Unification and Transduction in Computational Phonology

Julie CARSON

Universität Bielefeld
Fakultät für Linguistik und Literaturwissenschaft
4800 Bielefeld
West Germany

Abstract

In this paper unification and transduction mechanisms are applied in a new approach to phonological parsing. It is shown that unification in the sense of Kay as used in unification grammars, and transduction, a process deriving from automata theory, are both valuable tools for use in computational phonology. By way of illustration, a brief outline of the allophonic parser described by Church is given. Then a linear unification parser for English syllables is introduced. This parser takes phonetic input in the form of feature bundles and uses phonological rules represented by networks of transduction relations together with unification, and an iterative finite-state process to produce phonemic output with marked syllable boundaries. A fundamental distinction is made between two domains: the representations at the phonetic and phonological levels, and the processing of these representations. On this basis, a distinction is made between networks of transduction relations (e.g. between allophones and phonemes), and a set of possible processors (i.e. parsers and transducers) for the interpretation of such networks.

1. Transduction and Unification in Phonology

The proposal to use finite-state transducers in morphology and phonology has been advocated in recent years by Kaplan and Kay /1981/, Koskenniemi /1983/ and others. It has been suggested /Gibbon 1987/ that finite-state transducers are the most appropriate devices for use in other areas of computational phonology. In Koskenniemi's system, single finite-state transducers act as parallel filters in the analysis of Finnish morphology. However, in his morphophonological analysis Koskenniemi has been criticised for using monadic segments rather than the feature bundles which play such an important role in phonology /Gazdar 1985:601/. In the proposal presented below, segments regarded as feature bundles are essential components in the model. The question as to whether it is better to represent the phonological rules as a cascade of transducers or to incorporate them into a single transducer will not be considered here. Kaplan and Kay /1981/ have already put foward a method of compiling the series of transducers into a single transducer (described by Kay /1982/)). Below, for discussion purposes, a single transducer is assumed.

Furthermore I would like to stress that on the phonological level I will discuss network representations of phonotactic and allophonic constraints. The transitions in these networks consist of transduction relations. In the process domain a finite-state transducer will be used to interpret the networks. This is a distinction which is not always made but is beneficial for abstracting the attributes of the model from the processing of the model. Below more emphasis will be placed on the representation domain as it is this which is most interesting for discussion purposes. The actual implementation of the processing domain as a program is regarded, theoretically, as a secondary but by no means a minor issue.

Unification is a concept which has become common in linguistics in recent years due to the important role it plays in current syntactic theories such as FUG, LFG and GPSG. However, it has not as yet played an explicit part in phonological analysis. Below I propose that, by employing elementary unification mechanisms, assimilation and dissimilation can be dealt with in a most satisfactory way. The unification used in this connection is based on the functional description unification described by Kay /1984/.

Here I will give an informal definition of unification based on contradiction and set union and in terms of feature bundles, since this is the representation which will be used below. Two feature bundles composed of attribute-value pairs may be said to unify if for each attribute in their union there does not exist an attribute of the same name with a contradicting value. Where a variable, say X, is found in place of a value in one feature bundle, this variable will be assigned permanently the value from the corresponding attribute-value pair in the other bundle if this exists. This definition of unification, and its implementation, differs from Prolog term unification.

2. Allophone-Phoneme Transduction

In the proposal presented here, segments regarded as feature bundles are essential components. The feature bundles used in this model are sets of attribute-value pairs in line with traditional distinctive feature terminology. The features are not complex and are generally based on those of Chomsky and Halle /1968/. A fully specified feature bundle contains all the features, together with their values, needed to describe one particular sound. Where a phonetic symbol occurs in the text below this is merely an abbreviation convention for a fully specified feature bundle. Rather than being fully specified, a feature bundle may be underspecified. That is to say, only those features appear in the feature bundle which are necessary to describe a class of sounds which participate in a particular phonetic process. For example, the underspecified feature bundle {[+ voc], [- cons]} describes all vowels. The feature bundles are generalisations for sets of input symbols, and resemble the classification in terms of N and V features found in syntax which allows generalisation over categories. They are thus termed C-features (for Category-features).

In Church /1983/ the claim is made that allophonic cues can be extremely useful in phonological parsing. Selkirk /1982/ also maintains that investigation of allophonic variation may be advantageous for syllable analysis since the realisation of particular allophones of a language is strongly dependent on their position within the syllable. Thus in order to take advantage of allophonic cues a distinction must be made between variant and invariant features. Variant features, such as [+ aspiration], occur when discussing allophones of /p/ for example. Thus underspecified feature bundles also contain variant features in order for us to incorporate allophonic information into our classification.

Using variant and invariant features, following Church /1983/, the aim is, given phonetic input in the form of fully specified feature bundles, discard allophonic information (variant features) and produce phonemic output also in feature bundle form with syllable boundaries marked. Church's /1983/ system has a number of stages from *phonetic* input to the point where *phonemic* output is matched with a syllable dictionary. A *phonetic* feature lattice incorporating generalisations about allophones is input to a bottom-up chart parser. This chart parser, which works on a similar basis to the CYK algorithm, provides the *phonetic* input with a syllable structure. A canonicaliser then discards the allophonic information and outputs a *phonemic* feature lattice preserving the syllable structure. It is this structure which then comprises the input to the lexical matcher.

Taking a closer look at the canonicaliser the first thing which springs to mind is a simple transduction process, that is to say, a translation from phones to phonemes. The chart parser has the task of providing syllable structure using phonotactic and allophonic constraints. However, the question here is, are two separate procedures, namely parsing and canonicalisation, really necessary or can they be incorporated into a single process? Below I will sketch a proposal which, with the help of a finite-state transducer, does just this.

3. Phonotactic Nets

Let us first consider the representation level. Following the *on-line feature specification recogniser for English syllables* presented in Gibbon /1985/ a syllable template was constructed as a discrimination network on the basis of phonotactic rules, thus working on the principle of "allowable" combinations of phonemes rather than limiting acceptable strings to those clusters which actually occur. Syllables are not discussed explicitly in terms of onset, peak and coda in this model. Rather these sub-structures and the phonotactic and allophonic rules which depend on them are implicit in the network. The structures, however, can be derived immediately from the topology of the network as represented in a transition diagram. This network is referred to as a phonotactic net. Allophonic constraints were then introduced as part of the input specifications.

Each transition in the phonotactic net models a phonemic segment. The advantage of the feature bundle representation is that segments can be viewed in terms of natural classes, which simplifies the network considerably. The transition labels of the network consist of a pair of feature bundles each containing C-features. One of these bundles represents input specifications and the other output specifications; both are in general underspecified. For example, the bundle of C-features which describes the voiceless plosive consonants is {[- cont], [- voice], [- son], [- strid]}. However, where we need to describe the aspirated allophones of the voiceless plosives the variant feature [+ asp] must be added: {[- cont],

[− voice], [− son], [− strid], [+ asp]}. Therefore when a particular transition in the network is responsible for removing this allophonic information the input transition specification is {[− cont], [− voice], [− son], [− strid], [+ asp]}, and the output transition specification is {[− cont], [− voice], [− son], [− strid]} (see Fig.1). When this phonotactic net is interpreted by a particular parser the phonetic input is generally a string of fully specified feature bundles and in order to use the output for recognition purposes the phonemic output will also be fully specified. It is here that unification plays an important role.

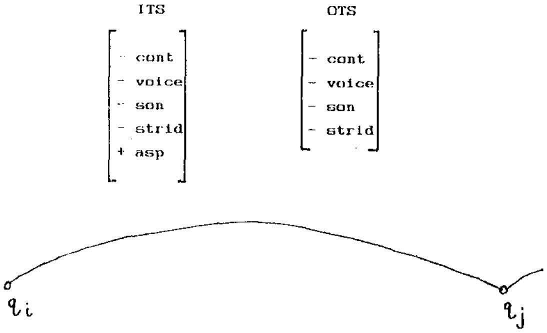

Fig. 1.
Transition accepting voiceless aspirated plosives

When attempting to traverse the network the fully specified input feature bundle must unify with the input transition specification (in terms of C-features) of the current transition. If unification succeeds, the fully specified output bundle must contain the output transition specifications together with all those features from the fully specified input bundle not contained in the input transition specification. In set theoretic terms, let us call the fully specified input feature bundle InFB, the input and output transition specifications ITS and OTS respectively; if unification of InFB with ITS succeeds, the fully specified output bundle OutFB is OTS $\cup$ (InFB / ITS).

The phonetic input feature bundles may be also underspecified however. This allows for circumstances where the values of some features may not be known or indeed the features themselves may not be recognisable. This facility is advantageous for working with feature detectors at the front end as it is still possible to analyse what is known. This, of course, leads to underspecified output which may be used in connection with a lexicon for recognition hypothesising. In such cases the underspecified output, although representing classes of phonemes in the various positions, will only allow those combinations of such classes which actually exist, thus limiting possibilities available for hypothesis. Thus it is not necessary to check the lexicon for forms which according to the rules of the language cannot exist.

4. Constraining Principles

Church discusses a number of factors, most of which date back to work by Morris Halle and are discussed by Chomsky and Halle /1968/, which must be taken into consideration when designing the model /1983:128/ − length, idiosyncratic systematic gaps, voicing assimilation, place assimilation and dissimilation, sonority. These can all be incorporated very easily into the network. The fact that languages restrict sound combinations (idiosyncratic gaps) and the length of initial/final consonant clusters is in any case the basis on which this network is constructed. Decreasing sonority from the nucleus of the syllable towards the margins would seem to be a matter of having [son] as a C-feature and adjusting the value at the appropriate transition.

With regard to phonotactic constraints, the C-features on the transition labels may have variable values. In other words we may cater for the fact that an initial /s/ in English may not be followed by voiced plosives by having as input specifications for one of its following transitions the C-features {[− voc], [α cont], [α voice], [α son], [− strid]} (see Fig.2). α here must have the same value in the three cases, this value being assigned during unification. Unification would fail in this case for voiced plosives as they would be specified for the features {[− voc], [−cont], [+ voice], [− son], [− strid]}. A further convention is introduced, namely that once a feature has been specified on a particular transition it remains until it is explicitly altered on a subsequent transition. In this way vowel harmony may be incorporated into such a network whereby the vowel specifications would remain for subsequent transitions since they would not be relevant for intervening consonants.

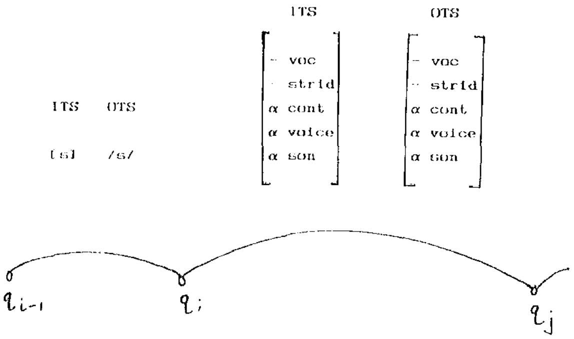

Fig. 2

' Initial /s/ may not be followed by voiced plosives in English. ([s] and /s/ are abbreviations for fully specified feature bundles)

It should be clear also that feature bundle representation together with unification is an elegant way of dealing with assimilation, dissimilation and neutralisation. Assimilation and dissimilation are dealt with by Chomsky and Halle /1968/ in terms of variables as feature coefficients and it is this method which has been incorporated into the network here. So for example, in cases of voice assimilation, the feature [voice] may be checked using a variable, say [α voice]. Therefore, where the particular input segment has the feature [+ voice], unification assigns the value + to the undefined variable α permanently, and similarly in the case of a negative value. This newly found value together with the attribute will then be a C-feature in the input specification for the following transition unless explicitly changed on that transition. This is a type of feature-passing technique similar to that employed in unification-based syntactic theories, but essentially simpler, since it is non-recursive.

Transition weighting is also very important in this model. Selkirk /1980/ emphasises that it is all very well to cater for collocational restrictions but other constraining principles such as maximising onsets should also be incorporated into a syllable parser. Thus transitions are weighted in such a way that the most preferred path out of the network is sought. 'Early closure' /Kimball 1973/ for example, which seeks the shortest path out of the network, is equivalent to the maximal onset principle. Stress resyllabification is similarly dealt with using weighting. Thus, such constraints are incorporated into the network in a simple and principled fashion.

5. Syllable Parsing

Up to now we have been discussing the representation level, namely the phonotactic net envisaged as a syllable template. The phonotactic net in this case was for English but it should be clear that this representation may be used for other languages, dialects or codes. Since the phonotactic net is a network of transduction relations between allophone and phoneme it should be a useful tool for both speech analysis and synthesis. It is important to note at this stage however, that on the processing level we are not restricted to what parsing algorithm we employ. The phonotactic net may be interpreted by any one of a number of parsing procedures. The strategy employed (i.e. depth-first, breadth-first, best-first, lookahead etc.) is also totally independent of the representation.

In the model described here the aim was to use the simplest formalism possible. Thus the parsing and translation processes are undertaken by a depth-first nondeterministic finite-state transducer. That is to say, the phonotactic nets of transduction relations are interpreted by a finite-state machine. Given the phonetic input in the form of feature bundles, the transducer moves from state to state in line with the unification procedure described in section 3 above.

Every time the transducer reaches its final state a "possible" syllable has been found. Therefore,in order to find more than one syllable the transducer iterates so that phonological units and syllable boundaries are output until the input string is empty. Thus we have a single iterative finite-state process. The parsing and canonicalisation processes referred to in section 2 above are incorporated into a single procedure. What is interesting to note in this connection is that since the parsing procedure is nondeterministic in fact all "possible" syllables from the beginning of the input are checked internally (i.e. in the intermediate stages before producing output). Thus the notion of a "possible" syllable of English is catered for.

From a psychological viewpoint it is an interesting fact that only the "possible" syllables are considered. This would also be the case in human processing of neologisms whereby no attempt would be made to form a syllable with an impossible initial/final consonant cluster combination: humans can accept words which conform to the rules of their language even if the words do not actually exist. Thus, with this model we can distinguish between "possible" and "actual" words. If we tested Carroll's Jabberwocky using this model we would get a correct syllable structure. As already noted, the lexicon filters out actual words.

6. Conclusion

The implementation of this model does not claim to be a speech recognition system as it stands but is rather an attempt to deal with a small component of such in a new, elegant and theorectically satisfying way. Unification and transduction can be seen to be useful mechanisms in syllable parsing. Unification provides underspecification-manipulation and feature-passing facilities and transduction provides a translation facility between allophones and phonemes. Transduction relations interpreted by a finite-state transducer have the further advantage of bidirectionalty. That is to say, one can translate from allophones to phonemes or vice versa (perhaps with some ambiguity in the phoneme-allophone direction). This system, however, should be a useful tool in both speech synthesis and speech analysis.

An extension of this notion of a syllable parser is to talk in terms of phonological words, whereby at the representation level the network would consist of two sub-nets catering for reduced and unreduced syllables respectively. A further extension is to use a tree-structured lexicon could be employed in a similar way to that proposed by Kay /1982/ to distinguish *actual* words from *possible* words. Representing the lexicon as a discrimination net and in terms of distinctive feature bundles makes it possible to deal with various parts of a recognition system in a uniform way. The movement of the transducer may then be directed by using the tree-lexicon in parallel (see Fig.3). In cases where the input segment is underspecified hypotheses could be made immediately as to the values of particular features thus excluding paths which will eventually lead to impossible sequences hence increasing the efficiency of the parser.

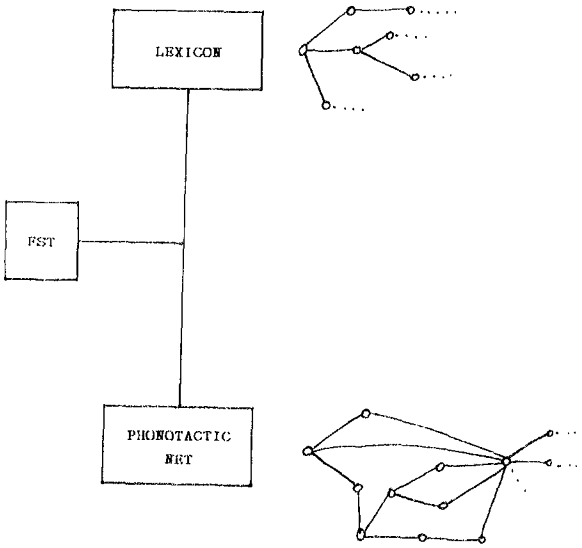

Fig.3

The model has been implemented in C-Prolog on a Hewlett Packard 9000.

References

Chomsky, N. , M. Halle 1968. *The Sound Pattern of English.* Harper and Row, New York

Church, K. 1983. *Phrase Structure Parsing. A method for taking advantage of allophonic constraints.* Ph.D. Thesis, MIT. Published Indiana University Linguistics Club

Gazdar, G. 1985. "Review article: Finite state morphology". In: *Linguistics* 23:597-607

Gibbon, D. 1985. "Prosodic Parsing in English and Akan". Paper held at 21st International Conference on Contrastive Linguistics, Błazejewko

Gibbon, D. 1987. "Finite State Processing of Tone Systems". Paper held at ACL European Chapter Meeting, Copenhagen

Kaplan, R. M. , M. Kay 1981. "Phonological rules and finite-state transducers". Paper held at the Annual Meeting of L.S.A. in N.Y.C.

Kay, M. 1982. "When meta-rules are not meta-rules". In: Spark-Jones & Wilks: (eds) *Automatic Natural Language Parsing* CSCM-10, University of Essex

Kay , M. 1984. "Functional Unification Grammar: A formalism for machine translation". Proceedings of 10th International Conference on Computational Linguistics: 75-78

Kimball, J. 1973. "Seven Principles of Surface Structure Parsing in Natural Language" In: *Cognition* 2/1:15-47

Koskenniemi, K. 1983. *Two-level Morphology: A general computational model for word-form recognition and production.* University of Helsinki, Department of General Linguistics. Publications No.11

Selkirk, E. O. 1980. *On Prosodic Structure and its Relation to Syntactic Structure.* Indiana University Linguistics Club

Selkirk, E. O. 1982 "The Syllable" In: Van der Hulst & Smith (eds) *The structure of phonological representations* (Part II) Foris Publications, Dordrecht