long Island' (Wells 1982).

6. Summary: How to Use the Lexicon

This section gives a summary of the main functions in the lexicon and rules, and describes how to operate, extend or edit the systm. Variables and file-names to be specified by the user are given in green. Basic command-line options, which are not all illustrated here, can be shown by typing the program names.

6.1. Producing accent-specific output

6.1.1. Producing an accent-specific lexicon

1: ***get-exceptions.pl*** **–a towncode –f** ***unilex*** **> intermediatelexicon**

2: ***post-lex-rules.pl*** **-a towncode –f intermediatelexicon > output1**

3. ***map-unique.pl*** **–a towncode –f output1 > output2**

Typing 'get-exceptions', or 'post-lex-rules' gives a list of available towncodes and other options.

Useful options: -d, on post-lex-rules or map-unique, gives the rule path for each transcription. (If using –d on post-lex-rules, the output is not usable for map-unique.) Further options are:

4: ***output-sam.pl*** **-a towncode –f output2 > output3**

This shows accent-specific SAMPA (each phonemic-level keysymbol in output2 is shown as a phonetic-level SAMPA symbol).

5. ***output-ipa.pl*** **–f output3 > output4.html**

This can be applied to the SAMPA transcriptions, and produces an html document with unicode ipa transcriptions.

Or, these are all combined in:

make-lex.pl **–a towncode –f** ***unilex*** **> outputlexicon**

This combines the previous three programs; running this on the whole lexicon is rather slow. Useful options: -s gives output in SAMPA rather than keysymbols; -p produces an html-format document showing IPA for the transcriptions.

Note that an accent-specific lexicon treats words as sentence-final for the purposes of post-lexical rules, and so it cannot be used to form running text.

6.1.2. Producing accent-specific running text

(1) Method 1 (automatic)

transform-text.pl **–[uw]** ***unilex***

This takes standard-input commands, and does all the necessary stages to transform text to pronunciations. Options are:

usage: [-a towncode] [options] input text

The following settings persist until changed:

to change towns:	-a towncode (obligatory on first use)
to list town codes:	-a ?
to use multiple town codes	-a xxx,yyy,zzz

output simplified for festival:	-form fest
output in SAMPA:	-form sam
output an html IPA file:	-form ipa (use with >)
output in keysymbols (default):	-form key
line nums repeated frominput:	-nums y
line nums not parsed (default):	-nums n
text included in output:	-text y
text not included (default):	-text n
show rule breakdown:	-d
plain output (default):	-o

These settings do not persist:

text input from file:	-f text_file
text input from keyboard:	text strings
output to file:	> outfile
append to file:	>> outfile

For the examples, user input is shown in green, program output in blue and comments in black

Example 1:

-a rpx

reading lexicon for RP, person 0, please wait...

(loads the lexicon and exceptions for rpx, i.e. RP)

Please type input

hello there

(sets lexicon to RP, runs Readtext.pm, post-lex-rules and map-unique)

#{ h @ . l * ou }#.#{ dh * eir }#

Example 2:

Please type input

-a rpx,edi

reading lexicon for Edinburgh, person 0, please wait...

(RP already loaded, now loads the lexicons and exceptions for edi, i.e. Edinburgh)

hello there

(for each accent, sets lexicon to that accent, runs Readtext.pm, post-lex-rules and map-unique)

rpx: #{ h @ . l * ou }#.#{ dh * eir }#

edi: #{ h @ . l * ou }#.#{ dh * eir r }#

Example 3:

Please type input

-a edi –f infile > outfile

(sets the lexicon to Edinburgh)

(reads in 'infile', which should contain only text, e.g. words, sentences)

(runs Readtext.pm, post-lex-rules and map-unique)

(prints accent-specific transcription to outfile)

(output file contains #{ h @ . l * ou }#.#{ dh * eir r }#)

Example 4:

Please type input

-a rpx –text y –form sam hello there

(sets the lexicon to RP)

(runs Readtext.pm, post-lex-rules and map-unique)

(runs output-sam)

hello there.: h@"l@U "DE@

(2) Method 2 (manual)

1: ***get-exceptions.pl*** **–a towncode –f** ***unilex*** **> intermediatelexicon**

2: ***read-text.pl*** **–f sentence-file –l intermediatelexicon > fileofpronunciationstrings**
or use own text processor to process punctuation etc. to make suitable text strings, and use own dictionary lookup to make a concatenated pronunciation string from the transcriptions in intermediatelexicon

4: ***post-lex-rules.pl*** **-a towncode –f fileofpronunciationstrings > output1**

or ***post-lex-rules.pl*** **-a towncode –i "strings from intermediate lex" > output1**

5. ***map-unique.pl*** **–a towncode –f output1 > output2**

or ***map-unique.pl*** **–a towncode –i "strings output from post-lex-rules.pl" > output2**

Further options are:

6: ***output-sam.pl*** **-a towncode –f output2 > output3**

or ***output-sam.pl*** **-a towncode –i "strings output from map-unique.pl" > output3**

7. ***output-ipa.pl*** **–f output3 > output4.html**

or ***output-ipa.pl*** **–i "strings output from output-ipa.pl" > output4.html**

Improving the pre-processor

The pre-processing supplied in ***Readtext.pm*** is fairly basic. The calling program (***read-text.pl*** or ***transform-text.pl***) groups words into sentences. ***Readtext.pm*** does some simple processing of numbers and acronyms. It groups the input into clauses, if applicable, and stores word-classes of the words in the immediate context: this information is used to select some stressed or unstressed variants. Obvious areas for improvement include allowing for a wider variety of text-types (e.g. e-mail formatting), including a wider variety of abbreviations, more sophisticated pre-processing of dates etc., and sentence parsing to give an appropriate choice for homographs.

6.2. Adding to the accent repertoire

Here are the main steps which might be required for adding a new accent. . When running post-lex-rules, errors in the specification of towns, rules and so on, for example setting scores for towns which do not exist in the towns file, will usually result in error messages.

6.2.1. Adding a new accent or speaker

(1) Placing in the hierarchy

In the file ***uni_towns***, add a new entry of the format

```
$towns{new}{CNY} = "C_NEW";
$towns{new}{REG} = "R_NEW";
$towns{new}{TWN} = "T_NEW";
$towns{new}{PER} = "0/NEW[1-9]";
$towns{new}{NOTES} = "";
```

where:

"new" is the towncode, and may consist of 3 letters plus optional numbers

0/NEW[1-9]: 0 is the standard setting for that accent, [1-9] is a number from 1-9, and differentiates speakers within an accent. By convention, a number of 1 or above is added to the three letters in the towncode, while 0 is not included in the towncode.

T_NEW is the town name, and should be a maximum of 20 upper-case characters

R_NEW is the region name, and should be a maximum of 8 upper-case characters

C_NEW is the country name, and should be a maximum of 5 upper-case characters

Notes are optional, and appear in the 'usage' output for each relevant script

These fields should not contain spaces or non-ASCII characters; E,W,N,S are used to denote East, West etc.

The new person, town, or region should be put in the most appropriate place in the hierarchy; check other town entries to ensure that upper levels of the hierarchy of the new entry are exactly the same as other relevant entries.

The examples below can generally apply to any one of these levels by specifying {TWN}, {REG} etc. as appropriate.

(2) Setting scores

In the file ***uni_scores***, for every conversion or rule which deviates from the upper levels in the hierarchy, set the scores as appropriate. Conversion scores may be 1-9, rule scores may be 0-9. For example, for a new rhotic region within the UK, find the non-rhotic rule:

```
#NON-RHOTIC
#removes rhotic r in non-rhotic accents
#1 for non-rhotic non-linking, 2 for non-rhotic linking, 3 for non-rhotic linking with intrusive /r/

    $ruleorder{ap} = do_non_rhotic;
    $rules{do_non_rhotic}{CNY}{UK} = 3;
```

and add the line

```
    $rules{do_non_rhotic}{REG}{R_NEW} = 0;
```

Using the –d option of ***post-lex-rules.pl*** is useful for checking output on particular examples, and also for comparing the scores given to each rule.

If the ruleorder is a problem for the new accent, this can be adjusted by changing the two letters in $ruleorder{} for rules, or $convertorder{} for conversions; an alphabetic sort is used; existing accents may of course be affected by this.

(3) Setting mappings

In the file ***uni_scores***, for a mapping which deviates from the upper levels in the hierarchy, set the scores as appropriate. For example, the UK level collapses the vowels of FORCE, NORTH and THOUGHT, and outputs |oo| (the final vowel in the list):

```
$map{our_or_oo}{CNY}{UK} = 1; force == north == thought
```

To override this for a new town, add a line such as

```
$map{our_or_oo}{TWN}{T_NEW} = 0;
```

Mapping settings are all 1 or 0. Note that mappings are order-independent. So, if we want to define a new (overlapping) mapping for our town, say

```
$map{our_or_oou}{TWN}{T_NEW} = 1;
```

we still need to switch off the previous mapping by setting the score to 0; otherwise we might get the wrong result as we do not know which mapping will apply first.

In the file ***uni_sam***, check that the mappings inherited by the new accent are appropriate and complete. Mappings take the form:

```
$sampa{ou_o}{CNY}{UK} = 1;
```

Unlike uni_scores, unwanted mappings do not need to be overridden with a 0 setting. The SAMPA mappings are all one-to-one, the first (key)symbol being converted to the second (SAMPA) symbol.

6.2.2. Adding new exceptions

In the file ***uni_exceptions***, add a line of the format:

```
{REG} = "N_REG" % hotel::NN/NNP: { h ou . t * e l } % { h * ou . t e l } :{hotel}:
```

for whole words, or, for part-words:

```
{REG} = "N_REG" % theory::NN: { th * ii @r . r == % { th * ir . r == :{theor==:
```

The section from % to % is taken from ***unilex***; the pronunciation given between the second % and the next : is the new pronunciation, and the remainder gives the enriched orthography. Only the base word needs to be added to the list. If required, ***get-exceptions.pl*** can produce a log (currently commented out of the script), 'exceptions_log', which lists the words which have been altered (noted with ->) and words which matched the orthography but not pronunciation, and have not been altered (noted with !->).

6.2.3. Adding new rules

Generally, conversions are used to eliminate non-basic keysymbols, and are usually context-free; rules are used for other processes.

(1) Conversions

For a new conversion, to the file ***uni_scores***, add a rule in the form:

```
#DESCRIPTION
#more notes
#summary of score meanings
    $convertorder{mn} = conv_new;
    $convert{conv_new}{REG}{R_NEW} = [2-9];
```

where 'conv_new' is the conversion name, 'mn' is the location in the conversion order, and [2-9] is a score from 1 to 9, and {REG}{R_NEW} (or {TWN}{T_NEW} etc.) covers the new town entry. Note that for conversions, 1 is the default; this example assumes that other existing towns will have the default score of 1 for the new conversion.

To the file ***post-lex-rules.pl***, add a function of the type:

```
sub conv_new {
    if ($convertscore{$convert} == 1) {
        while (s/ X / y /) {
            &debug_format(__LINE__);
        }
    }
    elsif ($rulescore{$rule} == N) {
        while (s/ X / z /) {
            &debug_format(__LINE__);
        }
    }

}
```

where 'conv_new' is the new conversion name, 'N' is the score assigned in ***uni_scores***, '$convertscore' and '$convert' are automatically derived variables and should not be edited, and the functions 's/ X / y /' and 's/ X / z /' specify the transformations required on the transcription. Some character sets are available, e.g. ($c) represents any consonant; see the program ***post-lex-rules.pl*** for details. More sets can be added to ***post-lex-rules.pl*** using the feature descriptions in ***uni_positions***.

(2) Rules

For a new rule, to the file ***uni_scores***, add a rule in the form:

```
#DESCRIPTION
#more notes
#summary of score meanings

    $ruleorder{mn} = do_new_rule;
    $rules{do_new_rule}{REG}{R_NEW} = [1-9];
```

where ‘do_new_rule’ is the rule name, ‘mn’ is the location in the rule order, and [1-9] is a score from 1 to 9, and {REG}{R_NEW} (or {TWN}{T_NEW} etc.) covers the new town entry.

To the file ***post-lex-rules.pl***, add a function of the type:

```
sub do_new_rule {
    if ($rulescore{$rule} == N) {
        while (s/ x  y / z  y /) {
            &debug_format(__LINE__);
        }
    }
}
```

where ‘do_new_rule’ is the new rule name, ‘N’ is the score assigned in ***uni_scores***, ‘$rulescore’ and ‘$rule’ are automatically derived variables and should not be edited, and the function ‘s/ x y / z y /’ specifies the transformation required on the transcription. All characters should be separated by double spaces. Some character sets are available, e.g. ($c) represents any consonant; see the program ***post-lex-rules.pl*** for details. More sets can be added to ***post-lex-rules.pl*** using the feature descriptions in ***uni_positions***.

It may be that, rather than adding a new rule or conversion, an existing rule needs modification in the light of a new accent; see Section 6.3.3..

6.2.4. Adding mappings

In the file ***uni_scores***, add a mapping of the format

```
$map{xxx_yy_zz}{TWN}{N_TWN} = 1;
```

where the keysymbols |xxx|, |yy| and |zz| will all be collapsed into |zz|. Any number of keysymbols can be contained in a mapping, and spaces are also permitted in either input or output, e.g.

```
$map{i@_ii @}{REG}{S_ENG} = 0;
```

Mappings are context-free and order-free. So, as noted earlier, if the new mapping overlaps with an already existing mapping at a higher level in the hierarchy, we must turn off the mapping which already exists by adding a line such as

```
$map{xxx_qq}{TWN}{N_TWN} = 0;
```

where xxx_qq is the old mapping.

Mapping settings are all 1 or 0. Note that a mapping can be set for all towns using

```
$map{xx_yy_zz}{ALL} = 1;
```

and overridden for the new town with

```
$map{xx_yy_zz}{TWN}{N_TWN} = 0;
```

which is useful if adding a new keysymbol which is not needed for other accents. Of course, mappings may apply to regions or countries as well as towns.

For SAMPA mappings, In the file ***uni_sam***, add a mapping of the format

```
$sampa{aa_b}{TWN}{N_TWN} = 1;
```

where the keysymbols |a|, will be converted into the SAMPA symbol 'b'. Occasionally new SAMPA characters may need quoting in the 'format-ipa' function of ***transform-text.pl*** for the IPA conversion option to function correctly. (See http://www.unicode.org, or Alan Wood's Unicode resources at http://www.hclrss.demon.co.uk/unicode/index.html, which give decimal numbers for unicode characters.)

6.2.5. Adding new keysymbols to the base lexicon

If a new accent requires a new keysymbol, whether basic or derived to represent an allophone, this needs to be added to ***uni_positions***, with features (look at similar symbols and notes in ***uni_positions*** to get features). It should be added to ***uni_rules***, for checking the validity of transcriptions. It is likely that new mappings will be needed in ***uni_scores*** to map the symbol back for existing accents (see Section 6.2.4. above). New mappings may also be needed in ***uni_sam*** (see above) and ***output-ipa.pl***.

New keysymbols should follow the conventions for existing symbols, namely:

i. Vowels should contain at least one ASCII vowel.
ii. Pre-rhotic vowels should end in |r|.
iii. Consonants should contain no ASCII vowels.
iv. Square brackets, round brackets, forward slashes, numbers and capitals are only used on pre-rule symbols and should not appear in output transcriptions.
v. Capitals are only used on vowels, and indicate vowels which may be reduced to schwa; numbers combined with capitals indicate other kinds of reduction.
vi. The pre-rule marker 1 is reserved for US-UK relationships; pre-rule markers with no numbering are reserved for UK-US relationships.
vii. As far as possible, keysymbols should bear some relationship to the normal orthography for the sound.

The keysymbol then needs to be added in the base lexicon ***unilex***; the simplest method is to extract words containing the symbol or symbols which will be edited, then edit those. The enriched orthography field is useful for finding related word forms, as these are likely to use the same keysymbols; it can also be useful for narrowing the search in cases where the new keysymbol is closely related to particular spellings.

If any of the edited words are in the file ***uni_exceptions***, these entries should also be edited. Rules and conversions in ***post-lex-rules.pl*** will mostly not be affected by new keysymbols as rules are specified where possible in terms of broad categories, but some rules are specified in terms of individual symbols and these should be edited where necessary to include the new symbol.

6.3. Modifying an existing accent

It is generally preferable to add a new speaker within an accent rather than modifying an existing accent; the settings can then be changed for the speaker without deleting existing options. However, if modifying an existing accent is preferred, these are the steps.

6.3.1. Changing exceptions

In the file ***uni_exceptions***, edit the exceptions.

6.3.2. Changing rule scores

In the file ***uni_scores***, change the scores for the conversions and rules to be changed.

6.3.3. Changing rule functionality

In the file ***post-lex-rules.pl***, edit the conversions or rules as desired. The –d option is useful for examining the resulting output. If the change involves a subset of the rule (for example, incorporating a new set of environments in the existing glottal stop rule), add a new score for the rule in ***uni_scores***, then extend the rule functionality in ***post-lex-rules.pl*** using transformations triggered by this score.

6.3.4. Adding new rules

See above, Section 6.2.3..

6.3.5. Changing symbol mappings

In the file ***uni_scores***, and/or ***uni_sam***, change the scores for the rules and conversions to be changed.

6.4. Extending the base lexicon

6.4.1. New entries

New entries should follow the lexicon format described earlier, which can be checked automatically (see Section 6.5. below). Word frequency for the frequency field can be taken from ***uni_freqs***.

6.4.2. New keysymbols

See Section 6.2.5. above.

6.5. Checking input and output

***check-trans.pl* –[ld] file**

This checks the format of lexicon files, including field numbers and their contents, and keysymbols. The –l option checks accent-independent input lexicons, and –d checks accent-specific output lexicons. Permitted keysymbols are specified in ***uni_positions***, and are different for input and output lexica. Basic checks are done for RP and General American in input files, but there are too many permutations to cover all accents; all accents should, however, be covered by the –d option.

***check-phon.pl* –[ld] –[01] file**

This automatically detects lexicon format or concatenated pronunciation strings, parsing the transcriptions into syllables and checking some features of syllable structure. The –l checks accent-independent input, and –d checks accent-specific output. –0 outputs a minimum of error warnings, and –1 reports uncommon consonant clusters, etc. The program uses ***uni_positions*** and ***uni_rules***. Note that not all error warnings should actually be classed as errors in all accents; this program is a guide and not an infallible mechanism.

7. Suggestions for Further Development

Here are some suggestions for development and improvement of the lexicon and scripts:

- Addition of more accents
- Extension of the pre-processor to cover a wider variety of text types
- Extension of the rule set to cover general phonological processes, such as elision and assimilation
- Adjustment of the rule system to read in and use non-phonological information, for example part of speech and word-frequency may affect some rules. Related to this, we have not included morpheme frequency in the lexicon, only the frequency of the orthographic form; morpheme frequency, however, could affect some rules
- At present, individual words cannot be specified as exceptions to rules, or as input to rules, other than by giving their phonological form in the rule specification; this would be a desirable feature

Appendix I: Bibliography

Abercrombie, David (1979). The accents of Standard English in Scotland. In: Adam J. Aitken and Tom McArthur (eds), *The languages of Scotland*, pp. 68-84. Edinburgh: Chambers.

Agutter, Alex (1988). The not-so-Scottish vowel length rule. In: John M. Anderson and Norman Macleod (eds), *Edinburgh studies in the English languag*e, pp. 120-32. Edinburgh: John Donald.

Aitken, Adam J. (1979). Scottish speech: a historical view with special reference to the Standard English of Scotland. In: Adam J. Aitken and Tom McArthur (eds), *The languages of Scotland*, pp. 68-84. Edinburgh: Chambers.

Akamatsu, Tsutomu (1990). Vowels in unaccented syllables in current British English. *La Linguistique: Revue de la Societe Internationale de Linguistique Fonctionnelle*, Vol. 26, No. 1, pp. 89-98.

Bailey, Charles-James N. (1978). Four low-level pronunciation rules of Northern States English. *Journal of the International Phonetic Association*, Vol. 8, pp. 24-33.

Bernstein, Cynthia (1993). Measuring social causes of phonological variation in Texas. *American Speech*, Vol. 68, pp. 227-40.

Bauer, Laurie (1984). Linking /r/ in RP: some facts. *Journal of the International Phonetic Association*, Vol. 14, No. 2, pp. 74-9.

Bird, Steve, and Liberman, Mark (1998). Towards a formal framework for linguistic annotations. *Proceedings: ICSLP 98.*

British National Corpus frequencies: ftp.itri.bton.ac.uk/pub/bnc/

Brown, Adam (1988). Linking, intrusive and rhotic /r/ in pronunciation models. *Journal of the International Phonetic Association*, Vol. 18, pp. 144-51.

(1990). *Chambers English Dictionary*. Edinburgh: Chambers.

Coates, Richard (1999). *Word Structure*. London: Routledge.

Collins and Mees, Inger M. (1990). In: Nikolas Coupland (ed.), *English in Wales: diversity, conflict and change*, p. 167-94. Clevedon: Multilingual Matters.

Connolly, John H. (1981). On the segmental phonology of a south Welsh accent of English. *Journal of the International Phonetic Association*, Vol. 11, No. 2, pp. 51-61.

Coupland, Nikolas (ed.) (1990). *English in Wales: diversity, conflict and change.* Clevedon: Multilingual Matters.

Emerson, Ralph H. (1993). The distribution of eighteenth-century prerhotic o- phonemes in Walker's critical pronouncing dictionary. *American Speech*, Vol. 68, No. 2, pp. 115-38.

Fitt, Susan, and Isard, Stephen (1998). Representing the environments for phonological processes in an accent-independent lexicon for synthesis of English. *Proceedings: ICSLP 98*.

Fitt, Susan (1999). The treatment of vowels preceding 'r' in a keyword lexicon of English. *Proceedings: ICPhS 99*.

Fitt, Susan (2001). Using real words for recording diphones. *Proceedings: Eurospeech 2001*. Aalborg.

Fitt, Susan (2001). Morphological approaches for an English pronunciation lexicon. *Proceedings: Eurospeech 2001*. Aalborg.

Fitt, Susan, and Isard, Stephen (1999). Synthesis of regional English using a keyword lexicon. *Proceedings: Eurospeech 9*9. Vol. 2, pp. 823-6.

Foulkes, Paul, And Docherty, Gerard (1999). Urban voices: *Accent studies in the British Isles*. London: Arnold.

Fox, Anthony (1978). To 'r' is human? Intrusive remarks on a recent controversy. *Journal of the International Phonetic Association*, Vol. 8, pp. 72-4.

Fudge, Erik C. (1977). Long and short [æ] in one Southern British speaker's English. *Journal of the International Phonetic Association*, Vol. 7, pp. 55-65.

Giegerich, Heinz (1994). Base-driven stratification: morphological causes and phonological effects of 'strict-cyclicity'. In: R. Wiese (ed.), *Recent developments in lexical phonology*, pp. 31-61. Dusseldorf: Heinrich Heine Universitat.

Gordon, Elizabeth, and Maclagan, Margaret A. (1989). Beer and bear, cheer and chair: a longitudinal study of the ear/air contrast in New Zealand English. *Australian Journal of Linguistics*, Vol. 9, No. 2, pp. 203-20.

Hargus, Sharon (1989). Mohanan: The theory of lexical phonology. *Language*, Vol. 65, No. 1, p. 164.

Harris, John (1985). The lexicon in phonological variation. In: John Harris, David Little and David Singleton (eds), *Perspectives on the English Language in Ireland*. Proceedings of the First Symposium on Hiberno-English held at Trinity College Dublin 16-17 September 1985, pp. 187-208.

Hawkins, Peter (1976). The role of New Zealand English in a binary feature analysis of English short vowels. *Journal of the International Phonetic Association*, Vol. 6, pp. 50-66.

Holmes, Janet (1994). New Zealand flappers: an analysis of t voicing in New Zealand English. *English World-Wide*, Vol. 15, No. 2, pp. 195-224.

Hughes, Arthur, and Trudgill, Peter (1979). *English accents and dialects.* London: Edward Arnold.

Johnston, Paul (1983). Irregular style variation patterns in Edinburgh speech. *Scottish Language*, Vol. 2, pp. 1-19.

Johnston, Paul A. (1985). The rise and fall of the Morningside/Kelvinside accent. In: Manfred Gorlach (ed.), *Focus on Scotland*, pp. 37-56. Varieties of English around the world, Vol. 5. Amsterdam: Benjamins.

Kaisse, Ellen M., and Shaw, Patricia A. (1985). On the theory of Lexical Phonology. *Phonology Yearbook*, Vol. 2, pp. 1-30.

Kaisse, Ellen M., and Hargus, Sharon (1994). When do linked structures evade structure preservation? In: Richard Wiese (ed.), *Recent developments in Lexical Phonology*, pp. 185-204.

Kenyon, John S., and Knott, Thomas A. (1953). *A pronouncing dictionary of American English.* Springfield: Merriam-Webster.

Kiparsky, Paul (1985). Some consequences of Lexical Phonology. *Phonology Yearbook*, Vol. 2, pp. 85-138.

Kreidler, Charles W. (1989). *The pronunciation of English - a course book in phonology.* Oxford: Blackwell.

Lau, Raymond, and Seneff, Stephanie (1998). A unified framework for sublexical and linguistic modelling supporting flexible vocabulary speech understanding. *Proceedings: ICSLP 1998.*

Lewis, J. Windsor (1975). Linking /r/ in the General British pronunciation of English. *Journal of the International Phonetic Association*, Vol. 5, pp. 37-42.

Lewis, J. Windsor (1977). The r-link business – a reply. *Journal of the International Phonetic Association*, Vol. 7, pp. 28-31.

Lindsey, Geoff (1990). Quantity and quality in British and American vowel systems. In: Susan Ramsaran (ed.), *Studies in the pronunciation of English: a commemorative volume in honour of A.C. Gimson*, pp. 106-18. London: Routledge.

Macaulay, Ronald (1985). The narrative skills of a Scottish coal miner. In: Manfred Gorlach (ed.), *Focus on Scotland* pp. 101-24. Varieties of English around the world, Vol. 5. Amsterdam: Benjamins.

Maclagan, Margaret A., and Gordon, Elizabeth (1996). Out of the AIR and into the EAR: another view of the New Zealand diphthong merger. *Language Variation and Change*, Vol. 8, No. 1, pp. 125-47.

Maptask frequencies: http://www.cogsci.ed.ac.uk/elsnet/Resources/Map-Task/mt_corpus.html

Marcus, Mitchell, Santorini, Beatrice, and Marcinkiewicz, Mary Ann (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, Vol. 19, No. 2, pp. 313-30.

McClure, J. Derric (1977). Vowel duration in aScottish accent. *Journal of the International Phonetic Association*, Vol. 7, pp. 10-16.

McDavid, Raven I. Jr (1979). *Dialects in culture: essays in general dialectology.* University of Alabama Press.

McMahon, April M.S. (1992). Underspecification theory and the analysis of dialect differences in lexical phonology. Transactions of the Philological Society, Vol. 90, pp. 81-119.

Mees, Inger M. (1990). Patterns of sociophonetic variation in the speech ofCardiff schoolchildren. In: Nikolas Coupland (ed.), *English in Wales: diversity, conflict and change*, p. 167-94. Clevedon: Multilingual Matters.

Milroy, James, Milroy, Lesley, and Hartley, Sue (1994). Local and supra-local change in British English. The case of glottalisation. *English World-Wide*, Vol. 15, No. 1, pp. 1-33.

Mohanan, Karuvannur Puthanveettil (1982). *Lexical Phonology*. Dordrecht: Reidel.

Mohanan, Karuvannur Puthanveettil (1985). Syllable structure and lexical strata in English. *Phonology Yearbook*, pp. 139-55.

Ó Sé, Diarmuid (1985). Word-stress in Hiberno-English. In: John Harris, David Little and David Singleton (eds), *Perspectives on the English Language in Ireland*. Proceedings of the First Symposium on Hiberno-English held at Trinity College Dublin 16-17 September 1985, pp. 97-107.

Pointon, G.E. (ed.) (1990). *BBC pronouncing dictionary of British names*. Oxford: Oxford University Press.

Pring, J.T. (1976). More thoughts on the *r*-link business. *Journal of the International Phonetic Association*, Vol. 7, pp. 28-31.

Reid, Euan (1978). Social and stylistic variation in the speech of children: some evidence from Edinburgh. In: Peter Trudgill (ed.), *Sociolinguistic patterns in British English*, pp. 158-71. London: Edward Arnold.

Romaine, Suzanne (1978). Postvocalic /r/ in Scottish English: sound change in progress? In: Peter Trudgill (ed.), *Sociolinguistic patterns in British English*, pp. 144-57. London: Edward Arnold.

Rubach, Jerzy (1985). Lexical phonology: lexical and postlexical derivations. *Phonology Yearbook*, pp. 157-72.

Scobbie, James M., Hewlett, Nigel, and Turk, Alice E. (1999). Standard English in Edinburgh and Glasgow: the Scottish Vowel Length Rule revealed. In: Paul Foulkes and Gerard Docherty (eds), *Urban Voices: Accent studies in the British Isles*. London: Arnold.

Seneff, Stephanie, Lau, Raymond, and Meng, Helen (1996). ANGIE: a new framework for speech analysis based on morpho-phonological modelling. *Proceedings: ICSLP 1996*, pp. 110-3.

Seneff, Stephanie (1998). The use of linguistic hierarchies in speech understanding. *Proceedings: ICSLP 1998.*

Tench, Paul (1990). The pronunciation of English in Abercrave. In: Nikolas Coupland (ed.), *English in Wales: diversity, conflict and chan*ge, pp. 130-41. Clevedon: Multilingual Matters.

Trudgill, Peter, and Hannah, Jean (1985). *International English. A guide to varieties of Standard English.* 2nd edn.

Trudgill, Peter, Gordon, Elizabeth, and Lewis, Gillian (1998). New-dialect formation and Southern Hemisphere English: the New Zealand short front vowels. *Journal of Sociolinguistics*, Vol. 2, pp. 35-51.

Unicode home pages: http://www.unicode.org

Wall, Larry, Orwant, Jon, and Christiansen, Tom (2000). *Programming Perl (3rd Edition).* O'Reilly and Associates, Inc.

Wells, John C. (1982). *Accents of English.* Cambridge: Cambridge University Press.

Wells, John C. (2000). Computer-coding the IPA: a proposed extension of SAMPA. http://www.phon.ucl.ac.uk/home/sampa/x-sampa.htm

Woods, Nicola J. (1997). The formation and development of New Zealand English: interaction of gender-related variation and linguistic change. *Journal of Sociolinguistics*, Vol. 1, pp. 95-125.

Alan Wood's Unicode resources: http://www.hclrss.demon.co.uk/unicode/index.html

Appendix II: Parts of Speech

	Tag	Categories	Examples
Nouns, pronouns	NN	common noun	horse, (the only) one
	NNS	common noun, plural	horses
	NNP	proper noun, singular	Fred
	NNPS	proper noun, plural	Himalayas
	PRP	personal pronoun, nominal possessive pronoun	I, mine, yours, (that's) his
	PRP$	possessive pronoun (adjectival possessive forms)	my, your, his (book)
	WP$	possessive wh-pronoun	whose
	WP	wh-pronoun	what, who, whom
Verbs	VB	verb, base form	be, do
	VBP	present verb (not 3ps)	am, are, do
	VBZ	present verb (3ps)	is, does
	VBD	past tense verb	were, did
	VBG	gerund, present participle	being, doing
	VBN	past participle	been, done
	MD	modal verb	can, may, ought
Adjectives, adverbs	JJ	adjective	happy-go-lucky, superior, fourth-largest, first, further (details), four
	JJR	adjective, comparative	more, easier
	JJS	adjective, superlative	most, happiest, worst
	RB	adverb	happily, quite, enough, not, later, most (everything)
	RBR	adverb, comparative	more, (run) further, further (from) closer (to), later (than), harder (than)
	RBS	adverb, superlative	most, furthest (from), closest (to), (try) hardest
	WRB	wh-adverb	how, where, why, when (temporal)
Grammatical functions	DT	article	the, every, no, another, some, either, that, all (roads)
	CC	coordinating conjunction	and, but, yet, or, plus, over (divided by), for
	IN	subordinating conjunction, preposition	when (meaning 'if'), (look) up (to), (look) at
	RP	particle	(bring) up, (badly) off, (break) down
	TO	the preposition 'to'	to
	POS	possessive ending	's, '
	WDT	wh-determiner	which, that (as relative pronoun)
	PDT	predeterminer	all (his marbles), such (a good time)
	EX	existential there	there (was a party)
Miscellaneous	CD	cardinal number	one, one (of), six
	UH	exclamation	my!, yes
	FW	foreign word	bête noire
	LS	list	(letters and numbers in a list)

Appendix III: Symbol tables

The tables illustrate the keysymbols used in the lexicon, and show transformations applied by ***post-lex-rules.pl***.

In the symbol list, derivation list, and regional variants:

green	is used for pre-rule symbols and annotations. Regional variants in pre-rule tables are given as keysymbols; these symbols may be subject to further rules and mergers
black	is used for global symbols. Regional variants of these are given in black IPA.
red	is used for post-rule symbols. Regional variants of these are given in red IPA.
pink	is used for generic symbols, such as v for lower-case vowel or ∅ for deletion
blue	is used for rule numbers, and rule-derived keysymbol transcriptions

The IPA transcriptions are at the level of broad phonetic, and are equivalent to ipa-formatted output from the scripts. As far as possible, symbols have been ordered so that keywords which merge are grouped together. So, for example, the keywords NORTH and FORCE are listed adjacently, so they can be illustrated as a single phone for accents which do not distinguish them (these symbols will be merged by ***map-unique.pl***). Where this is not possible, for instance with MAZDA and PALM, an asterisk shows that the phone is repeated elsewhere in the column. A double asterisk denotes a realisation which is equivalent to a series of phones used elsewhere, for example |aer| in some accents is equivalent to |ae| + |@r|, [aɪ] + [ə]. The asterisks aid an assessment of the distinctions made in each accent. IPA transcriptions for pre-rhotic vowels in rhotic accents are given the diacritic [˞]. This indicates that the vowel is likely to vary in quality from a non-pre-rhotic vowel (c.f., for instance, |@| and |@r|); however, users may find they can collapse these symbols for particular speakers.

KEYWORDS in upper-case are given for global symbols, these keywords define their word-sets. Example words are given in lower-case for pre-rule and post-rule symbols.

Rules and conversions which are applied directly to the keysymbols are shown with each symbol; mappings are not explicitly shown

Example tables: (x and y stand for any keysymbol(s); v stands for a lower case vowel; V for an upper case vowel; vr and VR stand for lower and upper-case pre-rhotic vowels).

	General pre-rule marker	Example	Example transcription	Derived global keysymbols	RP	Leeds	Edinburgh
Green: pre-rule marker Pink: generic symbol	V4	**a**dmit	{ A4 d . m * i t }	V4 → @, v conv_fullinit4	@	v	@
	Symbol or symbol marker	Lower case: example word (pre-/ post-rule symbols); UPPER CASE: keyword (global symbols)	Black: transcription from base lexicon Blue: derived, accent-specific transcription	Black: global keysymbol Green: pre-rule marker; Pink: generic symbol Red: allophone (post-rule keysymbol); Blue: rule number Function names which perform rules are also given	Black: global keysymbol (pre-rule tables) or IPA of global keysymbol shown in first column (global and post-rule tables) Pink: generic symbol Blue: rule number(s) applied in accent; Red: IPA of a post-rule keysymbol - indicates allophone doesn't apply		
					IPA		
	Key-symbol	Keywords	Example transcription	Derivations	RP	Leeds	Edinburgh
Black: global symbol	t	TEA	{ t * ii }	1 t y → ch do_ty_dy 2 t → ? do_glottal_stop	t 1	t 1	t 1,2
Red: post-rule symbol	?	ca**t**	{ k * a ? }	2 t → ? do_glottal_stop	-	-	? 2

Key-symbol	Keywords/ examples	Example transcription	Derivations	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Caro-lina	N. York

Pre-rule symbols
(function names of conversions applying to pre-rule symbols are listed)

VOWELS

Pre-rule key-symbol	Example	Example transcription	Derived global keysymbols, and function names of conversions	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Carolina	N. York
E0	enlist	< E0 n <.{ l * i s t }	E0 → i, e conv_fullinit4	i	e	i	i	i	i	i	@	@	i	i	i
E05	fastest	{ f * ah s t }.> E05 s t >	E05 → i, e conv_fullfinal5		i			e	e						
I2	africa	{ * a . f r I2 . k @ }	I2 → i, @ conv_i_schwa_2					i	i				@	i	@
I6	pirate	{ p * ae . r I6 t }	I6 → i, @ conv_i_schwa_6	@	@	@	@			@					
I7	wallet	{ w * o . l I7 t }	I7 → i, @ conv_i_schwa_7	i	i	i	i								i
iu3	bruise	{ b r * iu3 z }	iu3 → uu, iu conv_iu	uu	uu	uu	uu	iu	iu	uu	uu	uu	uu	uu	uu
ah2	dance	{ d * ah2 n s }	ah2 → ah, a conv_ah2	ah	ah	ah	ah	ah	ah	ah	a	ah	ah	ah	ah
@@r2	milieu	{ m ii . l y * @@r2 }	@@r2 → @@r, uu conv_longschwa	@@r	@@r	uu	uu	@@r	@@r	uu	@@r	@@r	uu	uu	uu
@@r3	goebbels	{ g * @@r3 . b l z }	@@r3 → @@r, ou conv_longschwa			ou	ou			ou			ou	ou	ou

VOWELS/CONSONANTS

Pre-rule key-symbol	Example	Example transcription	Derived global keysymbols, and function names of conversions	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Caro-lina	N. York
ii2	million	{ m * i . l ii2 @ n }	ii2 → y, ii conv_ii2	ii	ii	y	y	y	y	ii	y	y	y	y	y

GENERAL TYPOGRAPHICAL MARKERS

General pre-rule marker	Example	Example transcription	Derived global keysymbols, and function names of conversions	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Caro-lina	N. York
[x]	herb	{ [h] * er r b }	[x] → x, ∅ conv_y, conv_basic	x	x	x	x	x	x	x	x	x	∅	∅	∅
[x1]	suggest	{ s @ [g1] . jh * e s t }	[x1] → ∅, x conv_basic	∅	∅	∅	∅	∅	∅	∅	∅	∅	x	x	x
V	regime	{ r EE . zh * ii m }	V → v, @ conv_basic	v	v	v	v	v	v	v	v	v	@	@	@
VR	mallard	{ m * a . l AR d }	VR → vr, @r conv_basic	vr	vr	vr	vr	vr	vr	vr	vr	vr	@r	@r	@r
V1	anchovy	{ * a n . ch OU1 . v iy }	V1 → @, v conv_basic	@	@	@	@	@	@	@	@	@	@	@	@
VR1	caernarvon	{ k AR1 r . n * ar r . v n }	VR1 → @r, v conv_basic	@r	@r	@r	@r	@r	@r	@r	@r	@r	vr	vr	vr
V4	admit	{ A4 d . m * i t }	V4 → @, v conv_fullinit4	@	v	@	@	@	@	@	@	@	@	@	@

Key-symbol	Keywords	Example transcription	Derivations	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Caro-lina	N. York
V5	actress	{ * a k . t }> r E5 s >	V5 → @, v conv_fullfinal5	@	@	@	@	v	v	@	@	@	@	@	@
[V5]	decadent	{ d * e . k @ . d == [E5] n t }	[V5] → @, v, ∅ conv_fullfinal5								∅	∅	∅	∅	∅
[V50]	level	{ l * e . v [E50] l }	[V50] → ∅, v conv_fullfinal5	∅	∅	∅	∅			∅					
x/y	adolf	{ * a/ee . d o l f }	x/y → x, y conv_basic	x	x	x	x	x	x	x	x	x	y	y	y
(x x/y y)	schedule	{ (sh/s k) * e . d y uu l }	(x x/y y) → x x, y y conv_basic	x x	x x	x x	x x	x x	x x	x x	x x	x x	y y	y y	y y

G lobal and post-rule keysymbols

(function names of rules and conversions applying to global and post-rule symbols are listed)

CONSONANTS

Ascii				IPA											
Key-symbol	Keywords	Example transcription	Derivations, and function names of conversions/rules	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare (1)	General Austra-lian	General New Zealand	General Ameri-can	S. Caro-lina	N. York
p	PEA	{ p * ii }		p	p	p	p	p	p	p	p	p	p	p	p
t	TEA	{ t * ii }	1 t y → ch do_ty_dy 2 t → ? do_glottal_stop 3 t → r do_t_r 4 t,d → t^ do_taps	t 1	t 1,3	t 1,2	t 1,2	t 1,2,4	t	t 1,4	t 1,4	t 1,4	t 1,4	t 1,4	t 1,4
?	cat	{ k * a ?}	2 t → ? do_glottal_stop	-	-	? 2	? 2	? 2	-	-	-	-	-	-	-
t^	butter merry	{ b * uh . t^ @r r } { m * e. t^ iy }	4 t, d → t^ do_taps 5 r → t^ do_tapped_r	-	-	r 5	r 5	r 4,5	-	r 4	r 4	r 4	r 4	r 4	r 4
k	KEY	{ k * ii }		k	k	k	k	k	k	k	k	k	k	k	k
x	LOCH	{ l * o x }				x	x	x	x	x					
b	BEE	{ b * ii }		b	b	b	b	b	b	b	b	b	b	b	b
d	DYE	{ d * ae }	4 t, d → t^ do_taps 6 d y → jh do_ty_dy	d 6	d 6	d 6	d 6	d 6	d	d 6	d 6	d 6	d 4,6	d 4,6	d 4,6
g	GUY	{ g * ae }	7 ng → ng g do_ng_nng	g	g	g	g	g	g	g	g	g	g	g	g 7
ch	EACH	{ * ii ch }	1 t y → ch do_ty_dy	ʧ 1	ʧ 1	ʧ 1	ʧ 1	ʧ 1	ʧ	ʧ 1	ʧ 1	ʧ 1	ʧ 1	ʧ 1	ʧ 1
jh	EDGE	{ * e jh }	6 d y → jh do_ty_dy	ʤ 6	ʤ 6	ʤ 6	ʤ 6	ʤ 6	ʤ	ʤ 6	ʤ 6	ʤ 6	ʤ 6	ʤ 6	ʤ 6

Key-symbol	Keywords/ examples	Example transcription	Derivations	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Caro-lina	N. York
s	**S**EA	{ s * ii }	8 s y → sh do_sy_zy 9 s ii → sh do_sy_zy 10 z → s do_vless_plural	s 8	s 8	s 8	s 8	s 8	s 10	s 8	s 8	s 8	s 8,9	s 8,9	s 8,9
z	**Z**OOM	{ z * uu m }	10 z → s do_vless_plural 11 z y → zh do_sy_zy 12 z ii → zh do_sy_zy	z 11	z 11	z 11	z 11	z 11	z 10	z 11	z 11	z 11	z 11,12	z 11,12	z 11,12
sh	**SH**E	{ sh * ii }	8 s y → sh do_sy_zy 9 s ii → sh do_sy_zy	ʃ 8	ʃ 8	ʃ 8	ʃ 8	ʃ 8	ʃ	ʃ 8	ʃ 8	ʃ 8	ʃ 8,9	ʃ 8,9	ʃ 8,9
zh	BEI**G**E	{ b * ei zh }	11 z y → zh do_sy_zy 12 z ii → zh do_sy_zy	ʒ 11	ʒ 11	ʒ 11	ʒ 11	ʒ 11	ʒ	ʒ 11	ʒ 11	ʒ 11	ʒ 11,12	ʒ 11,12	ʒ 11,12
f	**F**AN	{ f * ah n }		f	f	f	f	f	f	f	f	f	f	f	f
v	VAN	{ v * ah n }		v	v	v	v	v	v	v	v	v	v	v	v
th	**TH**IN	{ th * i n }	13 dh → th do_vless_plural 14 ll → th l conv_ll	θ 14	θ 14	θ 14	θ 14	θ	θ 13	θ 14	θ 14	θ 14	θ 14	θ 14	θ 14
dh	**TH**EN	{ dh * e n }	13 dh → th do_vless_plural	ð	ð	ð	ð	ð	ð 13	ð	ð	ð	ð	ð	ð
h	**H**AT	{ h * a t }	15 h → ∅ do_h_drop	h	- 15	h	h	- 15	- 15	h	- 15	- 15	h	h	h
m	**M**E	{ m * ii }	16 m → m! convert_segments	m 16	m 16	m 16	m 16	m 16	m 16	m 16	m 16	m 16	m 16	m 16	m 16
m!	chas**m**	{ k * a . z m! }	16 m → m! convert_segments	m̩ 16	m̩ 16	m̩ 16	m̩ 16	m̩ 16	m̩ 16	m̩ 16	m̩ 16	m̩ 16	m̩ 16	m̩ 16	m̩ 16
n	**KN**EE	{ n * ii }	17 n → n ! convert_segments 18 ng → n, n! do_ing 19 @ n → n! do_syllab	n 17	n 17,18	n 17,18 19	n 17,18 19	n 17,18 19	n 17,18	n 17,18 19	n 17,18, 19	n 17,19	n 17,19	n 17,19	n 17,19
n!	mis**sion**	{ m * i . sh n! }	17 n → n! convert_segments 18 ng → n, n! do_ing 19 @ n → n! do_syllab	n̩ 17	n̩ 17	n̩ 17,18, 19	n̩ 17,18, 19	n̩ 17,19	n̩ 17	n̩ 17,19	n̩ 17,19	n̩ 17,19	n̩ 17,19	n̩ 17,19	n̩ 17,19
ng	SO**NG**	{ s * au ng }	7 ng → ng g do_ng_nng 18 ng → n, n! do_ing	ŋ	ŋ 18	ŋ 18	ŋ 18	ŋ 18	ŋ 18	ŋ 18	ŋ 18	ŋ 18	ŋ	ŋ	ŋ 7
l	**L**AY	{ l * ei }	14 ll → th l conv_ll 20 l → l! convert_segments 21 l,ll → lw do_dark_l	l 14,20, 21	ɬ 14,20	ɬ 14,20	ɬ 14,20	l 20,21	l 20	l 14,20	ɬ 14,20	ɬ 14,20	l 14,20, 21	l 14,20, 21	l 14,20, 21
ll	**LL**ANDUDNO	{ ll a n . d * i d . n ou }	14 ll → th l conv_ll 21 l,ll → lw do_dark_l					ɬ	ɬ						
lw	fee**l**	{ f * ii lw }	21 l, ll → lw do_dark_l	ɬ 21	-	-	-	ɬ 21	-	-	-	-	ɬ 21	ɬ 21	ɬ 21
l!	catt**le**	{ k * a . t l!}	20 l → l! convert_segments	l̩ 20	l̩ 20	l̩ 20	l̩ 20	l̩ 20	l̩ 20	l̩ 20	l̩ 20	l̩ 20	l̩ 20	l̩ 20	l̩ 20

Key-symbol	Keywords	Example transcription	Derivations	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Caro-lina	N. York
r	RAY	{ r * ei }	3 t → r do_t_r 5 r → t^ do_tapped_r 22 r → ∅ do_non_rhotic	ɹ 22	ɹ 3,22	ɹ 5	ɹ 5	ɹ 5,22	ɹ 22	ɹ	ɹ 22	ɹ 22	ɹ	ɹ 22	ɹ
y	YES	{ y * e s }	1 t y → ch do_ty_dy 6 d y → jh do_ty_dy 8 s y → sh do_sy_zy 11 z y → zh do_sy_zy 23 y (uu, iu, iu3, ur) → iu, iur conv_iu 24 ir → y @@r do_ur_ir 25 @@r → y @@r do_y_insert	j 1,6, 8,11	j 1,6, 8,11	j 1,6, 8,11	j 1,6, 8,11	j 1,6, 8,11 23,24 25	j 23,24 25	j 1,6 8,11	j 1,6, 8,11	j 1,6, 8,11	j 1,6, 8,11	j 1,6, 8,11	j 1,6, 8,11
w	WITCH	{ w * i ch }	26 ou,oow,oou → @, @ w do_sus_weak	w	w	w	w	w	w	w	w	w	w	w 26	w
hw	WHICH	{ hw * i ch }				ʍ	ʍ			ʍ			ʍ	ʍ	

VOWELS

Ascii				IPA											
Key-symbol	Keywords	Example transcription	Derivations, and function names of conversions/rules	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Carolina	N. York
e	DRESS	{ d r * e s }	27 e → a do_ou_l	ɛ	ɛ	ɛ	ɛ	ɛ	ɛ	ɛ	ɛ	ɛ 27	ɛ	ɛ	ɛ
ao	MAZDA	{ m * ao z . d @ }		æ	æ	a	a	a 29	a	a	æ	æ 27	ɑ*	ɑ*	ɑə*
a	TRAP	{ t r * a p }	27 e → a do_ou_l 28 a,ah,oa → eh do_us_eh										æ 28	æ 28	æ 28
ah	BATH	{ b * ah th }	28 a,ah,oa → eh do_us_eh 29 ah → aa do_class	ɑ:							a:	a:			
oa	BANANA	{ b @ . n * oa . n @ }	28 a,ah,oa → eh do_us_eh		a:			a: 29	a:	a:					
aa	PALM	{ p * aa [l1] m }	29 ah → aa do_class										ɑ*	ɑ*	ɑə*
ar	START	{ s t * ar r t }				aˌ	aˌ			aːˌ			ɑˌ	ɑ:	ɑəˌ
eh	ann	{ * eh n }	28 a,ah,oa → eh do_us_eh	-	-	-	-	-	-	-	-	-	æ̝ 28	æɪ 28	ɛə 28
oul	goal	{ g * oul l }	30 ou, ouw → oul do_ou_l	-	ɔʊ 30	-	-	ɤʊə** 30	o:ə** 30	-	ɔʊ 30	ɔʊ 30	-	-	-

Key-symbol	Keywords/ examples	Example transcription	Derivations	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Caro-lina	N. York
ou	G**OA**T	{ g * ou t }	30 ou,ouw → oul do_ou_l 26 ou,oow,oou → @, @ w do_sus_weak	əʊ	oː 30	o	o	ɤʊ 30	oː* 30	oː	ʌʊ 30	ʌʊ 30	oʊ	oʊ 26	oʊ
ouw	KN**OW**	{ n * ouw }	30 ou,ouw → oul do_ou_l 26 ou,oow,oou → @, @ w do_sus_weak						ou 30						
oou	ADI**O**S	{ ~ a . d ii . * oou s }	26 ou,oow,oou → @, @ w do_sus_weak	ɒ	ɒ	ɔ	ɔ	ɒ	ɒ	ɑ	ɒ	ɒ			
o	L**O**T	{ l * o t }											ɑ*	ɑ*	ɑə*
au	CL**O**TH	{ k l * au th }											ɔ	ɒʊ	ɔə
oo	TH**OUGH**T	{ th * oo t }		ɔː	ɔː			ɔː	ɒː	ɔː	ɔː 31	ɔː			
or	N**O**RTH	{ n * or r th }	31 ur → or do_ur_or			ɔˈ	ɔˈ			ɔːˈ			ɔˈ		ɔəˈ
our	F**O**RCE	{ f * our r s }				oˈ	oˈ		oː*	oːˈ				oə	
ii	FL**EE**CE	{ f l * ii s }	9 s ii → sh do_sy_zy 12 z ii → zh do_sy_zy 32 ii → ii; do_scots_long 33 ii,iy → ie do_short_ii 34 ii → iy do_short_ii 35 ir → ii @r do_ur_ir	iː 34	iː 33,35	i 32	i 32	iː 34	iː 34	iː 34	ɪiː 34,35	iː 34,35	i 9,12	i 9,12	i 9,12
iy	HAPP**Y**	{ h * a . p iy }	33 ii,iy → ie do_short_ii 34 ii → iy do_short_ii	i 34	ɪ 33			i 34	i 34	i 34	ɪiː 34	i 34		ɪ	
i	K**I**T	{ k * i t }	36 i → @ do_i_reduction	ɪ		ɪ	ɪ	ɪ	ɪ	ɪ	ɪ 36	ə*	ɪ		ɪ
ie	harr**ie**t	{ h * a . r ie @ t }	33 ii,iy → ie do_short_ii	-	i 33	-	-	-	-	-	-	-	-	-	-
ii;	agr**ee**d	{ @ . g r * ii; } d	32 ii → ii; do_scots_long	-	-	iː 32	iː 32	-	-	-	-	-	-	-	-

Key-symbol	Keywords	Example transcription	Derivations	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Caro-lina	N. York
@r	LETTER	{ l * e . t @r r }	35 ir → ii @r do_ur_ir 37 iur → iu @r do_ur_ir 38 ur → uu @r do_ur_ir	ə	ə 35,38	əˈ	əˈ	ə 19,37	ʌ 37	əˈ	ə 19,35, 36	əˑ 19,35	əˈ	ə 19,26	əˈ
@	COMMA	{ k * o . m @ }	19 @ n → n! do_syllab 26 ou,oow,oou → @, @ w do_sus_weak 36 i → @ do_i_reduction			ə 19	ə 19			ə 19			ə 19		ə 19
uh	STRUT	{ s t r * uh t }	39 uh → @@r do_hurry_furry	ʌ	ʊ	ʌ	ʌ			ʌ	ʌ	ʌ	ʌ 39	ʌ 39	ʌ 39
u	FOOT	{ f * u t }	32 uu,u,iu → uu; do_scots_long	ʊ		ʉ 32	ʉ 32	ɤ	ʊ	ʊ	ʊ	ʊ	ʊ	ʊ	ʊ
uu	GOOSE	{ g * uu s }	23 y (uu, iu, iu3, ur) → iu, iur conv_iu 32 uu,u,iu → uu; do_scots_long 38 ur → uu @r do_ur_ir 40 uu,iu → uw do_uw 41 uu → uul do_ou_l	uː 40	uː 38,40, 41			uː 23,40, 41	uː 23,40, 41	uː 40	ʊʉ 40	ʉː 38,40 41	u	u 40	u
iu	BLEW	{ b l * iu }	23 y (uu, iu, iu3, ur) → iu, iur conv_iu 32 uu, u, iu → uu; do_scots_long 37 iur → iu @r do_ur_ir 40 uu,iu → uw do_uw					ɪu* 23,37 40	ɪu* 23,37 40						ɪu
uu;	brewed	{ b r * uu; }> d >	32 uu, u, iu → uu; do_scots_long	-	-	ʉː 32	ʉː 32	-	-	-	-	-	-	-	-
uw	louise	{ l uw . * ii z }	40 uu,iu → uw do_uw	u 40	u 40	-	-	u 40	u 40	u 40	u 40	u 40	-	-	-
uul	ghoul	{ g * uul l }	41 uu → uul do_ou_l	-	uʊ 41	-	-	uːə ** 41	uːʌ** 41	-	-	uː 41	-	-	-
ei	WAIST	{ w * ei s t }		ɛɪ	eː	e	e	eɪ	eɪ	eː	ʌi	ʌi	e	eɪ	eɪ
ee	WASTE	{ w * ee s t }							eː						
ai	PRICE	{ p r * ai s }	42 ai, ae → aai do_aai_oow	aɪ	aɛ	ʌi	ʌi	əi	ai	aɪ	ɒɪ	ɑɪ	aɪ	aɪ 42	ɑɪ
ae	TIED	{ t * ae }> d >	42 ai, ae → aai do_aai_oow			ae	ae								
aer	FIRE	{ f * aer r }		aɪə**	aɛə**	aeˈ	aeˈ	əijə**	aijə**	aɪˈ	ɒɪə**	ɑɪə**	aɪˈ	ɑɪə	ɑɪˈ
aai	time	{ t * aai m }	42 ai, ae → aai do_aai_oow	-	-	-	-	-	-	-	-	-	-	ɑːɪ 42	-
oi	CHOICE	{ ch * oi s }		ɔɪ	ɔɪ	ɔɪ	ɔɪ	ɔi	ɒi	ɔɪ	ɔɪ	ɔɪ	ɔɪ	ɔɪ	ɔɪ
oir	COIR	{ k * oir r }		ɔɪə**	ɔɪə**	ɔɪˈ	ɔɪˈ	ɔijə**	ɒijə**	ɔɪˈ	ɔɪə**	ɔɪə**	ɔɪˈ	ɔɪə**	ɔɪˈ
ow	MOUTH	{ m * ow th }	43 ow → oow do_aai_oow	aʊ	aʊ	ʌʊ	ʌʊ	əu	au	aʊ	æo	æʊ	aʊ	æʊ 43	aʊ
owr	HOUR	{ * owr r }		aʊə**	aʊə**	ʌʊˈ	ʌʊˈ	əuwə**	əuwə**	aʊˈ	æoə**	æʊə**	aʊˈ	æʊə**	aʊˈ

Key-symbol	Keywords/ examples	Example transcription	Derivations	RP	Leeds	Edin-burgh	Aber-deen (1)	Cardiff	Aber-crave	County Clare	General Austra-lian	General New Zealand	General Ameri-can	S. Caro-lina	N. York
oow	l**ou**d	{ l * oow d }	ow → oow do_aai_oow	-	-	-	-	-	-	-	-	-	-	ɑːʊ	-
i@	ID**EA**	{ ai . d * i@ }		ɪə	iːə**	iə**	iə**	iːə**	iːə**	iːə**	ɪiːə**	iːə**	iə**	iə**	iə**
ir	N**EA**RING	{ n * ir r }.> i ng >	24 ir → y @@r, @@r do_ur_ir 32 ir → ir; do_scots_long 35 ir → ii @r do_ur_ir 44 ir → iir do_ur_ir 45 eir → ir conv_eir_ir		i 35	iˈ 32	iˈ 32	i 24	i 24	iˈ	ɪː 35,44	iˈː 35,45	ɪˈ	iˈ 24	ɪˈ
ir;	n**ea**r	{ n * ir; r }	32 ir → ir; do_scots_long	-	-	iːˈ 32	iːˈ 32	-	-	-	-	-	-	-	-
iir	b**ea**rd	{ b * iir r d }	44 ir → iir do_ur_ir	-	-	-	-	-	-	-	ɪə 44	-	-	-	-
@@r	N**UR**SE	{ n * @@r r s }	24 ir → y @@r, @@r do_ur_ir 25 @@r → y @@r do_y_insert 39 uh → @@r do_hurry_furry	ɜː	ɜː	ɜˈ	ɜˈ	øː 24,25	ɜː 24,25	ɜːˈ	ɜː	œː	ɜˈ 39	ɜ 39	ɜˈ 39
er	P**E**RT	{ p * er r t }				ɛˈ	ɛˈ								
eir	SQUA**R**ING	{ s k w * eir r }.> i ng >	45 eir → ir conv_eir_ir	ɛə	ɛː	eˈ	eˈ	ɛː	ɛː	eːˈ	ɛː	iˈː* 45	ɛˈ	ɛə	ɛəˈ
ur	J**U**RY	{ j * ur . r iy }	23 y (uu, iu, iu3, ur) → iu, iur conv_iu 31 ur → or do_ur_or 32 ur → ur; do_scots_long 38 ur → uu @r do_ur_ir	ʊə	ʊə 38	ʉˈ 32	ʉˈ 32	ʊə 23	ʊə 23	uːˈ	uː 31	uː 38	ʊˈ	ʊə 31	ʊəˈ
ur;	c**u**re	{ k y * ur; r }	32 ur → ur; do_scots_long	-	-	ʉːˈ 32	ʉːˈ 32	-	-	-	-	-	-	-	-
iur	c**u**rious	{ k * iur . r ii @ s }	23 y (uu, iu, iu3, ur) → iu, iur conv_iu 37 iur → iu @r do_ur_ir	-	-	-	-	ɪu* 23,37	ɪu* 23,37	-	-	-	-	-	-

STRESS, SYLLABLE AND MORPHOLOGICAL MARKERS

Global Keysymbol	Description	IPA (where appropriate]	Example word	Example keysymbol transcription
*	Primary stress	ˈ	newtown	{ n [y] * iu }.{ t – ow n }
~	Secondary stress	ˌ	beefburger	{ b * ii f }.{ b ~ @@r r . g == @r r }
•	Tertiary stress		newtown	{ n [y] * iu }.{ t – ow n }
.	Syllable boundary	.	acorn	{ * ee . k or r n }
<	Prefix boundary		debug	< d ii <.{ b * uh g }
>	Suffix boundary		bugged	{ b * uh g }> d >
{	Start of free root		bug	{ b * uh g }
}	End of free root		bug	{ b * uh g }
$	Root with altered pronunciation		election	{ i . l * e k . sh $}> n >
==	Boundary at junction of two bound morphemes		sonogram	{ s * o . n @ =.= g r ~ a m }
#	Phrase boundary in connected text (doesn't occur within lexicon)		fast food	#{ f * ah s t }#.#{ f * uu d }#
#.#	Boundary at junction of two lexical entries (doesn't occur within lexicon)		fast food	#{ f * ah s t }#.#{ f * uu d }#