



## A SYSTEM FOR LEFT-TO-RIGHT INTONATION SPECIFICATION FROM TEXT.

Alex Monaghan.\*

### ABSTRACT

This paper describes some computational strategies employed in the implementation of the model outlined in papers by Ladd and Ladd & Monaghan in this volume. There are two main categories of problem involved in extracting intonation from text: the first derives from the (currently) very limited nature of higher-level information deducible from text; the second consists of problems of interpretation which would exist even if perfect high-level analyses of text were available. Both these categories can be resolved with reasonable success in our Left-to-Right process model by using various levels of representation and employing computational techniques such as default specification and recursion. Factors affecting intonation at a high level include semantic, syntactic and pragmatic considerations, most of which are not explicit in text. Our model uses a small number of abstract PITCH ACCENT types in conjunction with limited syntactic and pragmatic information and a number of default clauses to generate a wide range of intonation contours.

### 1 LEVELS OF REPRESENTATION

Any adequate model requires explicit representation of all relevant levels, and considerable thought was given to determining precisely which levels were relevant to intonation. The desire to avoid speaker-specific representations as much as possible, and the adoption of intonational tunes (see Ladd, this volume), led to the choice of three descriptive levels: abstract phonological, abstract phonetic and concrete phonetic.

#### 1.1 Abstract Phonological Representation.

This level defines a fairly abstract intonational contour by specifying accent location and degree (major or minor). Also represented at this stage are register steps and phrasal boundaries (see Ladd, *ibid.*). The current system specifies default locations for all these elements automatically, and other items (such as non-default boundaries or extra register steps) can be entered by hand.

#### 1.2 Abstract Phonetic Representation.

The phonological representation is mapped onto this level using the tune chosen for the utterance: each accent degree phoneme is specified as the appropriate accent type from the tune, to give a number of abstract targets - generally three for major and two for minor accents. Boundaries are treated in a similar manner, but register steps are not interpreted at this level.

\*Centre for Speech Technology Research, University of Edinburgh.

There is another level which exists between this and the phonological, and which could be represented more explicitly: it might be called a concrete phonological level. Accent degrees are essentially independent of tune type, and do not specify the class of utterance (question, statement, command, etc.) being modelled: accent types do specify this, and a representation in terms of types between those of degrees and targets could be argued for. The view taken here, however, is that the abstract phonological specification together with the tune choice is sufficiently explicit and that another level would not enhance the model but would in fact complicate it unnecessarily.

### 1.3 Concrete Phonetic Representation.

This level is largely the result of putting the previous one through the F0 calculation algorithm mentioned in Ladd (this volume). Whereas both the preceding levels are entirely speaker-independent and can be used to produce an infinite number of utterances, this level is governed by speaker-specific parameters such as pitch range and normal speaking range. The intonation is represented by a series of absolute F0 target values through the utterance, and an interface program converts these into synthesiser input. Register steps are also interpreted at this stage, modifying the 'register' setting of the F0 calculation (see Ladd, *ibid.*).

### 1.4 An Illustration: Paragraph Intonation.

A discussion of the techniques involved in generating paragraph intonation will illustrate the interaction of the various levels mentioned above. Some form of 'preprocessor' is assumed which will identify paragraphs in text and mark them (as, say, 'B' for 'beginning' or 'E' for 'end') in its output.

The desired effect of a 'B' in the input to the intonation component is to issue an upstep ('<'), i.e. raise the register and thus ultimately the pitch of the current phrase. This can be achieved simply by the addition of a clause at the abstract phonetic level:

```
ass(Tune,['B'|Rin],A_so_far,Out:- ass(Tune,['<'|Rin],A_so_far,Out).
```

which simply replaces 'B' with an upstep and recurses through the rest of the input string, leaving the upstep itself to be interpreted at the next level. Achieving paragraph-final downstep appears at first sight to be a far thornier problem, but a multi-level representation coupled with the versatility of PROLOG data-structures permits a solution of the following form:

```
ass(Tune,['E'|Rin],A_so_far,Out):- ass(Tune,Rin,['>'|A_so_far],Out).
```

This clause inserts a downstep as the head of the Answer\_So\_Far argument, ensuring that the concrete phonetic level will downstep all pitch accents in the current domain. No complicated 'lookahead' procedures are involved, and two clauses are all that is required to model paragraph effects on intonation: a highly efficient solution. This approach has two more advantages over less abstract models:

firstly, the clauses can be rapidly and extensively modified without the need to alter other areas of the system, since each clause stands in isolation and performs purely symbolic manipulations; secondly, the very fact that a symbolic representation is preserved for as long as possible gives this system much greater flexibility.

## 2 DEFAULTS & EXCEPTIONS

The use of default specification is a common method of expressing generalisations, particularly in AI programming. In our system it has been extensively applied to those areas where inadequate input is the rule and good heuristics the only solution, particularly semantics and pragmatics. These areas are very important in determining the abstract phonological representation of any input text, but no automatic system currently approaches a complete analysis of either: a system of defaults, which can be refined and extended indefinitely, is thus essential.

### 2.1 Defaults.

Our current system relies on a minimal syntactic parse to identify the items to be accented. Each 'word' is assigned a clause of the form phrase (item) which represents the current construction and the item's grammatical category: the utterance The dead king cursed us would be represented thus:

[np(det), np(adj), np(n), vp(v), np(pr)]

From such a representation we extract essentially nouns, adjectives and verbs for default accent. Neutral intonation entails assigning accents to the heads of syntactic phrases such as NPs and VPs: the head must generally be a 'major lexical item', e.g. noun, verb, adjective. By default, nouns receive primary accent and the others secondary. This relatively simple procedure results in reasonable intonation patterns for many shorter sentences, although there are obviously many exceptions to such a general approach: adverbs of manner, for example, are often accented whereas those of time and place are not. As yet, no analysis of the criteria for accenting minor categories is available and so no attempt has been made to handle them in an 'intelligent' fashion. However, several classes of exceptions to the default procedures have already been resolved in a principled manner in the current system, and the two examples below illustrate our current approach to specifying such exceptions.

### 2.2 Deaccenting in Noun Phrases.

Deaccented nouns are those which are so predictable from context that they are semantically empty, or very nearly so. This class consists principally of proper nouns such as Street, Building, and Land: they constitute a small class, and could easily be specified in the lexicon component of any speech-output system. They are accommodated in the current model by the addition of a second argument to the PROLOG clause conveying syntactic information, thus:

np(pn,l)

where  $\bar{1}$  indicates that the item is deaccented: other values can be assigned where necessary to distinguish sub-classes of a syntactic category. This same approach can be extended to handle pragmatic deaccenting and rhythmic phenomena in general (see Ladd & Monaghan, this volume).

### 2.3 Predication.

It is widely accepted that the major semantic boundary in most sentences falls between the subject and its predicate. This is recognised in the present system by placing a downstep before the predicating verb or adjective to mark this boundary, and this provides an accurate approximation to F0 contours in natural language. There is, however, some difficulty in identifying the predicator in many sentences, and for this reason two types of verb phrase are currently distinguished on semantic grounds: standard verb phrases (vp) and copular phrases (cp).

Copular phrases contain a head verb which belongs to a finite set {GO, BE, HAVE, STAY, ...} followed by some predicator such as an adjective or a prepositional phrase. In such phrases the verb is DEACCENTED and the accent on the adjective, etc. is increased: there is no need to move the downstep.

The standard verb phrase is the default case, and currently a secondary accent is assigned to any verb not specified as copular and a downstep is placed immediately before it. There are still problems with this, particularly with intransitive verbs for which no specific rules have yet been developed.

## 3 SUMMARY

Two basic principles underlie our system design. The first is the versatility of symbolic computation and the preservation of an abstract representation using a multi-level design. The efficiency of symbolic processing, and its appropriateness for Natural Language tasks, allows us to dispense with complex time-related equations and lengthy search strategies and gives our Left-to-Right system greater generality and adaptability than many far less constrained.

The second principle is that of default specification to as great an extent as possible. Given an infinite domain as complex as intonation clearly is, a strategy of broad generalisation from minimal information is unavoidable and such a strategy lends itself well to subsequent refinement and modification, as in the approaches to deaccenting and the various uses of register steps discussed above. Our system is also a tool for its own improvement, and modularity and generality of rules are vital to that function.