PARSYL:
A Computer Model of Syllable Parsing and Acquisition

Barbara Brunson
Department of Linguistics
University of Toronto

1. Introduction

In this paper, I describe PARSYL\(^1\), a system designed to parse syllable structure and acquire some of the language-specific properties of syllable well-formedness on a given set of data, including phonotactic constraints within the domain of a single syllable. PARSYL assumes an independence of syllable structure acquisition from the acquisition of phonemes, phonological rules, stress, and the like. This is certainly an idealization of a child's actual acquisition experience. However, it is a necessary idealization from the point of view of the construction of a working system within a limited domain. Even making this simplifying assumption, that syllable structure is an independent, closed system, we can, nevertheless, develop a fairly robust system that can be used on large data sets to test predictions and locate data that are problematic to the theory\(^2\).

The paper will consist primarily of discussion of various aspects of PARSYL and implications of this system to claims made in the literature on syllable

\(^1\) PARSYL is implemented in LPA micro-PROLOG Professional (version 1.2) on an IBM AT.

\(^2\) The ease with which a linguistic theory can be implemented on a computer should not be considered a necessary criteria for a correct theory. It is entirely possible that current computer architecture is ill-suited for modelling theories about cognitive organization. However, when an implementation is possible within a limited domain, it does provide a valuable diagnostic utility to check the theory. An implemented system can test interactions of principles and rules. It can confirm correct interactions, and, at times, it can even detect unexpected interactions (whether they be correct or not).

To be useful to a linguist, the implementation should be true to the theory -- but in order to attain this, the theory must be sufficiently precise within a clearly-defined limited domain so that the parser can be implemented without needing to resort to unmotivated assumptions that are external to, or not supported by, the theory. In addition, the parser should be readily modifiable as the theory changes. This requires that the modularity of the implementation reflect the modularity of the theory. PARSYL has been designed with these considerations in mind. In particular, the implementation of distinctive features is a separate module that can be substituted with a more up-to-date theory of features. In addition, the parameters of acquisition are all implemented in a standard format, allowing further parameters to be added, as well as allowing selected parameters to be deactivated. This allows for the testing of implications and interactions of specified subsets of parameters.
structure and on parsing. This is followed by an appendix of screen images showing PARSYL in action.

2. Syllable Parsing and the Grain of Determinism

I accept the general syllable structure presented in (1) (Selkirk (1982), inter alia).

(1)

\[ \sigma \]

\[ \text{onset} \quad \text{rhyme} \]

\[ \text{nucleus} \quad \text{coda} \]

Syllable parsing involves the assignment of syllable structure to a sequence of segments. This will involve left to right processing of the input sequence. I assume that syllable parsing is strictly deterministic\(^3\) (in the sense of Marcus (1980)). Within some limited look-ahead buffer, the structure of a syllable must be assigned without back-tracking or pseudo-parallelism.

I also assume that the acquisition of syllable structure parameters is deterministic. That is, each modification of the initial grammar's parameter settings is a local, immediate decision that is irreversible. In addition, the cues that the parser uses to change parameter settings are appropriate to the parameter that is involved (see also Drescher and Kaye (1987)). The data is provided one item at a time, and the new grammar is acquired as this data is presented, incrementally. No ordering of the data is required. This approach necessitates that parameters be changed in a conservative fashion, only on the basis of positive evidence in the data.

2.1. Defining a Buffer

In syntactic parsing, the buffer is assumed to be 3 syntactic constituents in length (Marcus (1980)), where a syntactic constituent is either a word or an already-parsed phrasal constituent. The choice of the buffer length is somewhat arbitrary - the main argument for 3 as a buffer length is that, in the vast majority of cases, it works. In any system that deals with a larger perspective

\(^3\) Cairns and Feinstein (1982) propose a non-deterministic approach to syllable parsing. They propose that a set of alternative syllable parses are constructed for a sequence, with markedness indices being computed to select the most highly valued syllabification from this set.
on language rather than just limiting itself to the syntax, if we want strict
determinism to hold of the entire system rather than just within given modules,

it would follow that we must interleave the parsing of the different domains of
linguistic structure. That is, in a strictly deterministic system, one could
not, for example, parse the syntax and pass the result off to a
pragmatic/semantic processor to be further analyzed, since semantic information
is often required in order to decide on a syntactic analysis. Instead, the
various linguistic domains would need to be interleaved together, so that each
component can contribute the information required of it to obtain a correct
analysis deterministically.

I believe that the key to extending deterministic parsing to handle a
larger range of linguistic phenomena lies in the definition of the buffer
length. Intuitively, we can nest linguistic domains by the type of constituents
that they deal with. For example, morphology deals with word-internal structure
and, hence, logically nests within the syntax, which deals with supra-word
structure. Similarly, syntax defines clausal constituents which are the building
blocks of discourse.

From the point of view of parsing, if we were to design a morphological
parser that worked within a syntactic parser, we would need to provide the
morphological parser with limited access to the syntactic parser's current state.
Consider the following two sentences.

(2) He never talks about his likes and dislikes.
(3) He likes to jog on Sunday mornings.

In order to provide a morphological analysis of likes, we need to know whether
it is being used syntactically as a noun or as a verb in the sentence. Most
parsers deal with such situations by constructing all possible morphological
structures of a word and allowing the syntactic parser to select among them.
However, this violates the principle of strict determinism, since structure is
constructed that is not used in the final analysis. In order to have a
deterministic morphological parse, it must have access to the syntax in some,

albeit limited, way. The buffer provides us with such a device. Suppose the
morphological parser buffer did not have a static size, as is claimed to be the
case for syntactic parsers. Suppose, instead, that the buffer is dynamically
defined in terms of higher structure under construction, namely syntactic
structure. In this way, the morphological parser could have access to the
syntactic parser states. Certainly, the morphological parser need not have
access to the entire syntactic buffer, it need only access the constituent
currently under construction. It is not obvious, of course, if there exists an
interleaving which would permit strict determinism of every level. I will use
the term "grain" of determinism to indicate the relevant buffer constituents and
size. So, for example, morphological parsing is finer grained than syntactic
parsing, with the buffer size being projected from syntactic considerations, and
buffer contents being morphemes or preconstructed pieces of words.
Similar arguments can be raised that the syntactic three-position buffer is derivative of prototypical predicate argument/thematic structures\textsuperscript{4}. A complete discussion of this is beyond the scope of this paper.

Extending these sorts of arguments to the domain of syllable parsing should help to clarify the particular implementation issues surrounding determinism within this component. In linguistic structure, syllables nest within prosodic foot structure. A prototypical prosodic foot - a binary foot - provides a good working hypothesis regarding the size of the syllable parser buffer. Note that using prototypical higher level structure rather than actual higher-level structure is necessitated when the parser does not actually construct the higher structure.

When parsing inter-syllabic consonant clusters, one needs to see the entire cluster in order to make the appropriate syllable division. This can be accomplished by projecting the buffer to include the second syllable peak - all inter-peak consonants will then be simultaneously within the view of the buffer. The buffer size, then, is justified in terms of prototypical higher structure, as well as pragmatically -- it works.

3. Calculating the Buffer Size

In calculating the size of the buffer, PARSYL employs a procedure to locate the syllabic peaks. The peaks are located by scanning the sequence of sounds from left to right, counting as a peak any local maximum in terms of major sonority differences in adjacent sounds. This has the advantage of being able to deal with non-vocalic peak -- that is, the use of sonorants as syllable peaks. Typically, the peak locating function must scan one segment beyond what will be set as the buffer in order to determine if a sonorant is a local maximum. With each successive syllable that is parsed, the buffer is reset, in effect stepping left-to-right across the word.

In some analyses (for example, Levin (1985)), the syllable nucleus can be specified in the underlying lexical representations. Notice, however, that Levin's claim is that, in the underlying representation, syllable nuclei can be specified since there are contrasts in syllabification that could otherwise not be dealt with on the basis of segmental evidence alone. The task of PARSYL is not to assign syllable structure to underlying forms, but rather to determine the syllabification of surface phonetic forms. PARSYL does not pre-compile syllable nuclei -- it works with segmental information only and computes the location of syllable nuclei.

4. Syllable Well-Formedness

PARSYL defines an acceptable syllabification of a word as a sequence of valid syllables, each constructed within the view of the buffer. The syllabification must include in the analysis all of the segments in the sequence (cf. Chomsky's (1986) principle of Full Interpretation). At the end of each

\textsuperscript{4} See also Brunson (1988) and Pritchett (1988) for discussion of the relevance of thematic information in deterministic syntactic parsing.
successful assignment of structure to a syllable, the buffer is reset to include the next sonority peak.

A well-formed syllable consists of a valid onset followed by a valid rhyme, where the rhyme is a valid nucleus followed by a valid coda.

4.1. Word-edge Syllables
4.1.1. Extrasyllabicity

Word-initially, anything that would count as an acceptable coda before the onset is taken to be extrasyllabic (Steriade (1982)). Since onsets are maximized\(^5\), the word-initial extrasyllabic sequence, being a coda, is minimized.

Word-finally, anything that would count as a valid onset after a valid coda is taken to be extrasyllabic (Carlyle (1988), Borowsky (1986)). Since onsets are maximized, however, here we find that word-final extrasyllabic sequences are also maximized. Intuitively, this seems to result in too many extrasyllabic final sequences. This may be fine theoretically, or we may wish to shut off maximization of onsets in the case of word-final extrasyllabic "onsets". One good result of this from an acquisition point of view is the avoidance of acquiring larger and larger coda sizes on the basis of word-edge-information, without the need to bypass the acquisition processes. By maximizing extrasyllabic "onsets" word-finally, we can preserve a conservative learning strategy for coda sequences without requiring that word-final codas be treated as exceptional in this regard. Effectively, it treats word edges as if they were sonority peaks, anchoring an extrasyllabic sequence of consonants, either as "coda" (word-initially) or "onset" (word-finally) sequences. The current implementation does not acquire any onset constraints on the basis of these maximized word-final "onset" sequences. This violates the general principle that I have been employing that these "onsets" should really not be treated as

---

\(^5\) In the sequence \([xyz]\), where \([xy]\) is a valid coda and \([yz]\) is a valid onset, then the coda is set to \([x]\) and the onset is set to \([yz]\). A number of syllable parsing strategies have made use of the principle of maximization of onsets (Kahn (1976), Lowenstamm (1981), Steriade (1982)). Kahn's approach involved scanning for vowels and first assigning all pre-vocalic consonants to the syllable (subject to restrictions governing possible clusters), then assigning any left-over consonants as post-vocalic consonants (subject, again, to restrictions on possible clusters). By assigning pre-vocalic consonants to the syllable first, Kahn effectively maximizes onsets (although he did not make use of a hierarchical syllable representation). Steriade (1982) follows Kahn -- maximizing onsets by assigning the onset to the syllable prior to the coda.

Lowenstamm (1981), on the other hand, makes maximization of onsets an explicit requirement of a well-formed syllabification, rather than a consequence of a parsing strategy.

My approach employs a parsing strategy similar to that of Kahn and Steriade. Given a sequence of inter-vocalic consonants entirely within the view of the buffer: \(C_1 C_2 \ldots C_n\), for \(i\) ranging from 1 to \(n\), the first sequence \(C_1 \ldots C_n\) that is a valid onset is assigned as the onset (subject to coda conditions on \(C_1 \ldots C_{i-1}\)). By considering the onset prior to the coda, PARSYL implements maximization of onsets.
exceptions. In practice, I have not come across any data sets where it would make a difference to the grammar that is acquired. The question remains, does a language ever employ final extrasyllabic sequences (not counting appendices, see section 4.1.2) that would not count as valid onsets in non-word-edge syllables? The conservative implementation allows us to temporarily ignore this question. If extrasyllabic "onsets" are, indeed, subject to exactly the same constraints as word-medial onsets, then the defining principles for valid onsets can be acquired on the basis of word-medial onsets, if enough data is provided, without ever overtly making use of the information from the extrasyllabic "onsets". The only disadvantage is that this could require more data to acquire onset constraints than if it also took the extrasyllabic "onsets" into consideration. If, on the other hand, extrasyllabic "onsets" are not subject to the same constraints as word-medial onsets, then the strategy employed in PARSYL is the only correct way to implement this independence. Of course, this still begs the question of just what it is that constrains what can count as an extrasyllabic "onset". I suspect that there really is no empirical reason for making any distinction at all between the word-final "onsets" and word-medial onsets -- the implementation subsumes this possibility in a conservative fashion that should avoid any problems for deterministic parsing.

4.1.2. Appendices

In addition to word-final extrasyllabic sequences, a word-final syllable may have an appendix. An appendix is assumed to be a third rhyme constituent in the display of the syllable parse, but there is no theoretical claim being made by this constituency. It could equally well be displayed as adjoined to the entire syllable rather than within the rhyme of the syllable. An appendix is assumed to be a sequence of consonants between a well-formed coda and onset (extrasyllabic) in the word-final syllable. Clearly there are sonority constraints on what can occur in an appendix, but the consensus in the literature is not sufficiently precise for implementation. The approach taken in PARSYL is to grossly over-accept appendices during the acquisition stage of the parse and, at the same time, acquire fine-scale sonority templates of the appendices. When acquisition mode is shut off, these templates define valid appendices. (See sections 5 and 6 for discussion of these points.)

4.2. Word-medial Syllables

Word-medial syllables do not allow appendices, or extrasyllabic sequences. Word-internally, all segments must be analyzed into the basic syllable structure consisting of an onset, nucleus and coda.

Valid onsets are sequences of sounds that are strictly increasing in sonority. Valid codas are sequences of sounds that are strictly decreasing in sonority. Valid nuclei are sequences (up to a maximum of two) that are either level in sonority or decreasing in sonority, according to a separate scale of nucleus sonority that sets sonority as an inverse of vowel height ( Carlyle (1988), Levin (1985)). The feature "syllabic" is not used. Vowels and glides are all [-consonantal]. The feature "vocalic" is used, but the parser does not use it crucially to distinguish vowels from off-glides. Any vowel, provided
sonority conditions are met, can be analyzed as an off-glide. The feature "vocalic" is primarily used so that PARSYL is compatible with alternative transcriptions that use y and w as off-glides distinct from their corresponding vowels.

Two separate parameters are used for level sequences and long vowels in order to distinguish between languages that allow only long vowels and those that also allow non-identical vowel sequences in one nucleus where there is no sonority difference between the two vowels. There is probably a dependency between these parameters that PARSYL does not overtly capture: it is probably the case that if a language does allow non-identical vowel sequences with level sonority within a single nucleus, it will also exhibit long vowel sequences. PARSYL, by keeping these two parameters independent, predicts that languages should allow level (non-identical) nuclear sequences without permitting long vowels. If this implication is correct, then it should be the case that if enough data were provided, the system would eventually encounter both level (non-identical) nuclear sequences and long vowels and, hence, set both of the parameters correctly. So, once again, the implementation subsumes a possible theoretical implication while maintaining a conservative approach that, although requiring more data in the acquisition stage, will eventually reach the correct parameter settings. The disadvantage remains that PARSYL doesn't narrowly define and enumerate only possible human syllabification systems if this implication is true. Ideally, if the implication is true, then it should follow from something else in the theory of syllabification -- this, once again, is an area that the literature on syllabification offers no consensus and little discussion that I am aware of.

5. Sonority Hierarchies and PARSYL

In recent theories of syllable parsing, a universal sonority hierarchy is proposed (Carlyle (1988), Levin (1985), Selkirk (1982), Steriade (1982)). Sonority distances are calculated on this hierarchy and minimum distances (Minimum Sonority Distance, henceforth MSD) are computed for a given language (Steriade (1982)). Parts of the sonority hierarchy seem to be subject to language-specific setting, while other parts are universally set. From an acquisition point of view, this theory results in some rather complex and circular interactions.

Assume that maximizing onsets is a universal strategy and not subject to language particular variation (compare Carlyle (1988)). The universal sonority hierarchy is claimed to be as follows, where the left-hand value is greater in sonority than the right-hand value (Carlyle (1988), Levin (1985)):

---

It has been suggested that a homorganic vowel parameter would be more in keeping with the treatment of consonant clusters that is adopted here. This has not yet been implemented. I wish to thank Karen Carlyle (personal communication) for this suggestion.
Now consider what would happen if PARSYL were to try to acquire the MSD for a language given this scale, together with maximization of onset. Suppose a monomorphic item like [alra] were encountered. According to the sonority scale above, [lr] is a sequence with increasing sonority ([l] lateral is more sonorant than [+ lateral]). Maximizing onsets would then predict that [lr] should be syllabified into a single onset, and the MSD would be set sufficiently small to allow any sequences as close in sonority as [lr] to occur tautosyllabically. This could have serious ramifications since suddenly we would find that all clusters that are as close in sonority as [lr] would be accepted in the onset. In order to implement syllable parsing acquisition, given these assumptions, we would need to initialize the MSD as the maximum possible value, and have the setting of this parameter interact with the maximization of onsets — perhaps allowing maximization of onsets to turn off, all else being equal, to preserve a higher MSD.

In PARSYL, a somewhat different approach is taken. PARSYL has two levels of sonority checking. There is a processing level of large sonority differences that is checked for each onset and coda. This level essentially encodes the top part of the universal sonority hierarchy (down to and including the feature "sonorant"). In addition, there is a template level of finer sonority differences. This encodes the entire hierarchy — assigning a separate slot in an ordered list of distinctly defined sonority values. These slots are numbered\(^7\), and (pairwise) each tautosyllabic cluster is assigned the pair of sonority slot values that they match to in this scale. This pair is added to

---

\(^7\) The current implementation has the following values:

<table>
<thead>
<tr>
<th>feature(s)</th>
<th>slot value</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-cons]</td>
<td>1</td>
</tr>
<tr>
<td>[+cons;-nas;-lat]</td>
<td>2</td>
</tr>
<tr>
<td>[+cons;-nas;+lat]</td>
<td>3</td>
</tr>
<tr>
<td>[+cons;+nas]</td>
<td>4</td>
</tr>
<tr>
<td>[+cons;+cont]</td>
<td>5</td>
</tr>
<tr>
<td>[+cons;-cont;+voice]</td>
<td>6</td>
</tr>
<tr>
<td>[+cons;-cont;-voice]</td>
<td>7</td>
</tr>
</tbody>
</table>

This essentially ranks the leaf-level elements in the sonority hierarchy.

The features and values used here are readily modifiable to implement and test alternative hypotheses.
the templates assertion for the particular type of cluster (onset, coda or appendix). So, for example, if an onset sequence [f1] is encountered, template slots 5 for [f] and 3 for [l] are matched, and the onset template [onset-template 5 3] is asserted. If a coda sequence [ns] is encountered, template slots 4 for [n] and 5 for [s] are matched, and the coda template [coda-template 4 5] is asserted. When the parser is actively acquiring syllabification, it accumulates templates for the onset, coda and appendix. When acquisition mode is turned off (see section 6), the existing set of templates define syllable well-formedness.

MSD values can be inferred from these templates as the minimum difference between the pairs of numbers. This can be done deterministically across the data -- with each successive template that is acquired, it can simply compare the numerical difference in the template to the current MSD -- if it is numerically smaller, we revise the MSD; otherwise, it isn’t changed. By initializing the MSD to an arbitrary yet sufficiently large value, the MSD can be revised on the basis of positive evidence only.

The approach that I have adopted will run into difficulty when presented with a language that has a very small MSD and does allow tautosyllabic sequences of sonorants or non-sonorants. For implementation purposes, I suspect that the best approach would be to have the parser wait until such time as it encounters such a sequence with no other option but to make it tautosyllabic. It then could invoke secondary sonority scales, as necessary, to flesh out more fully the universal sonority hierarchy at the processing level. This has not yet been implemented.

There is another problem of a similar nature. As already mentioned, the current implementation of PARSYL takes a fairly narrow focus and does not make reference to other sources of evidence that bear on syllable structure assignment (eg. stress). So, for example, PARSYL will acquire that a language has branching onsets if presented with the form [abrad], given sonority considerations and maximization of onsets. Apparently, such forms do exist in languages that have maximally CVC syllables, with the syllable boundary occurring between the [b] and the [r]. PARSYL will not acquire the correct parameter settings for such languages, although it will correctly parse the syllable structure of such languages if the parameter settings are given and acquisition mode is turned off. This sort of evidence suggests that an even more conservative learning strategy is called for. Again, the problem seems to rest with the maximization of onsets strategy. If PARSYL used maximization of onsets to acquire complex onsets only if the current grammar cannot assign a structure to the sequence, then given a grammar that allows CVC syllables, it would not acquire complex onsets in this case. One drawback to this approach is that it imposes an ordering on the data -- the system must already have acquired that there are codas before it can assign any syllable structure to this sequence. Perhaps more serious is the apparent implication here that the syllabification [ab.rad] is somehow more basic than the syllabification that would maximize the onset. Once again, there is a tension that emerges between maximization of onsets and the conservative approach to acquisition that is necessitated by a deterministic, incremental

---

8 I wish to thank Elan Drescher for pointing out this problem.
model. I have not attempted to resolve these difficulties in the current version of PARSYL.

There are two versions of PARSYL with respect to the sonority templates. One version not only acquires the slot values, but also acquires the features "coronal" and "anterior", where appropriate, for each slot. The features "coronal" and "anterior" can be relevant for language-specific sonority considerations (Carlyle (1988)). The problem with this version arises when presented with relatively small test data sets for acquisition. Unless all possible combinations of values for "coronal" and "anterior" are encountered for each template, the parser will tend to under-accept when acquisition mode is shut off. For this reason, I have created a second version that does not acquire the features "coronal" and "anterior" at all. This allows the parser and templates to be tested while running on a smaller set of data.

In addition to subsuming MSD, the templates also subsume many phonotactic constraints in a language. Together with the homorganic clusters parameter, which is set depending on whether tautosyllabic adjacent homorganic clusters are allowed or not, most phonotactic constraints within the domain of a single syllable are covered. As a side note, in English we find sequences [tr] and [dr] together with [br], [gr], [pr], and [kr], but not *[tl] and *[dl], although there are sequences [bl], [gl], [pl], and [kl]. These facts follow from the homorganic cluster constraint by not specifying [r] for features "anterior" and "coronal", but fully specifying [l] for these features. In the version where templates are acquired with the values of "anterior" and "coronal", these facts would also be able to be accounted for, but even still, if [r] were specified for the features "anterior" and "coronal", it would incorrectly acquire that homorganic clusters were permitted tautosyllabically in English when presented with the [tr] or [dr] sequences.

9 This suggestion is due to Keren Rice (personal communication).

10 In the current implementation of PARSYL, a somewhat surprising problem emerged with respect to this proposal. The implementation of distinctive features marks only the "plus" values, leaving the "minus" values unmarked, but inferable. The parser is, therefore, unable to distinguish between [k], which is unmarked for "anterior" and "coronal" (and, hence, [- anterior], [- coronal]) and [r], which is underspecified for these values. As a result, when presented with [kr] sequences, the current version incorrectly (although correct, given the feature implementation) acquires that this is a homorganic cluster.

This is a new form of a familiar problem -- the problem of ternarity (see Stanley (1967), inter alia) in a binary feature system.

Clearly, the problem is with the implementation of features. The analysis of [r] with respect to homorganicity requires a more sophisticated theory of features for correct implementation than the current version of PARSYL provides. The analysis is essentially demanding a "place node" in the feature implementation to give the required contrast. There is, however, considerable current controversy over the correct formalism for theories of features, and this makes it difficult to settle into a particular implementation. The implementation of features in PARSYL has been set apart as a module of code that could readily be substituted for an alternative implementation without causing
6. **Parser Modes**

The parser has two modes. The modes are defined by the predicate ACQUIRE. When ACQUIRE binds its argument to "t", then acquisition mode is set on, and the parser will acquire the particular syllable structure constraints that are needed to account for the data presented to it. When ACQUIRE binds its argument to "f", then acquisition mode is set off, and the parser will then only succeed in parsing a syllable if the current parameter values and templates can correctly account for this syllable. When acquisition mode is shut off, all of the "check and acquire" routines become static tests of well-formedness against the current parameter setting.

One interesting application of this feature is to acquire parameters for a small set of data, and then shut off the acquisition mode to test the compatibility of other words with the grammar that has been acquired.

7. **The Parameters**

The parser has 11 parameters with initial values as follows:

<table>
<thead>
<tr>
<th>PARAMETER NAME</th>
<th>INITIAL VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-non-vocalic-peaks</td>
<td>0</td>
</tr>
<tr>
<td>P-branching-nuc</td>
<td>0</td>
</tr>
<tr>
<td>P-length-of-onset</td>
<td>1</td>
</tr>
<tr>
<td>P-length-of-coda</td>
<td>0</td>
</tr>
<tr>
<td>P-length-of-rhyme</td>
<td>1</td>
</tr>
<tr>
<td>P-homorganic-cluster</td>
<td>0</td>
</tr>
<tr>
<td>P-level-nuclear-seq</td>
<td>0</td>
</tr>
<tr>
<td>P-long-vowels</td>
<td>0</td>
</tr>
<tr>
<td>P-appendix</td>
<td>0</td>
</tr>
<tr>
<td>P-initial-extrasyll</td>
<td>0</td>
</tr>
<tr>
<td>P-final-extrasyll</td>
<td>0</td>
</tr>
</tbody>
</table>

With the exception of the "length-of" parameters, the value 0 means that the language does not exhibit the property in question, while the value 1 means that the language does exhibit that property. The three "length-of" parameters are set to the actual numerical maximal value that has occurred for that particular constituent.

The initial values of the "length-of" parameters are set so that the initial expectation is for the least marked CV type syllable (Kaye and Lowenstamm (1981)). The P-length-of-rhyme parameter is independent of the P-length-of-coda parameter. If a language allows either a branching nucleus or a coda but not both, then the parameter values would be set as follows:

---

serious repercussions for the rest of the implementation.
P-length-of-rhyme  2
P-length-of-coda    1
P-branching-nuc    1

What this means is that the rhyme can have at most two leaf-level things in it. It can have a branching nucleus, which would fill the possible elements in the rhyme, or it could have a coda, but then the length of the coda together with the length of the nucleus must be less than 2 - hence there cannot be both a branching nucleus and a coda in a language with these final settings.

The length-of-onset parameter is increased only with word-internal onsets. This is a conservative approach due to the minimization of word-initial extrasyllabic "coda" sequences caused by the maximization of onset principle. Word-final codas do affect the length-of-coda parameter since they are minimized by the same principle. This may need to be revised, subject to empirical evidence.

The parameter P-non-vocalic-peaks is initialized at 0, (i.e. no non-vocalic peaks). This is set to 1 if other sonorants can be syllable peaks.

The parameter P-branching-nuc is initialized to 0, (i.e. no branching nuclei). This is changed to 1 when a sequence of vowels with falling sonority or a sequence of vowel plus glide is encountered.

The parameter P-homorganic-cluster is initialized to 0 (i.e. no tautosyllabic sequences of adjacent homorganic consonants). This is changed to 1 when such a sequence is found in either the onset or coda.

The parameter P-level-nuclear-seq is initialized to 0 (i.e. no sequences of non-identical vowels with the same vocalic sonority are allowed in a single nucleus (eg. *[Nuc e o])). This is changed to 1 when such a sequence is encountered.

The parameter P-long-vowels is initialized to 0 (i.e. no long vowels), but it is changed to 1 when two adjacent identical vowels are encountered. The current version of PARSYL does not include reference to morpheme boundaries, hence apparent OCP violations at morpheme junctures are currently not-handled. Regardless of the morphological environment, two identical vowels in a sequence will be treated as a long vowel. It actually would violate the general premise of the system that it deal with only audible surface phenomena if we annotated the input with morpheme boundaries. Again, the problem here is the assumption that we have an independence of other sub-systems of grammar. Clearly, if we embedded PARSYL in a more complete linguistic analysis that included morphological processing, the required information would be readily available. Annotating the input, in lieu of a fuller implementation that would include actual morphological processing, is acceptable if the status of the annotations is clearly understood.

The parameter P-initial-extrasyll is initialized to 0 (i.e. no word-initial extrasyllabic sequences). This is revised to 1 if an initial extrasyllabic sequence is encountered. Similarly, the parameter P-final-extrasyll is initialized to 0, but can be revised to 1 if a word-final extrasyllabic sequence is encountered.
The parameter P-appendix is initialized to 0 (ie. no appendix in the word-final syllable). This is revised to 1 if a word-final syllable is encountered that does have an appendix.

8. Conclusions

PARSYL is a system that both parses syllable structure and acquires the various syllable well-formedness parameters for a set of data. Most of its design has been motivated by considerations of linguistic theory, and some limitations and lack of adequate specification in the theory have been identified. Some of the shortcomings of PARSYL have been discussed and extensions suggested. Despite these shortcomings PARSYL is a fairly robust system and can acquire the required parameters of a wide range of syllable structure possibilities. Since PARSYL does quite strictly adhere to the linguistic theory of syllable parsing, wherever possible, it provides a valuable test of the theory. Effort has been made to mirror the modularity of the theory in the implementation. Aspects of the implementation are readily modifiable to test and compare alternative hypotheses and theories. Although PARSYL will break down where the theory lacks mechanism to deal with a phenomena, I believe this is far better than to have the implementation impose ad-hoc changes on the theory. In this way the parser is constrained by the theory and can be used as a diagnostic tool on the theory.

Acknowledgements

I would like to thank Peter Avery, Karen Carlyle, Elan Drescher, Geoff Loker, and Keren Rice for helpful discussion and comments. Much of the design of this system is inspired by the YOUPAK implementation (see Drescher and Kaye (1987)). This work was supported by a fellowship from the Social Sciences and Humanities Research Council of Canada.

References


Appendix -- Screen Images of PARSYL in Action

The following is a sample run of PARSYL on the small set of words:
(silks9s alr9ytt flaweds taloita tapro ptenktos)
The first three words are simply the transcribed English words "sixths", "alright", and "flounce". The remaining three words are taken from Steriade (1982): taloita (p. 25), tapro (p. 26), and ptenktos (p. 219). These words were selected to illustrate a variety of the syllabification properties that PARSYL is able to handle. (NOTE: N is the velar nasal.)
The final lists of templates that are acquired for this set of data are as follows:
((templates appendix ((5) (5))))
((templates onset ((7) (2))))
((templates coda ((4) (7))))
SCREEN IMAGE #1 (INITIAL SETTINGS)

buffer

\[
\# s I k s \Theta s \#
\]

- initial parameter values -
- revised parameter values -

- P-non-vocalic-peaks 0
- P-branching-nuc 0
- P-length-of-onset 1
- P-length-of-coda 0
- P-length-of-rhyme 1
- P-homorganic-cluster 0
- P-level-nuclear-seq 0
- P-long-vowels 0
- P-initial-extrasyll 0
- P-final-extrasyll 0
- P-appendix 0

SCREEN IMAGE #2

buffer

\[
\sigma\\
\begin{array}{c}
O\\
R\\
\end{array}
\]

\[
\# s I k s \Theta s \#
\]

c to continue

- initial parameter values -
- revised parameter values -

- P-length-of-rhyme 2
- P-final-extrasyll 1
- P-appendix 1
SCREEN IMAGE #3

```
buffer

σ
\[\begin{array}{c}
R \\
O \\
R \\
N \\
C \\
\end{array}\]
\[
# \text{array} #
\]
c to continue
```

**Initial parameter values**
- P-non-vocalic-peaks 0
- P-branching-nuc 0
- P-length-of-onset 1
- P-length-of-coda 0
- P-length-of-rhyme 1
- P-homorganic-cluster 0
- P-level-nuclear-seq 0
- P-long-vowels 0
- P-initial-extrasyll 0
- P-final-extrasyll 0
- P-appendix 0

**Revised parameter values**
- P-branching-nuc 1
- P-length-of-coda 1
- P-length-of-rhyme 2
- P-final-extrasyll 1
- P-appendix 1

SCREEN IMAGE #4

```
buffer

σ
\[\begin{array}{c}
O \\
R \\
N \\
C \\
\end{array}\]
\[
# \text{flawns} #
\]
c to continue
```

**Initial parameter values**
- P-non-vocalic-peaks 0
- P-branching-nuc 0
- P-length-of-onset 1
- P-length-of-coda 0
- P-length-of-rhyme 1
- P-homorganic-cluster 0
- P-level-nuclear-seq 0
- P-long-vowels 0
- P-initial-extrasyll 0
- P-final-extrasyll 0
- P-appendix 0

**Revised parameter values**
- P-branching-nuc 1
- P-length-of-coda 1
- P-length-of-rhyme 3
- P-final-extrasyll 1
- P-appendix 1
SCREEN IMAGE #5

\[ \begin{array}{c}
\sigma & \sigma & \sigma \\
O & R & O
\end{array} \]

\# t a l o i t a \#

c to continue

**initial parameter values**
- P-non-vocalic-peaks 0
- P-branching-nuc 0
- P-length-of-onset 1
- P-length-of-coda 0
- P-length-of-rhyme 1
- P-homorganic-cluster 0
- P-level-nuclear-seq 0
- P-long-vowels 0
- P-initial-extrasyll 0
- P-final-extrasyll 0
- P-appendix 0

**revised parameter values**
- P-branching-nuc 1
- P-length-of-coda 1
- P-length-of-rhyme 3
- P-final-extrasyll 1
- P-appendix 1

SCREEN IMAGE #6

\[ \begin{array}{c}
\sigma & \sigma \\
O & R
\end{array} \]

\# t a p r o \#

c to continue

**initial parameter values**
- P-non-vocalic-peaks 0
- P-branching-nuc 0
- P-length-of-onset 1
- P-length-of-coda 0
- P-length-of-rhyme 1
- P-homorganic-cluster 0
- P-level-nuclear-seq 0
- P-long-vowels 0
- P-initial-extrasyll 0
- P-final-extrasyll 0
- P-appendix 0

**revised parameter values**
- P-branching-nuc 1
- P-length-of-onset 2
- P-length-of-coda 1
- P-length-of-rhyme 3
- P-final-extrasyll 1
- P-appendix 1
buffer

\[ \sigma \rightarrow^\sigma 0 \quad R \quad 0 \quad R \quad N \quad C \quad N \quad \#
\]
pentektos

c to continue

<table>
<thead>
<tr>
<th>initial parameter values</th>
<th>revised parameter values</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-non-vocalic-peaks 0</td>
<td>P-branching-nuc 1</td>
</tr>
<tr>
<td>P-branching-nuc 0</td>
<td>P-length-of-onset 2</td>
</tr>
<tr>
<td>P-length-of-onset 1</td>
<td>P-length-of-coda 2</td>
</tr>
<tr>
<td>P-length-of-coda 0</td>
<td>P-length-of-rhyme 3</td>
</tr>
<tr>
<td>P-length-of-rhyme 1</td>
<td>P-homorganic-cluster 1</td>
</tr>
<tr>
<td>P-homorganic-cluster 0</td>
<td></td>
</tr>
<tr>
<td>P-level-nuclear-seq 0</td>
<td>P-initial-extrasyll 1</td>
</tr>
<tr>
<td>P-long-vowels 0</td>
<td>P-final-extrasyll 1</td>
</tr>
<tr>
<td>P-initial-extrasyll 0</td>
<td>P-appendix 1</td>
</tr>
<tr>
<td>P-final-extrasyll 0</td>
<td></td>
</tr>
<tr>
<td>P-appendix 0</td>
<td></td>
</tr>
</tbody>
</table>