# A Computational Model of P&P: Dresher & Kaye (1990) Revisited*

**Steven Gillis, Gert Durieux, Walter Daelemans**
**University of Antwerp, U.I.A.**

Verrips & Wijnen (eds) Approaches to parameter setting, ASCLD 4, 135-173, 1995

Language acquisition research in the Universal Grammar tradition has witnessed a wealth of studies focusing on various aspects of phonology and syntax. The concept of parameter setting as the core of acquisition is at the heart of these studies.

As a methodology, computational modeling has hardly given rise to experimental studies that actually implement the theoretical constructs invoked by and utilized in acquisition studies. Nevertheless, computer modeling is a powerful tool for studying highly complex phenomena such as the intricate interactions between the language acquisition data and the process of parameter setting.

A notable exception to this situation is Dresher & Kaye's (1990) computational model YOUPIE that incorporates a UG approach to the acquisition of a phonological subsystem, i.e. stress assignment as it is treated in metrical phonology. We analyze this model focusing mainly on the learning theory incorporated in the model, i.e. the way in which UG mediates between the data and the grammar constructed by the learner.

This investigation will focus on two aspects of the learning theory. First of all, the requirements formulated with respect to the learning theory will be evaluated against their implementation in the actual model. We will note several mismatches between the two. Secondly, we present an empirical test of the model. The model's production component is used to generate a 'language' for each possible parameter setting. Then, the model's learning component is used to acquire the grammar of each individual language. The outcome of the experiment reveals several problems in empirical coverage of the model, and relates some of them to inherent design choices.

## 0. Introduction

In a recent paper, Goldsmith (1994: 95) accurately describes the language acquisition scene as follows:

> Any working generative grammarian is well aware that s/he is working within an edifice whose theoretical foundations are composed of the learning principles used by the human Language Acquisition Device.

* The authors want to express their gratitude to E. Dresher for providing the source code of the computer program discussed in the text. Preparation of this paper was supported by a Research Grant of the Fund for Joint Basic Research (FKFO 2.0101.94) of the National Fund for Scientific Research (NFWO) and by the Interuniversity Research Program on Linguistic Pragmatics (IUAP-II, contract number 27). S. Gillis is research associate of the Belgian National Science Foundation (NFWO). Address for correspondence: Center for Dutch Language and Speech, Department of Linguistics (GER), University of Antwerp (U.I.A.), Universiteitsplein 1, B-2610 Wilrijk (Belgium). E-mail: gillis@uia.ua.ac.be.

> Virtually all generative work over the last thirty years has appealed to one version or another of Chomsky's fundamental argument, the argument that concludes that there is a rich mental endowment containing a number of principles of Universal Grammar, in roughly propositional form, without which it would be impossible to imagine that children could arrive at the grammatical conclusions that they apparently do.

This quote captures the tremendous impact of the theory of Universal Grammar (henceforth UG) on the study of natural language acquisition. The study of syntax has largely been cast in the terms of the Chomskyan Principles and Parameters approach (Chomsky 1981, 1986). To a lesser extent, the same is true for phonology as well, where a number of parametrized theories of stress have been proposed (Halle & Vergnaud 1987, Hayes 1981, 1987). Recently, these theories have provided the background for psycholinguistic studies of L1 and L2 acquisition (Fikkert 1994, Archibald 1993).

The theory of UG, as embodied in the Principles and Parameters approach aims at providing an answer to two of the fundamental questions that Chomsky posed as a research program for linguistics: (i) What constitutes knowledge of language? and (ii) How do humans acquire the knowledge of their language? The first question is traditionally answered by equating knowledge of a particular language with knowledge of an explicit grammar capable of generating all well-formed sentences of the language. The second problem is of a still more challenging nature, as it involves a paradox: on the one hand, language acquisition is remarkably robust, both cross-linguistically and among individuals. On the other hand, the input to the human learner is assumed to be deficient, in the sense that it is both unsystematic and incomplete, and may be distorted by factors such as performance errors, slips of the tongue etc. The key to resolving this paradox has been to assume a rich genetic endowment, embodying the innate cognitive principles that enable a learner to acquire any language, hence the term Universal Grammar.

Under the Principles and Parameters approach, Universal Grammar consists of a finite number of principles, each of which involves a finite number of parameters. The parameters can take only a finite number of settings, so that the set of possible target grammars is restricted. This bounded number of possible grammars has been invoked to explain cross-linguistic variation, and is regarded as a considerable aid in acquisition: since UG puts firm constraints on the form of possible grammars, the learner does not entertain implausible hypotheses during acquisition. Moreover, grammar learning is taken to consist of a relatively simple process of parameter fixing, and not of inducing grammar rules by other (as yet unraveled) means.

It is rather surprising that, notwithstanding the pervasive influence of the paradigm, little or no attempts have been made to build computer models that incorporate the theory and could thus provide further substantiation to the underlying assumptions. In other words, although in the Chomskyan paradigm a well-articulated theory exists that is formalized to the extent that it would permit computer modeling, the latter step has hardly ever been taken. A notable exception is Dresher and Kaye's implementation of a UG grammar approach to stress acquisition (Dresher & Kaye 1990, henceforth D&K, see also Dresher 1992). They consider the problem of stress acquisition a major testing ground for parameter based theories, because "the parameters of metrical theory exhibit intricate interactions which surpass in complexity the syntactic parameters studied so far. The issues raised by this study thus provide another perspective on the learnability of parameter systems, with consequences extending beyond the domain of stress" (D&K, p. 136). They advocate the use of computer models in this particular area, both because computers are "useful tool[s] in the study of interacting parameters that combine to create systems of some complexity" (D&K, p.137), and because "parametrized theories lend themselves quite well to computer modelling" (D&K, ibid.).

More generally, as a methodology, computer modeling offers a number of interesting perspectives. First of all, building a working program involves making every detail of a theory explicit, and may thus help pinpoint ill-understood or only partially specified mechanisms. Sometimes simulation is the only way to study the predictions of a theory, viz. those which defy formal analysis due to the intricate interactions among components. Gibson & Wexler (1994) provide a pertinent illustration: they construct a parameter space that comprises only three binary parameters relevant to account for word order variations, and apply these to the problem of Verb Second phenomena, a well-studied domain in syntax acquisition. In that parameter space a simple program discovers the existence of *local maxima*, i.e. intermediate grammars that do not allow the learner to eventually reach the target grammar. The program achieves this result by computing every possible intermediate grammar and checking if the available data can eventually lead the learner to the final or adult grammar. The enormous amount of possible combinations to be checked is easily dealt with by a computer program, while for a human analyst, the task of laboriously investigating every single combination is hardly feasible. The latter point is clearly illustrated by the fact that in the well-studied domain of verb placement, the existence of local maxima has never been hinted at in the psycholinguistic literature.

Finally, in the absence of detailed and empirically verified psycholinguistic models, a computer implementation may provide an account of cognitive phenomena at some abstract computational level.

In this paper we provide an in-depth analysis of D&K's stress learner. YOUPIE (for Universal Phonology) is a running computer model that incorporates the relevant aspects of UG. The model is ultimately meant to provide a computational account of how in a UG the projection problem is solved: how does UG act as an intermediate between the primary linguistic data (the system's input) and the grammar eventually constructed by the learner (the system's output). In a first section we will provide some background about the domain of linguistic stress, viz. the theory of metrical phonology. The basic theoretical constructs will be discussed as far as they are relevant to D&K's model. Crucial in this respect is, of course, the formulation of the parameters that are part of UG. This section will be mainly expository since we do not aim at providing a contribution to metrical phonology and/or its acquisition. Next, we will introduce D&K's model: the general architecture of the model and - especially important for a model of language acquisition - the learning theory incorporated in the system will be presented. The information and information processing mechanisms the learner needs to arrive at a correct language-specific setting of the universal parameters will be highlighted.

The merits of YOUPIE as an acquisition model will be assessed from two different angles. First of all, D&K formulate a number of requirements a P&P model and the learning component it incorporates should adhere to. We will investigate to what extent these requirements are met in the actual implementation. Secondly, we will present the results of an empirical test of the model. In several publications (see Dresher 1992, Dresher & Kaye 1990) YOUPIE is described in considerable detail. However, the discussion of the actual success of the model in acquiring the stress systems of various languages is fairly limited. In this paper we offer a test of the empirical adequacy of the learner: YOUPIE's production component will be used to generate a 'language' for every possible configuration of parameter settings, and these 'languages' will in turn be fed into the learning component of YOUPIE in order to assess their learnability by the system. For this test of the actual performance of the system, as well as for the analysis of the implementation, we crucially depended on the program code provided by Elan Dresher.

## 1. Metrical Phonology: Stress Assignment

### 1.1 Metrical Phonology

Metrical phonology is a branch of non-linear phonology that is concerned with phonological constituency and the prominence relations that hold between categories at various hierarchical levels. One of the major innovations of metrical phonology as it evolved[1] from work by i.a. Liberman (1975) is that stress is not a phonemic feature of

[1] We will restrict our discussion here to the approach adopted by D&K, viz. an analysis in terms of metrical trees (see also Hayes, 1981, Giegerich 1985, i.a.). Though the details of various alternative formalisms differ, this does not affect the arguments developed in this paper in any substantial way. Kager (in press) provides a state-of-the-art overview of the metrical literature. For descriptions of Dutch we refer the reader to Trommelen & Zonneveld (1989), Kager (1989) and Daelemans et al. (1994). Fikkert (1994)

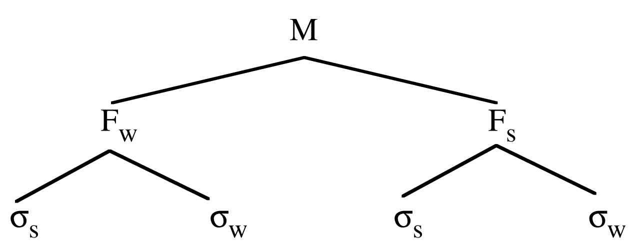
individual vowels in a word (as earlier generative accounts, exemplified in Chomsky and Halle's Sound Pattern of English, would have it), but that stress is a relative property to be captured in a hierarchical structure. Stress is conceptualized in terms of the relative prominence of syllables. This characteristic of stress is represented in metrical trees in which nodes are labeled strong or weak. A node is strong not by virtue of some inherent property, but because its sister node (in a binary branching structure) is weak.

A major contribution to metrical phonology that set the standard for much subsequent research was Hayes (1981). Hayes brought together data from a wide variety of stress systems, incorporated them into a unified metrical framework and showed how parametric variation brings about the observed cross-linguistic variety. The parameters (which are incorporated in D&K's learner) relate to the various levels in the metrical hierarchy represented in a metrical tree.

We will first turn to the structure of metrical trees and to the distinct levels of the prosodic hierarchy. Then we will discuss the metrical parameters and their relation to the distinct levels of the hierarchy.

In (1) we illustrate prosodic constituency up to the word level: a prosodic word (M) is analyzed as consisting of metrical feet (F), which in turn consist of syllables ($\sigma$). Each higher level consists of units from the level immediately below it, and the hierarchy is closed: the prosodic word consists of feet, and feet consist of syllables. Moreover, every syllable of a word should be included in the metrical structure (*exhaustivity condition*), and every word should have a stressed syllable, or in other words, a word should consist of at least one foot (*culminativity condition*).

(1)




The structure in (1) shows a binary branching metrical tree in which every node is labeled either strong or weak: feet are labeled weak-strong and syllables are labeled strong-weak. Determining the main stress of a word is accomplished by following the path that consists exclusively of strong nodes. Candidates for secondary stress are the strong syllables dominated by a weak foot. The tree in (1) represents, of course, only one

and Nouveau (1994) are empirical studies of stress acquisition.

possible configuration. The work of Hayes (1981) raised the status of metrical tree theory to a universal stress theory: examination of different tree-geometrical properties led to the formulation of a limited number of parameters that characterize possible stress systems. Consequently, a dividing line is drawn between possible and impossible stress systems.

## 1.2 Metrical parameters

Metrical parameters basically provide the guidelines for building metrical trees. Languages appear to show systematic variations with respect to the impact of syllable structure, foot formation and, at the highest level considered, the word-tree. The relevant parameters as they are presented in Dresher & Kaye (1990: 140-1) are enumerated in (2).

(2)

| Parameter | Description and values |
|---|---|
| P1 | The word-tree is strong on the [Left/Right] |
| P2 | Feet are [Binary/Unbounded] |
| P3 | Feet are built from the [Left/Right] |
| P4 | Feet are strong on the [Left/Right] |
| P5 | Feet are quantity-sensitive [No/Yes] |
| P6 | Feet are quantity-sensitive to the [Rhyme/Nucleus] |
| P7 | A strong branch of a foot must itself branch [No/Yes] |
| P8A | There is an extrametrical syllable [No/Yes] |
| P8 | It is extrametrical to the [Left/Right] |
| P9 | A weak foot is defooted in clash [No/Yes] |
| P10 | Feet are noniterative [No/Yes] |

These parameters are taken to be part of UG. The grammar of a particular language consists of the appropriate values for the parameters. The acquisition process takes the form of determining the values of the parameters for the language to be learned. This means that the parameters have a universally preset value, and the learner decides whether this value agrees with the appropriate value for the input language. If this is not the case, the learner 'switches' to another value (which is also given in UG). The values between square brackets in (2) are those provided in UG. The first value mentioned is proposed by D&K as the 'unmarked' value, or more accurately, the default value for the parameter. The exact meaning of 'unmarked' or 'default' value will be clarified in our discussion of D&K's learner. We will first explore these parameters in some more detail.

The prosodic word is the appropriate domain for stress: main stress is controlled by an unbounded word tree in which the leftmost or rightmost node is labeled strong (P1). The other nodes are labeled weak.

A number of parameters relate to the shape of metrical feet: P2 (*boundedness parameter*), P4 (*headedness or foot dominance parameter*), P5 (*quantity sensitivity parameter*), P6 (*weight parameter*), and P7 (*quantity determined or obligatory branching parameter*). P2 distinguishes languages with feet consisting of up to two syllables (bounded) from languages in which no such restriction applies. P4 determines the head of a foot. In a right-headed foot the right node is dominant while the left node(s) is/are recessive. In left-headed feet the converse holds. P5 distinguishes quantity-sensitive from quantity-insensitive feet. In languages with quantity-insensitive feet, the distinction between light and heavy syllables has no effect on foot construction, in the sense that both can occur equally well in head or recessive position. Quantity-sensitive languages have a restriction which prevents heavy syllables from occurring in recessive positions, i.e. as weak branches of a foot. P7 adds an extra requirement in this respect, viz. that dominant nodes must be heavy syllables. P6 determines syllable weight on the basis of the internal structure of the rhyme or the nucleus of the syllable. If feet are quantity sensitive to the rhyme then branching rhymes (syllables with a long vowel and closed syllables) are heavy, non-branching rhymes (open short voweled syllables) are light. If feet are quantity sensitive to the nucleus then syllables with a long vowel are heavy and all others are light.

Parameters P3 (*directionality parameter*), P8A, P8 (*extrametricality parameters*), and P10 (*iterativity parameter*) concern foot construction. P3 indicates the direction in which foot construction proceeds: either from the right word edge onwards or from the left word edge onwards. P10 determines if the string of syllables is parsed into feet entirely or if only a single foot is constructed at a word edge. In languages which only exhibit primary or main stress and no secondary or ternary stresses, foot construction is noniterative.

Extrametricality (parameters P8A and P8) became a key concept in metrical phonology. It entails that an element marked as extrametrical becomes 'invisible' for the stress rules. The aim of extrametricality is to widen the window for stress assignment. For instance in a language with bounded feet and final stress (P1 = Right), main stress will be restricted to the final foot. Hence, stress on the third syllable from the word edge is excluded. In languages of this type (e.g., English and Dutch) main stress on the antepenultimate syllable does occur. By making the final syllable extrametrical, the word's final foot can consist of the penultimate and antepenultimate syllables, and hence, antepenultimate stress becomes a possibility. Extrametricality is restricted to peripheral elements and is further constrained by a number of conditions (see Kager 1995).

Parameter P8A captures the fact that a language may or may not have an extrametrical peripheral syllable, and parameter P8 refers to the right and the left peripheral elements that may be extrametrical.

Finally, parameter P9 (*defooting parameter*) refers to an operation of defooting in the case of a clash which results in destressing.

From the description of the parameters it is obvious that all parameters are not independent (D&K, p. 145 list the interdependencies). For instance, if P5 (the quantity sensitivity parameter) has the value quantity-insensitive, P6 is suspended, i.e. syllable weight is of no importance whatsoever. Moreover, for a quantity-insensitive language, P2 must be binary. As a result, the combination of P2, P5 and P6 allows five instead of eight different possibilities. A second example of the dependence of parameters: if a language does not have any extrametrical elements (P8A = No), P8 (the edge of extrametricality) is suspended. This means that P8A and P8 allow three values instead of four.

These parameters define the possible stress systems of the world's languages.[2] As such they constrain the hypothesis space to a considerable extent. Whereas an unrestricted mapping between weight strings (i.e. strings of syllable weights) and stress strings (i.e. strings of stress values for each syllable of a word) would allow a number of stress systems in the order of $1.2 \times 10^{24}$ for strings consisting of up to four syllables, the interaction of eleven binary parameters reduces this number to $2^{11}$, i.e. 2048 possible stress systems. Taking into account the interdependencies between parameters, there remain 216 possible stress systems.[3] In this sense, metrical theory seriously cuts down the size of the hypothesis space, i.e. the number of possible grammatical hypotheses.

Thus, one of D&K's explicit motivations for incorporating parameters in their computational model is that parameters restrict both the number and form of possible stress systems. By imposing this restriction, possible stress systems (those agreeing with a parametric configuration) and impossible stress systems (systems that cannot be described in terms of such a parametric configuration) are easily distinguished. The learner can greatly benefit from this, since, in the course of grammar acquisition, he will not entertain hypotheses that can be designated a priori as impossible.

---

[2] Note, however, that D&K admit that the current set of parameters may well need further elaboration: "The current set of parameters covers many of the basic cases treated by Hayes (1981) and Halle and Vergnaud (1987), though these works also treat many languages whose stress patterns could not be accounted for in terms of these parameters alone. To bring the model to the point where it would equal their empirical coverage, a number of other parameters would have to be added." (D&K, p. 175) This claim is also made by Gupta & Touretzky (1994) who point out a number of languages whose stress systems cannot be accurately covered by the proposed parameter set.

[3] For reasons to be explained later, D&K do not take the defooting or destressing parameter (P9) into account when calculating this figure.

## 1.3 Parameters and Core Grammar

Parametric variation brings in another distinction that is a commonplace in the UG approach, viz. the distinction between the 'core' and the 'periphery'. In the UG approach, the parameters account for what is called the core grammar of the language. In other words, that part of the grammar of a particular language that can be captured in terms of a specific setting of the parameters, is said to constitute its core grammar. In addition to the core grammar, a language may have language-specific constraints that lie outside the core grammar. As a reflection of the metrical universals, the core grammar does not account for language-specific constraints. The latter may well be parametrized (Fikkert 1994), but language-specific peculiarities may as well arise from other sources: (i) effects of phonology, i.e. phonological processes may obscure the underlying stress pattern of a language. An example is provided by languages which delete unstressed vowels. (ii) Effects of morphology, i.e. morphological structure may influence stress placement, as is the case with certain suffixes which consistently attract stress. (iii) Lexical accent, i.e. stress is determined by rule but in addition also by lexical specification of particular stress-related properties. For instance, a language may have extrametrical syllables, but for some (types of) words a lexical marking may be required to indicate that the appropriate syllable is not subject to extrametricality.

Thus, languages may have specific properties that fall outside the scope of UG. Moreover, peripheral phenomena (viewed as distortions of core phenomena) may stem from interactions with phonological regularities, morphological interference, as well as (partial) lexical marking. Last but not least, a language may show particular exceptions.

The distinction between 'core' and 'peripheral' grammar raises two crucial issues: (i) To what extent can core grammar be acquired in the presence of input items that are peripheral? and (ii) What mechanisms are responsible for the acquisition of the periphery? It is clear that in a computational model these issues should be dealt with explicitly: it is not sufficient to make an analytic distinction between core and periphery, and then turn exclusively to the acquisition of the core grammar. Learnability results would then be valid only for the acquisition of the core grammar, leaving the matter undecided whether the obtained results are generalizable in any way to the acquisition of the core grammar plus the peripheral grammar. This is an important point, because in the input, core and peripheral items are not marked as such and peripheral items may well disturb the acquisition of the core grammar in a devastating way.

D&K make a firm claim with respect to the acquisition of the core grammar: "Core parameters must be learnable despite disturbances to the basic pattern caused by opacity, language-particular rules, exceptions, etc." (D&K, p. 156) They reason that, since core grammar is determined by UG, it must be largely preprogrammed and learnable by robust processes that will not be misled by peripheral phenomena. Moreover, because there are few criteria for making a principled distinction between core and periphery, learnability may ultimately prove to be a touchstone: those phenomena that can be acquired despite peripheral 'disturbances', may well be considered to be core phenomena.

As to the acquisition of the periphery, D&K are less articulate: "The periphery would have to be learned by less principled means; the task, however, is much simplified once the core system is in place." (D&K, p. 163) This contention is fully in line with the claims about the acquisition of the core grammar: learnability of the latter should be robust, not disturbed by the more bizarre peripheral phenomena. Once the core is in place, acquisition of the periphery could then be achieved by "less principled means". What these means might be, is left undecided. In the next section we will present YOUPIE in some more detail, and it will become clear that the system's input has to be filtered to a considerable extent. More specifically, the input is to be completely transparent, not messed up by peripheral phenomena: "In terms of the learning system under discussion here, it is necessary to prevent the learner from receiving conflicting information." (D&K, p. 183) D&K devote an entire section to the question of how the system may retain its robustness notwithstanding conflicting information (stemming from language-specific rules, exceptions, morphological sensitivity, etc.) The mechanisms they propose have not been implemented in YOUPIE yet, so that it is unclear whether the system can acquire the core grammar in the presence of those disturbing factors. Moreover, the mechanisms for acquiring the peripheral grammar fall completely outside the scope of D&K's system.

## 2. YOUPIE: System architecture and functionality

In this section we will describe D&K's YOUPIE program, focusing on the general architecture of the system and on the different stages involved in arriving at the grammar of a language. In the next section the actual learning component will be analyzed.

The system presented by D&K takes as input a list of words of a language and outputs a specification of the values of the metrical parameters. These parameter settings constitute the core grammar of the language under consideration. In YOUPIE, learning is thus equated with parameter setting: "by 'learn', we mean that, given input data and a model of universal grammar (UG) which includes a set of open parameters, the program contains a procedure which can correctly fix the parameters, and can then apply the system so as to generate well-formed strings" (D&K, p. 136).

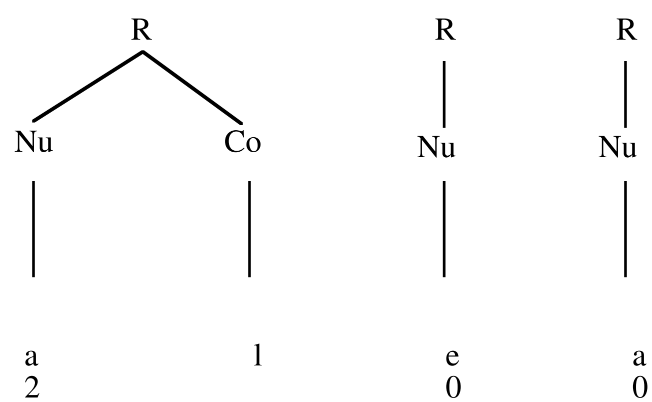
The system consists of five modules: the *Syllable Parser*, the *Classifier*, the *Learner*, the *Applier*, and the *Cranker*, and operates in batch mode. Batch mode learning consists in gathering inputs for a fixed amount of time, and processing those inputs all at once as soon as the collecting phase has halted. We will briefly discuss these modules on the basis of Dresher & Kaye (1990).[4]

Input to the system are words of a target language. These words are phonemically transcribed and their stress pattern is indicated: main stress, secondary stress and stressless vowels are represented by 2, 1 and 0 respectively. An example is given below for the English word *algebra*.

(3) a2lge0bra0

Forms such as these are the input for the *Syllable Parser*. Since only the rhyme of a syllable is relevant for stress, the syllable parser constructs rhyme projections of the input words (such as exemplified in (4) for *algebra*) and returns these as the input for the next component. At the same time, a copy of the structure created in (4) is stripped from its stress markings and this stressless structure is set aside to be used later on by the *Applier*.

(4)




R = Rhyme, Nu = Nucleus, Co = Coda, 2 = main stress, 0 = no stress

Since the program operates in *batch mode*, performing an all-at-once learning procedure in which all relevant input is available at any time and learning is in principle instantaneous, rhyme projections of all input words are constructed before they are sent to the next module. Thus the complete set of input words is handled by the *Syllable*

---

[4] Dresher (1992) and the actual program code show a more complicated picture than the one presented here. However, the basic functionality of the system remains the same.

*Parser* before any other module performs any computation.

The rhyme projections are then passed from the *Syllable Parser* to the *Classifier*. This component checks whether "the system is transparent, that is, that there do not exist obvious conflicts whereby two words with identical syllable structure have different stress patterns." (D&K, p. 186) When such conflicts are detected they have to be solved before any learning can take place: "... the learning model will be unable to arrive at a successful setting of parameters. It is at this point, presumably, that other linguistic modules can be consulted in an effort to resolve the contradictions. Such modules can include an exception analyzer (...), a morphological component, other parts of the phonology, and so on." (D&K p. 186) These components are not implemented in YOUPIE, which is why the requirement that the input be completely transparent acts as a conditio-sine-qua-non for learning to take place.[5]

If the input is completely transparent, the rhyme projections are sent to the *Learner*. This module is responsible for analyzing the input and setting the metrical parameters to their appropriate values. The Learner incorporates a cue-driven learning mechanism that will be discussed in the next section. The output of the learner is a value for each of the metrical parameters, and this output constitutes the grammar of the language.

The parameter settings can now be used to determine the stress pattern of words. The *Applier* is the module that uses the grammar (parameter values) to check the correctness of the grammar by determining the stress patterns of the words that initially constituted the input to the *Learner*. Recall that the *Syllable Parser* gave as output two rhyme projections: one in which stresses were indicated (these were the input for the *Learner*) and a second from which the stress marks were removed. The latter are now used by the *Applier* to predict stress patterns. If the *Applier* predicts the correct stress pattern for each of the original input words, it may be assumed that the parameters are correctly set[6]. If not, parameter settings are assumed to be defective and the learning material is sent to the *Cranker*, a brute force learner that in principle checks every possible combination of parameter values against the input. The *Cranker* will not be dealt with explicitly, as it is a rather ad hoc component meant mainly to help resolve extrametricality issues. As D&K note (D&K 1991, p 193): "an improvement in the Learner's ability to deal with extrametricality could make the Cranker obsolete".

---

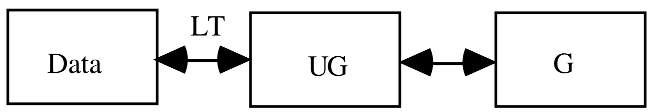
[5] Note that this situation is not very different from that encountered in psycholinguistic studies - as indicated above - where an analytic distinction is made between a core grammar and a peripheral grammar. Typically, the acquisition of the core grammer is studied, leaving the interference of peripheral phenomena aside. The complication this may cause is discussed in Wexler (1990), Valian (1990).

[6] In the actual program code, some mismatches trigger a second learning module, designed to deal with destressing issues.

## 3. The learning theory

UG, as specified in the eleven parameters proposed above and their mutual interdependencies, puts powerful constraints on the number and form of hypotheses that the learner may entertain. Still, to converge on the final grammar for a given language, a learning theory must come in to relate the primary linguistic data to the appropriate parameter settings. This threefold relation between input data, UG and target grammar is captured in the diagram in (5) (D&K, p. 137).

(5)




A learning theory may be conceived of in quite diverse ways and may involve different strategies on the part of the learner. D&K consider different empirically adequate learning theories which they evaluate with respect to the relationship they exploit between the data and the parameters of UG. The alternatives range from a brute-force learner, with minimal information about the relationship between the data and the associated parameter values, over a cue-based learner, which proceeds on the basis of specific patterns in the input, to a primed learner, where the relationship between parameters and the data is completely prewired. They ultimately reject both extreme approaches, because they fail to elucidate (and/or exploit) the highly structured nature of UG and argue in favor of a cue-based learner, for which they formulate several requirements. In the following section, we will first describe the brute force approach and the primed learner, because many of the issues involved will resurface in the discussion of the cue-based learner. We will also highlight D&K's main criticisms of these approaches. The cue-based learner will be treated in more detail in section 3.2. Section 3.3 rounds off with a discussion of the choices made in YOUPIE against the requirements formulated in the previous section.

### 3.1 Two alternatives to cue-based learning

D&K discuss two learning theories that they eventually reject in favor of a cue-based learning theory. These two alternatives represent two extremes with respect to the relation that is established between the data and the parameters of UG: a brute force learner is deprived of any insight in this relationship, while a primed learner has all information about the compatibility between a surface form and a set of parameters precompiled.

The brute-force learner is based on the assumption that UG restricts the solution space sufficiently to make exhaustive search feasible. Therefore, no special insight into the relationship between primary data and specific parameter settings is needed. The brute-force learner comes equipped with an initial set of parameter settings, a fixed agenda of other configurations to explore, and a procedure which applies the current settings to unstressed forms. Learning proceeds incrementally, i.e. if necessary the agenda of parameter settings to investigate is updated with every input word. Moreover, learning is driven by failure in the following manner: on encountering surface forms, the learner tries to match these with self-produced forms. In the case of an incompatibility, it tries out another configuration, until matching succeeds. Learning thus requires no memory of previously encountered inputs. The only knowledge needed is whether the current input is compatible with its current grammar. Provided that the learner never explores a previously rejected hypothesis again, this strategy will uncover the correct parameter setting if one exists, although the time required to do so may vary substantially both with the set of initial parameters and the kind of grammar to be learned.

Although this kind of learner may thus prove to be empirically adequate, D&K reject it because it is deemed uninteresting. First of all, there is no direct relationship between the primary data and the learning strategy. Although learning (change in parameter values) is triggered by failure, no lessons are learned from this failure. The order in which configurations are tried is completely inflexible and change in the parameters is not guaranteed to be for the better: due to the interactions of the different parameters in the production of output forms, a set of parameters that is off in a number of parameters, may still lead to the correct stress pattern for a particular type of word. Secondly, this kind of learner is incapable of exploiting local constraints, or alternatively, of linking up certain surface phenomena to the parameter settings that cause them.

The primed learner is situated at the other end of the scale: here all information about which parameter settings are compatible with which surface forms is built into the learner. This information takes the form of a table in which combinations of stress and weight strings are related to all grammars capable of generating them. A small portion of such a table is shown (6).

(6)

| Weight | Stress | Compatible grammars |
|---|---|---|
| LLL | 200 | G1 G2 G3 G4 ... |
| LLL | 002 | G5 G6 G7 G8 ... |
| HLL | 200 | G1 G2 G4 G5 G7 ... |
| LLH | 002 | G5 G6 G7 G9 .... |
| ... | ... | ... |

In this table, each of the grammars $G_i$ represents a possible configuration of parameter values. Weight strings (represented in this table as strings of light - L - and heavy - H - syllables) and stress strings (in which the stress level of each syllable is indicated, as for instance by the digits 0 and 2) are connected with appropriate grammars. Learning then proceeds by removing incompatible grammars as forms come in, until the learner converges on the correct setting(s). If more than one possibility remains open, the least marked setting is chosen, according to some criterion of markedness.

Contrary to the brute-force learner, this learner can discard large parts of the solution space based on a single form. However, it can only do so because it has been primed in advance, thus remaining subject to the same criticism, i.e. that it is unable to relate surface phenomena to the parameters that cause them. Learners of this type are at odds with the idea incorporated in UG that the diversity of stress systems is caused by the interaction of a small number of parameters, instead of being represented directly in the mind.

Thus, the gist of D&K's counterarguments is that in neither of these two learners a relationship is exploited between the data and the setting of particular parameters, so that ultimately both learners fail to shed any light on why stress systems are structured the way they are. Both learners reject a hypothesis upon failure without considering the cause of that failure, i.e. the relevance of a particular input item for a particular parameter setting. Upon failure the brute force learner takes another path in its search space, and 'blindly' explores an alternative configuration of parameter settings. The primed learner incorporates the complete set of possible stress systems and upon failure it simply discards a possible configuration without considering why the data are incompatible with the rejected stress system. Neither learner exploits the relationships among the different parameters themselves, nor the combined effect they have on output forms.

## 3.2 Cue-based learning

To remedy the shortcomings in the above mentioned learning theories, D&K propose a cue-driven learner, which proceeds on the basis of very specific patterns, called *cues*, in the input. Essentially a cue identifies which aspects of the data are relevant to a particular setting of a parameter. The presence of that cue in the input then triggers the appropriate parameter setting. Unlike the brute-force learner and the primed learner, a cue-driven learner systematically exploits the relationship between underlying parameter values and their surface manifestations in the input data to keep problems local.

With respect to the amount of knowledge required, cue-based learners occupy a middle ground between the two rejected learning theories, being neither completely ignorant nor completely primed in advance. Of course, knowledge about rhyme structure and syllable weight is assumed, together with knowledge of UG and the mutual parameter dependencies it stipulates. Constructing a cue-based theory involves solving a number of additional issues: first of all, for each parameter, a cue has to be determined. The problem here is that, contrary to what is claimed to be the case in the acquisition of syntactic parameters, it is not immediately obvious which elements in the data are relevant to which specific parameter, since the data are relatively undifferentiated: for each syllable only a rhyme projection is provided. A further complication arises from the fact that the effect of changing a single parameter varies considerably, depending both on the setting of the other parameters and on the type of word to be stressed. A simple example of this is the following: in quantity-sensitive languages, assigning binary feet from left to right results in the same structures as proceeding from the other direction for words with an even number of syllables. But for words with an odd number of syllables, all structures change. The challenge for a cue-based theory, then, is to define cues which take into account all possible surface manifestations of the underlying parameters.

Second, the choice of initial (or default) settings of the parameters is crucial as well. Ideally a cue-based learner has at its disposal a positive cue for every value of each parameter. In such a situation, both values of a binary parameter can be determined on the basis of positive evidence, and the choice of a default is essentially arbitrary (or can be guided by typological considerations of markedness). However, for some metrical parameters the situation is more complicated, since positive evidence appears to be available for only a single value of a parameter. The solution adopted by D&K is the following: "In such cases, one assumes as the initial setting the value for which there is *no* positive evidence. The learner is driven to the marked value by encountering just such positive evidence." (D&K, p. 164) In other words, if positive evidence exists to change a parameter in one direction but not for changing it into the other direction, the setting for which no evidence exists becomes the default. An illustration of this issue is the choice of P2's default. Here, positive evidence exists for boundedness, so unbounded feet should be the default in systems that allow them. At the same time, UG itself may pose restrictions on the choice of defaults (D&K, p. 160-1). Quantity sensitive languages only allow binary feet, so this should be the initial default option. However, once it is determined that the language is quantity insensitive, both options for P2 become open, and the default shifts to Unbounded feet, following the reasoning above.

This last point raises another issue: the availability of information over time. To the extent that some cues are dependent upon knowledge of previously set parameters, an order in which parameter setting proceeds needs to be imposed. A case in point is the cue for setting P1[7] which determines whether the word tree is strong on the left or the right. The cue is conceived in such a way that a foot-sized window at the word edges is examined in order to determine in which word peripheral foot main stress occurs. Since this cue involves a foot-sized window at the word edge, at least information about foot size is needed, i.e. the value of P2 (feet binary or unbounded) should be available. P2's value itself depends on whether the system is QS or QI, since P5 = QI only allows binary feet, whereas the default in QS systems is P2 = Unbounded. Additionally, the correct value of the extrametricality parameters P8A and P8 is required for P1, since those influence the choice of the proper word edge. Thus, a cue-based learner faced with interdependencies between parameters such as these should accommodate a mechanism for setting the value of dependent parameters only when the needed information is available. D&K's learner accomplishes this by imposing a strict order on the acquisition process, setting independent parameters first. Hence, not all parameters are readily available from the very beginning of acquisition.

Consequently, D&K propose a threefold distinction between the initial value of a parameter, the marked value, obtained by a parameter switch, and a "frozen", unmarked value (D&K, p. 172). The latter denotes an initial value which is confirmed by a cue in the data. Both "frozen" unmarked values and marked values can be used reliably in setting dependent parameters, whereas initial unmarked values need further confirmation.

Apart from those general issues, which any cue-based theory must deal with, D&K pose three requirements which constrain the choices implied above, viz. *appropriateness*, *robustness* and *determinism*.

Appropriateness is defined as follows: "Cues must be appropriate to their parameters with regard to their scope and operation." (D&K, p. 155) This requirement is meant to

[7] See below for a detailed discussion of the cues for each parameter.

establish a principled relationship between cues and parameters. Since the definition is rather uninformative in itself, D&K illustrate the requirement with a cue for the detection of P3/P4 in Quantity-sensitive systems. The cue is formulated as in (7) (D&K, p. 155):

(7) a. P3 Left, P4 Left: scanning from the left, a light syllable following anything must be l (i.e. bearing neither primary nor secondary stress)
b. P3 Left, P4 Right: scanning from the left, a light syllable preceding anything must be l
c. P3 Right, P4 Left: scanning from the right, a light syllable following anything must be l
d. P3 Right, P4 Right: scanning from the right, a light syllable preceding anything must be l

These cues slide a foot-sized window across words, and reject the associated settings whenever a stressed syllable appears in what should be an unstressed position. These cues are appropriate because they mirror their parameters. Moreover, actual constituents of metrical theory are used, viz. syllables and feet, and they operate with notions such as weight, precedence and direction that the parameters themselves are also sensitive to. According to D&K, one of the beneficial effects of enforcing appropriateness is that it bans deduction and/or extensive computation from the Learning Theory.

The second requirement for a cue-based learner is robustness: core parameters must be learnable despite disturbances in the surface patterns. These disturbances may stem from various sources, such as exceptional words, lexical accents, morphological interference, language-particular rules or from destressing and rhythmic adjustments. Robust cues are cues that are not distracted by these influences. The cue for P3/P4 mentioned above illustrates this by abstracting away from the influence of destressing. Only stressed syllables in what should be an unstressed position influence the parameter setting. Proceeding on the basis of unstressed syllables in what should be stressed positions would break down whenever destressing is involved.

The third requirement is determinism. Determinism in this domain means that once a parameter is set to its marked value, it can never revert to its unmarked or initial state. The learning process thus proceeds monotonically, or, in other words, additional information can never revoke previous decisions. Determinism meshes well with the other two requirements, since a deterministic learner needs both robust and appropriate cues, whereas for a non-deterministic learner these properties of cues would be purely accidental.

Taken together, then, these requirements may partially account for the highly structured nature of the UG, and thus enhance the explanatory adequacy of the theory as a whole.

## 3.3 YOUPIE as a cue-based learner

D&K's main motivation for constructing a cue-based learner was to establish a principled relation between the input data and the relevant parameters, avoiding both blind search and unlimited deduction. They raised a number of general issues for this undertaking, involving the identification of cues, the choice of initial values, and the order in which acquisition proceeds. In addition three requirements were formulated (appropriateness, robustness and determinism) which serve to set up this relationship between data and parameters in a principled way, a way which does justice to the highly structured nature of the underlying UG.

In the next section, we will present D&K's choices as they are implemented in YOUPIE, and discuss them against the background given above. First we will focus on the nature of the parameter/data relationship, which we find problematic in several respects: topics which we will single out include the kind of evidence used, the way in which parameters are set, and the choice of (some) defaults. Then we will evaluate the system against the three requirements and show how these are only partially met.

In Table 1 all parameters are listed in the order they are set according to the learning theory D&K propose. Each parameter is accompanied by its default setting and a brief description of the cue(s) to determine the proper value.

Table 1: Parameters and their related cues

| | |
|---|---|
| *parameter:* | P5: feet are QS (Yes/No) |
| *default:* | No |
| *cue:* | If no pair of words with the same number of syllables, but differing stress patterns can be found, conclude P5 = No. |
| *parameter:* | P6: feet are QS to the (Rhyme/Nucleus) |
| *default:* | Rhyme |
| *cue:* | Convert all words to weight strings; assume that both long vowels in open syllables and closed syllables with short vowels count as heavy. If all words with the same syllable representations have the same stress patterns, conclude P6 Rhyme. If not, convert all words to weight strings where only branching nuclei count as heavy. If all words with the same weight string have the same stress patterns, conclude P6 = Nucleus. |
| *parameter:* | P10: feet are non-iterative (No/Yes) |
| *default:* | Yes |

| | |
|---|---|
| *cue:* | If no secondary stresses show up in the data, conclude P10 = Yes |
| *parameter:* | P8A: there is an extrametrical syllable (No/Yes)<br>P8: it is extrametrical on the (Left/Right) |
| *default:* | P8A:No, P8:Left |
| *cue:* | Extrametricality at an edge can be ruled out by the presence of a stress at that edge. |
| *parameter:* | P2: feet are (Binary/Unbounded) |
| *default:* | Binary (shifts to Unbounded if P5 = QS) |
| *cue:* | If a non-peripheral stressed light syllable is encountered, conclude P2 = Binary.<br>Alternatively, if both left and right peripheral stressed light syllables occur, not necessarily in the same word, conclude P2 = Binary. |
| *parameter:* | P1: the word-tree is strong on the (Left/Right) |
| *default:* | Left |
| *cue:* | If main stress falls consistently within a foot sized window at the left edge, conclude P1 = Left.<br>Conversely, if main stress occurs consistently within a foot sized window at the right edge, conclude P1 = Right. |
| *parameter:* | P7: a strong branch of a foot must itself branch (No/Yes) |
| *default:* | No |
| *cue:* | If there are stressed light syllables whose stress is not due to P1, conclude P7 = No.<br>Otherwise, conclude P7 = Yes. |
| *parameter:* | P3: feet are built from the (Left/Right)<br>P4: feet are strong on the (Left/Right) |
| *default:* | P3:Left, P4:Left |
| *cue:* | If a moving window sliding over the word encounters a stressed syllable in what should be an unstressed position, this setting is ruled out. (See also section 3.2) |

From Table 1 it appears that there are different relationships between the cues and the relevant parameters. Recall that, in principle, a cue-based learner switches from the unmarked initial value of a parameter to the marked value using positive evidence: "... one assumes as the initial setting the value for which there is no positive evidence. The learner is driven to the marked value by encountering just such positive evidence." (D&K, p. 164)

An analysis of the cues defined in Table 1 displays a much more varied picture, and does so on two different planes. On the one hand, the kind of evidence used for determining the values of parameters is not restricted to straightforward positive evidence, but also takes various other forms. On the other hand, this evidence does not always lead to the adoption of the marked value: sometimes the marked value is excluded, and sometimes the initial value is "frozen" (cf. section 3.2 above). In Table 2 a systematic overview is shown. For each parameter the kind of evidence used is indicated together with the action entailed by the evidence: cues may either look for positive evidence or rely on the absence thereof, and thus use some kind of indirect negative evidence. The action spelled out may consist in the adoption of a value, by which we mean that the parameter is either set to its marked state or frozen to the initial state, or may lead to the rejection of a value, i.e. this value is merely shown to be incompatible with the data, initiating a further search for the correct value.

Table 2: Overview of the cues, the kind of evidence exploited and the change entailed

| Cue for Parameter | Evidence<br>+ : positive evidence<br>- : absence of evidence | Action<br>+ : adopt value<br>- : reject value |
|---|---|---|
| P5 | - | + |
| P6 | + | + |
| P10 | - | + |
| P8 | + | - |
| P2 | + | + |
| P1 | + | + |
| P7 | + | + |
| P3, P4 | + | - |

As noted above, cues are preferably of the form ++ , i.e. using positive evidence to take positive action. Yet, less than half of the cues conform to this schema: P6, P2, P1 and P7. The cues for P8 and P3/P4 are of the form + -, which can be paraphrased as 'If X then exclude Y'. These cues merely establish that the current value of the relevant parameter is incompatible with the data. The cues for P5 and P10 are of the form -+, which can be paraphrased as 'If not (or never) X, then conclude Y' and thus rely on indirect negative evidence. In the context of a deterministic learner, this has an undesirable consequence: since a deterministic learner has no possibility to revoke earlier decisions, it has to have access to all relevant information before it commits itself to a choice. D&K explicitly opt for a batch mode learner in which learning is initiated after the data gathering phase, thus observing this requirement. However, as envisaged, that batch mode learner needs all relevant information before learning can start, and in so doing even slight oversights during data collection render the learner extremely vulnerable.

Another interesting aspect of the cues in relation to the kind of evidence used by the learner concerns cross-word comparisons. As discussed up to this point, the learner evaluates a particular cue, or more precisely, evaluates the test that the cue incorporates for each individual word, and decides on the basis of the outcome of the testing what action should be undertaken regarding the value of the appropriate parameter. This is for instance the case for P10: the cue entails testing every word on the presence of secondary stress. If no secondary stress occurs, P10 = YES may be concluded. An alternative procedure employed in D&K's learner does not only test single ('isolated') words but in addition requires cross-word comparisons. This is the case for the cues for P5, P6, P1 and (possibly) P2. For instance, the cue for P5 explicitly states that words with the same number of syllables have to be compared in search of word pairs that have different stress patterns. The cue for P6 even goes one step further: it requires that words should first be converted into weight strings (using two different conversion schemes) before cross word comparisons are undertaken.

Relying on this kind of evidence presupposes full memory for all input data. This situation is clearly in dissonance with one of the leading assumptions behind UG, where memory limitations on the part of the learner are invoked to substantiate the claim that a generative rule system is required. D&K try to weaken the requirement for full memory by maintaining that for the cues for P5 and P6 only weight and stress strings need to be stored. Their argument runs as follows: "That the LT [Learning Theory] could have access to cross-word comparisons of this type is supported by developmental studies which suggest that children in early stages of acquisition organize words in terms of preferred or canonical segmental and prosodic patterns (see Ingram, 1974; Macken, 1979; Menn, 1978). This kind of organization is compatible with the idea that children keep track of weight-string to stress-string mappings." (D&K, p. 174). While this argument may hold for P5, we believe that it is invalid for P6. In effect, P6 does not entail the direct comparison of word forms, but stipulates an additional operation on the data: the learner has to convert words to stress strings and to weight strings and should use two different conversion schemes for the latter. If full memory of word forms is to be avoided, this implies maintaining two parallel sets of weight strings, which is less rather than more economical compared to recoding stored word forms on the fly. Regardless of the plausibility of full storage, there is another potential problem with these cross-word comparisons: requiring consistency, as in the cues for P1 and P6, has the same consequences as using indirect negative evidence: it requires that all relevant evidence is present at the moment the learning process is initiated since a determinisrtic learner as the one proposed by D&K is not allowed to revert any previously taken decisions.

With respect to the actions spelled out by the cues, we already noted that parameters may be set, which involves a change into the marked value, or frozen, which means that the initial value is no longer changeable. Although the preferred form of cues is one that uses positive evidence to set parameters (e.g. P2), there are some cues such as P5, P7 and P10 that freeze the initial value. These particular cues can be brought in line with the preferred form by restating their condition part positively, and spelling out the opposite action (P5, P10), or by choosing the other parameter value as the default (P7). A different situation holds for the cues for P1, P6 and P3/P4. Both P1 and P6 use a two-way test to settle on the appropriate value, checking each value for compatibility with all input data. The cues for P3/P4 take this idea one step further, and check all combinations of parameter values against the data, eliminating incompatible configurations as they proceed. This procedure is essentially the one employed by the brute force learner discussed above, and dismissed by D&K because such a procedure lacks a principled connection between the data and the eventual parameter setting.

For the cues P5, P7 and P10, the issue of freezing versus setting and the related choice of an appropriate initial value was largely immaterial, as we discussed above. Yet this is not always the case. The reason D&K introduced the distinction in the first place was that a number of cues depend on the availability of values for other parameters. These dependencies are reflected in the order in which parameters are set or frozen. For a deterministic learner, however, reliance on a previous value requires this value to be absolutely certain. A problematic case here is P8/P8A, on which P2, P1 and P7 depend. There is a reliable cue to rule out extrametricality at an edge, viz. the presence of a stressed final or initial syllable. In line with the heuristic for choosing initial values, a default of P8A = Yes could then switch to the marked value 'No' upon encountering such evidence. However, D&K adopt 'no extrametricality' as the default option, "using a conservative strategy". Instead of provoking a switch to the marked value, the cue now allows the learner to freeze the initial value. Freezing this parameter value requires ruling out both Left and Right extrametricality. While the presence of stresses at those edges is a robust cue, absence of stresses at the edges stemming from other causes (e.g. non-iterativity, destressing) may prevent confirmation that the initial value is the right one, without leading to the opposite conclusion, which leaves an uncertain value for P8/P8A. To a certain extent, the same problem occurs with the cues for P1 and P6: when both tests succeed (P1) or fail (P6), no certain conclusions can be drawn.

Summing up the discussion so far, we found that in order to determine the value of parameters, much more is needed than cues that look for positive evidence to switch parameters to their marked value. A much more complex construction is needed: (i) the kind of evidence required is not restricted to positive evidence, but also includes indirect negative evidence, possibly with serious implications for robustness; (ii) the mechanisms invoked include using straightforward positive evidence, next to cross-word comparisons, sometimes involving extensive recoding of the input material. In most cases, these cross-word comparisons share the undesirable properties of using indirect negative evidence; (iii) in addition to mere switching to the marked value, or freezing of the initial value, exhaustive search is used in particular areas of parameter space; (iv) in some cases the appropriate value of a parameter cannot be established conclusively, causing the values of parameters which depend on them to be equally uncertain.

When the cues are evaluated against the requirements for a cue-based learner, viz. appropriateness, robustness and determinism, the following remarks can be made.

With respect to appropriateness, the cues for P5 en P6 are seriously lacking, since neither the number of syllables, nor weight strings and stress strings have any independent status in metrical phonology. Also, the cue for P7, which deals with constraints on foot formation, crucially depends on the value for P1 (word tree labeling), which is a distinct metrical domain, and might thus be considered inappropriate. Moreover, the mechanisms of cross-word comparisons, extensive computation required for recoding the input materials as well as consistency checks before and during learning are clearly in contradiction with the alleged beneficial effects of enforcing appropriateness upon the cues ("Adoption of the Appropriateness Condition ensures a desirable result, namely: no deduction or computation is permitted to the learner" D&K, pp. 157-158)

Robustness is at best partially obtained, given the requirement of transparent input. This requirement stipulates that all noise is to be removed from the input data before they are passed on to the Learner. Consequently, the requirement that the cues should be so robust as to 'look through' noisy input data, is considerably relaxed in the present system. Robustness of individual cues will be dealt with in the following section, where the performance of the system will be tested.

Determinism is maintained throughout in the Learner module, but the existence of the Cranker seriously compromises this achievement, as it can undo some results of the Learner. Recall that the Cranker is introduced to resolve inconsistencies between the actual stress pattern of the system's input words and the stress pattern of the same words generated by the Applier on the basis of the parameter settings provided by the Learner. If such differences occur, the Cranker looks for the setting of parameters that is most in accord with the input data by exhaustively searching the parameter space. The Cranker's output will replace the parameter settings provided by the Learner, even though this requires resetting the values of previously set or fixed parameters.

## 4. An empirical evaluation of YOUPIE's performance

Although D&K give a very comprehensive description of their model and discuss the underlying assumptions and issues involved at great length, they provide only a rather limited assessment of their model's performance. Apart from the general statement that YOUPIE performs quite well within the limits of the UG, no quantified results are communicated. The YOUPIE software comes with 18 data files (languages), containing between 2 and 15 words. Presumably these stress systems fall within YOUPIE's scope, though verification of this is hindered by the fact that no indication of the intended target grammar is given. Although the range of data files could easily be extended using the available metrical literature, we chose to test the system's performance in an alternative way.

In order to provide a first assessment of the system's performance, we conducted an experiment using artificially constructed data. The aim of the experiment was to test YOUPIE's accuracy in learning the stress system of languages in terms of their parameter setting. In order to respect the inherent limitations of the system, the experiment was restricted to learning those languages that are generated by the parameter settings that YOUPIE's UG allows. In section 1.2 it was calculated that the 11 metrical parameters define 216 possible parameter configurations if the value of P9 is kept constant to 'No destressing'. For each of those 216 stress systems, a set of pseudo-words was stressed by YOUPIE's Applier module, i.e. a system-internal component. As such the system's input was completely transparent since no noise was introduced in the learning material. Each of the ‘languages' obtained in this manner was in turn given as input to YOUPIE. The system returned for each language a configuration of parameters, as discovered by the Learner. This output could easily be compared with the parameter setting that was originally used to generate the language, and discrepancies in terms of the value of individual parameters could be detected.

In what follows we will first provide details about the generation of the learning materials, i.e. the artificial languages, and then proceed to a discussion of YOUPIE's accuracy in learning these languages.

### 4.1 Generation of artificial languages

In order to make a systematic assessment of the model's performance, we conducted a single, comprehensive experiment with synthetic data, encompassing all 216 'possible languages' that YOUPIE's UG allows. The rationale for using synthetic data sets was threefold: first, working with constructed data sets sidesteps the issues of data gathering and subsequent analysis. Second, D&K state quite clearly that many of the world's attested stress systems fall outside the scope of YOUPIE's UG. Moreover, for correct operation, YOUPIE's Learner module enforces the strict requirement that the data are completely transparent. Working with real data would therefore imply discarding all exceptional or language specific phenomena. Finally, we wanted to make a systematic assessment of the system's performance. Constructed data permit us to vary all relevant parameters without going beyond the system's inherent limitations.

The data for the experiment were obtained according to the following procedure. First, we generated all weight strings for words of two, three and four syllables. The weights used were L for light syllables, R for syllables with a branching rhyme and N for syllables with a branching nucleus, as these are the relevant oppositions within YOUPIE's UG. For disyllabic words this yields $2^3$ strings (LL, LR, LN, RL, RR, RN, NL, NR and NN). For trisyllabic words there are $3^3$ strings, for words of four syllables $3^4$, yielding a total of 117 different weight strings. These weight strings were then converted into pseudo-words conforming to YOUPIE's input format using a simple syllable grammar which rewrites the weight symbols into corresponding syllables. The grammatical component was set up to be unambiguous with respect to YOUPIE's syllable parser, ensuring the intended parse tree for each generated syllable. The words were then stressed under each of the 216 possible parameter settings, using YOUPIE's Applier module.

Constructing the data in this manner ensures two important properties: first, the generated artificial languages are completely transparent, since all surface forms are derived exclusively from the application of core grammar rules, without interference from exceptions, destressing, morphology or language-particular rules. Second, for words up to four syllables, all possibly relevant oppositions are present, yielding highly systematic input data for the learner.

## 4.2 Learning artificial languages

Each individual language was given as input to YOUPIE. Thus, the Leaner was given a total of 216 different languages to learn. Although in YOUPIE, the parameter setting obtained by the Learner module is subsequently passed on to the Cranker module, which can make changes if incompatibilities are found between input data and self-produced forms, we decided to disable this module. The reasons for doing this were the following: as D&K note, such a cranking module is quite alien to a cue-based learner, and should be theoretically unnecessary anyway. Furthermore, to assess the merits of the cue-based learning theory, it is important to get at the results of the Learner module directly, even if a subsequent module of the system might improve on them.

Before we go on to discuss the results, a word of caution concerning the artificial languages is in order here. D&K observe that all parameter configurations assign the same structure to monosyllables, and conversely, that monosyllables are thus compatible with all parameter settings. They further note that any disyllabic word is compatible with an impressive number of different parameter settings. This does not imply, however, that any set of disyllabic words is compatible with all those stress systems. The difference between two different systems may reveal itself only in particular weight combinations, and may be obscured in others. Thus, the more important issue is the question to what extent languages are distinguishable. Restricting the 216 'languages' to disyllabic words, only 8 of them uniquely determine a parameter set, in the sense that no other configuration generates the same surface forms. Incorporating all three-syllabic words raises this number to 70. For our experimental input, 168 of the 216 'languages' are compatible with only a single set of parameters. We will refer to those languages as the 'unique' languages. In the discussion of the learning results, we will distinguish between the complete set of languages generated in the way described above, and the 'unique' languages.

YOUPIE's learner module found the correct set of parameter settings for 130 out of the 216 'languages', i.e. for 60% of the languages the values of the parameters detected by the Learner were completely identical to the parameter settings used in generating the languages. Considering the fact that some languages could be generated by a number of different settings, an additional 43 languages were learned, in the sense that the program came up with a compatible parameter set. This raises the success rate to 80%. For the 'unique languages' the success rate was 126 out of 168, i.e. 75%. Although these success rates appear to be reasonably high, they should be qualified in the light of two important characteristics of the experiment: on the one hand, the input data were completely transparent; on the other hand, they were highly systematic in the sense that all potentially relevant oppositions were present. The fact that, even under these strong simplifications of the learning task, the system fails to account adequately for about one quarter of the data, raises serious doubts concerning the universal applicability of the learning theory under consideration.

Let us now consider the experimental results in some more detail. In Table 3 success scores for the individual parameters are given.

Table 3: Success scores for individual parameters

| Parameter | All Languages (N = 216) | | 'Unique' Languages (N = 168) | |
|---|---|---|---|---|
| | N | % | N | % |
| P1 | 206 | 95 | 158 | 94 |
| P2 | 184 | 85 | 148 | 88 |
| P3 | 195 | 90 | 166 | 99 |
| P4 | 186 | 86 | 152 | 91 |
| P5 | 180 | 83 | 166 | 99 |
| P6 | 198 | 92 | 168 | 100 |
| P7 | 204 | 94 | 156 | 93 |
| P8A | 156 | 72 | 138 | 82 |
| P8 | 188 | 87 | 155 | 92 |
| P9 | N/A | / | N/A | / |
| P10 | 216 | 100 | 168 | 100 |

From this table, it appears that success scores for the individual parameters are higher than the overall success rates, with the notable exception of the parameter P8A (extrametricality). In section 3.3, we pointed out that there were several problems associated with the operation of the cues for this parameter. These theoretical considerations now seem to be confirmed by the facts. Another observation to be made is that, although the parameters P3 and P4 (direction of foot construction and headedness within a foot) are set simultaneously, this does not entail equal success rates for both.

Perhaps the most striking result is that only a single parameter, P10, is learned consistently across all languages. For the 'unique' languages, the same is true for P6. All other cues, even those which are robust on theoretical grounds, fail a number of times. We hypothesize that this is due mainly to their reliance on incorrect values for previously set parameters. To verify this claim, consider Table 4 below. In this table, the first and second columns indicate how many errors were made for each parameter. Next, in the column 'Previous Errors', the number of errors on previously set parameters is given. Only those parameters are considered whose value is used (directly or indirectly) by the cue for the parameter in the first column. In the final column ('Remaining Errors') the number of errors is given for the case where all previously set parameters were correctly set.

Table 4: Errors in setting dependent parameters

**All Languages**

| Parameter | Number of Errors | Previous Errors | Remaining Errors |
|---|---|---|---|
| P1 | 10 | P2: 8<br>P8A: 4 | 2 |
| P2 | 32 | P8: 4<br>P8A: 18 | 14 |
| P3 | 21 | P2: 2<br>P8: 8<br>P8A: 15 | 0 |
| P4 | 30 | P2: 12<br>P8: 11<br>P8A: 26 | 0 |
| P6 | 18 | P5: 18 | 0 |
| P7 | 12 | P1: 8 | 4 |

**'Unique' Languages**

| Parameter | Number of Errors | Previous Errors | Remaining Errors |
|---|---|---|---|
| P1 | 10 | P2: 8<br>P8A: 4 | 2 |
| P2 | 20 | P8A: 10 | 10 |
| P3 | 2 | P8: 1<br>P8A: 2 | 0 |
| P4 | 16 | P8: 4<br>P2: 8<br>P8A: 12 | 0 |
| P7 | 12 | P1: 8 | 4 |

The figures in Table 4 give a good impression of the extent to which errors made by the Learner are caused by the interdepencies between parameters. When we take the subtable with the figures for all languages, it becomes clear that P3, P4 and P6 involve errors only when previous parameters are incorrectly set. In other words, the cues for these parameters are well-defined, but are not robust enough to defeat the influence of incorrectly set parameters on which they depend. P1, P2 and P7 undergo the same influence, although here we find a number of remaining cases where all previous parameters were set correctly. These errors therefore are due to malfunctioning of the cues associated with these parameters.

Of course this experiment is only a first attempt at testing YOUPIE and further experimentation is needed to establish the exact nature of the data and/or parameter interactions that lead to erroneous settings. Pinpointing exactly where and why the system errs calls for a more thorough investigation of the system's operation than mere inspection of its output can provide. However, two lessons can be learned even from this small experiment: (i) extensive testing with synthetic data may reveal substantial gaps in the empirical coverage of the model; (ii) cues which are theoretically robust when considered in isolation, may well break down in their actual operation due to interactions between different parameters.

## 5. Conclusion

Dresher & Kaye (1990) present a computer program, YOUPIE, which purports to model a particular aspect of language acquisition, viz. the acquisition of stress systems. They ascribe explicitly to the "Chomskyan" paradigm, in the sense that their program incorporates the view that the acquisition of (subparts of) the grammar of a language amounts to determining the appropriate values for a set of parameters provided by Universal Grammar. Although this view has gained widespread acceptance in the language acquisition scene, and has been the guiding assumption underlying various psycholinguistic research efforts, YOUPIE is one of the few attempts to incorporate the theory in a computer program, and as such deserves the attention of students of language acquisition. The demands a similar enterprise imposes are extremely high: every single detail of the theory needs to be spelled out explicitly in order for the program to run. This is especially the case whenever mechanisms are called upon which are usually glossed over in theoretical work. Consequently, the strengths as well as the major weaknesses of the theory will show up in a simulation such as YOUPIE.

D&K’s paper provides a detailed description of the model. Major emphasis is placed on the necessity of a learning theory, an explicit way of relating the data given to a learner to the parameters to be set. They propose a specific format for such learning theory, viz. a cue-based model. Cues are specific patterns in the input which act as triggers for determining parameter values. In this cue-based model, every individual cue must obey the requirements of appropriateness and robustness. Moreover, the system as a whole is required to proceed in a deterministic fashion, i.e. to make decisions on the basis of positive evidence that is strong enough to guarantee that, once a decision is reached, it should not be reconsidered afterwards.

At the heart of the model lies the assumption that the parameters provided by Universal Grammar give the learner a serious headstart: in the process of acquisition, the learner will not entertain any useless hypotheses regarding the grammar to be acquired, because the parameters delineate the (very restricted) space of possible grammars. The cue-based learning theory is meant to bridge the gap between the data and the grammar to be acquired. Cues should enable the learner to select in a straightforward manner the grammar to be acquired. As such the theory is very attractive. In the words of Fodor (1989: 133): "... parameter theory presupposes a great deal of innate mental programming, with essentially the whole grammar mapped out in advance. But what it gets in return for this is an extremely simple learning mechanism: a mere switch setter, with the switches tripped by just a handful of observations about the input." The crux of the model, then, is the formulation of a set of parameters and a collection of appropriate cues so that with "just a handful of observations" the parameters can be assigned their correct value.

In this paper, we investigated to what extent D&K's model succeeded in attaining precisely this goal. YOUPIE's merits were assessed from two different angles: on the one hand, the actual implementation of the cue-based learner was analyzed and judged against the requirements that D&K pose. On the other hand, an experiment was set up to test the program's empirical coverage.

To start with the former, we found that in formulating the cues, much more was needed than positive evidence: some cues rely heavily on indirect negative evidence, whereas others involve extensive cross-word comparisons and multiple ways of recoding the input material. Furthermore, a rather elaborate ordering of the parameters was set up so that parameter settings could be made dependent upon the value of previously set parameters. Finally, exhaustive search was used in restricted parts of the solution space. Thus, what Fodor refers to as "an extremely simple learning mechanism" turns out to be an extremely complex mechanism that is only simple in principle. Apart from the complexity issue, we indicated a potential source of problems in relation to the above mechanisms. Both the use of indirect negative evidence and extensive cross-word comparisons in the form of consistency checks require all relevant data to be present when learning is initiated. In the context of a deterministic learner, oversights in the data collection phase may lead to wrong initial decisions, with potentially far-reaching effects whenever later decisions are dependent upon them. Consequently a similar learner which can in principle attain its target, crucially depends on the availability of an encompassing data set. The mechanisms it incorporates cannot be transferred to an incremental learner (or a learner that works on several consecutive batches) without altering it in a substantial way (e.g. by removing the determinism requirement).

Examining the requirements that should be met, it was indicated that neither appropriateness, robustness, nor determinism were fully adhered to. Appropriateness was violated in several respects: in the formulation of several cues, the model resorts to concepts (such as the number of syllables, weight strings and stress strings) which lack any independent status in metrical phonology. Certain cues crucially depend on values which belong to distinct metrical domains, etc. Moreover, the mechanisms of cross-word comparison, extensive computation required for recoding the input materials as well as consistency checks are clearly in contradiction with the appropriateness requirement.

Robustness was not met in several respects. First of all, the requirement that the input material be completely transparent renders the criterion almost vacuous if robustness is taken to mean that "core parameters must be learnable despite disturbances to the basic patterns caused by opacity, language-particular rules, exceptions, etc." (D&K, p. 156). Second, even under this considerable simplification of the learning task, certain parameters were simply not correctly set in our empirical test. Hence, the associated cues are not robust in that they failed to handle the transparent input materials adequately.

Determinism was observed in the Learner module, though the existence of the Cranker seriously compromises this achievement.

Many of these theoretical shortcomings were reflected in the empirical test of the model. Taking for granted that the set of parameters in YOUPIE's UG is still incomplete and defective, in the sense that it does not cover the full range of observed stress systems, we generated artificial languages encompassing all stress systems the current set of parameters allows. These languages thus provide fully transparent learning material that can be completely accounted for by the parameters represented in YOUPIE (due to the simple fact that the parameter set itself was instrumental in constructing the learning materials). Under these conditions, the system could reasonably be expected to detect the parameter settings of the languages that were initially generated by them. It was shown that the Learner was indeed successful in accounting for a large portion of those languages, though coverage was by no means complete. More precisely, 75 to 80% of the languages were assigned a correct grammatical description. A second result of our empirical test was that a number of cues were fairly accurate, but that others were extremely susceptible to the (in-)correct setting of previous parameters, even if the cue in itself was theoretically robust.

Summing up, the difficulties encountered in formulating appropriate and robust cues, combined with the simple fact that some parameters are still incorrectly set, even if all disturbing factors are eliminated from the input, cast some doubt upon the efficiency of the proposed learning theory. The question is whether a mechanism of parameter setting can be found that is able to perform its core function correctly without ignoring the fact that every language eventually brings along a considerable amount of noisy data (language particular rules, plain exceptions, etc.) To what extent such a system will be able to retain the characteristics of a cue-based learner such as the one that D&K envisage remains to be seen.

## References

Archibald, J 1993. *Language learnability and L2 phonology. The acquisition of metrical parameters.* Dordrecht: Kluwer Academic Publishers.

Chomsky, N. 1981. Principles and parameters in syntactic theory. In Hornstein, N. & Lightfoot, D. eds. *Explanations in linguistics.* London: Longman. 32-75.

Chomsky, N. 1986. *Knowledge of language: Its nature, origin and use.* New York: Praeger.

Daelemans, W., Gillis, S. & Durieux, G. 1994. The acquisition of stress: A data-oriented approach. *Computational Linguistics* 20: 421-451.

Dresher, E. & Kaye, J. 1990. A computational learning model for metrical phonology. *Cognition* 34: 137-195.

Dresher, E. 1992. A learning model for a parametric theory of phonology. In Levine, R. ed. *Formal grammar: Theory and implementation.* Oxford: Oxford University Press. 290-317.

Fikkert, P. 1994. *On the acquisition of prosodic structure.* Dordrecht: ICG.

Fodor, J. D. 1989. Learning the periphery. In Matthews, R. & Demopoulos, W. eds. *Language learnability and linguistic theory.* Dordrecht: Kluwer. 129-154.

Gibson, E. & Wexler, K. 1994. Triggers. *Linguistic Inquiry* 25: 407-454.

Giegerich, H. 1985. *Metrical phonology and phonological structure.* Cambridge: Cambridge U.P.

Goldsmith, J. 1994. Grammar within a neural network. In Lima, S., Corrigan, R. & Iverson, G. eds. *The reality of linguistic rules.* Amsterdam: Benjamins. 95 - 113.

Gupta, P. & Touretzky, D. 1994. Connectionist models and linguistic theory: Investigations of stress systems in language. *Cognitive Science* 18: 1-50.

Halle, K. & Vergnaud, J.-R. 1987. *An essay on stress.* Cambridge: MIT Press.

Hayes, B. 1987. A revised parametric metrical theory. *NELS* 17: 274-289.

Hayes, B. 1981. *A metrical theory of stress rules.* MIT Doctoral Dissertation.

Kager, R. 1989. *A metrical theory of stress and destressing in English and Dutch.* Dordrecht: Foris.

Kager, R. 1995. The metrical theory of stress. In Goldsmith, J. ed. *A handbook of phonological theory.* Oxford: Basil Blackwell.

Liberman, M. 1975. *The intonational system of English.* Doctoral Dissertation. MIT.

Nouveau, D. 1994. *Language acquisition, metrical theory, and optimality.* Utrecht: OTS publications.

Trommelen, M. & Zonneveld, W. 1989. *Klemtoon en metrische fonologie.* Muiderberg:

Coutinho.

Valian, V. 1990. Logical and psychological constraints on the acquisition of syntax. In Frazier, L.& De Villiers, J. eds. *Language processing and language acquisition.* Dordrecht: Kluwer. 119-145.

Wexler, K. 1990. On unparsable input in language acquisition. In Frazier, L.& De Villiers, J. eds. *Language processing and language acquisition.* Dordrecht: Kluwer. 105-117.