6. Syllable Structure Typology I: the CV Theory

6.1 The Jakobson Typology

It is well-known that every language admits consonant-initial syllables \( .CV. \), and that some languages allow no others; that every language admits open syllables \( .~V \). and that some admit only those. Jakobson puts it this way:

“There are languages lacking syllables with initial vowels and/or syllables with final consonants, but there are no languages devoid of syllables with initial consonants or of syllables with final vowels.” (Jakobson 1962:526; Clements & Keyser 1983:29.)

As noted in the fundamental work of Clements & Keyser 1983, whence the quotation was cadged, these observations yield exactly four possible inventories. With the notation \( \Sigma_{XYZ} \) to denote the language whose syllables fit the pattern XYZ, the Jakobson typology can be laid out as follows, in terms of whether onsets and codas are obligatory, forbidden, or neither:

(113) **CV Syllable Structure Typology**

<table>
<thead>
<tr>
<th></th>
<th>onsets</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>required</td>
<td>not required</td>
</tr>
<tr>
<td><strong>codas</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>forbidden</td>
<td>( \Sigma^{CV} )</td>
<td>( \Sigma^{(C)V} )</td>
</tr>
<tr>
<td>allowed</td>
<td>( \Sigma^{CV(C)} )</td>
<td>( \Sigma^{(C)VC(C)} )</td>
</tr>
</tbody>
</table>

There are two independent dimensions of choice: whether onsets are required (first column) or not (second column); whether codas are forbidden (row one) or allowed (row two).

The **Basic Syllable Structure Constraints**, which generate this typology, divide notionally into two groups. First, the structural or ‘markedness’ constraints – those that enforce the universally unmarked characteristics of the structures involved:

(114) **ONS**
A syllable must have an onset.

(115) **–Cod**
A syllable must **not** have a coda.
Second, those that constrain the relation between output structure and input:

(116) **PARSE**

Underlying segments must be parsed into syllable structure.

(117) **FILL**

Syllable positions must be filled with underlying segments.

PARSE and FILL are Faithfulness constraints: they declare that perfectly well-formed syllable structures are those in which input segments are in one-to-one correspondence with syllable positions. Given an interpretive phonetic component that omits unparsed material and supplies segmental values for empty nodes, the ultimate force of PARSE is to forbid deletion; of FILL, to forbid insertion.

It is relatively straightforward to show that the Factorial Typology on the Basic Syllable Structure Constraints produces just the Jakobson Typology. Suppose Faithfulness dominates both structural constraints. Then the primacy of respecting the input will be able to force violations of both ONS and \( \neg \text{COD} \). The string /V/ will be parsed as an onsetless syllable, violating ONS; the string /CVC/ will be parsed as a closed syllable, violating \( \neg \text{COD} \): this gives the language \( (C) V (C) \).

When a member of the Faithfulness family is dominated by one or the other or both of the structural constraints, a more aggressive parsing of the input will result. In those rankings where ONS dominates a Faithfulness constraint, every syllable must absolutely have an onset. Input /V/ cannot be given its faithful parse as an onsetless syllable; it can either remain completely unsyllabified, violating PARSE, or it can be parsed as \( \square V \), where ‘\( \square \)’ refers to an empty structural position, violating FILL.

Those rankings in which \( \neg \text{COD} \) dominates a Faithfulness constraint correspond to languages in which codas are forbidden. The imperative to avoid codas must be honored, even at the cost of expanding upon the input (*FILL) or leaving part of it outside of prosodic structure (*PARSE).

In the next section, we will explore these observations in detail. The resulting Factorial construal of the Jakobson Typology looks like this (with ‘\( \mathcal{F} \)’ denoting the Faithfulness set and ‘\( \mathcal{F}_i \)’ a member of it):

---

51 Both FILL and PARSE are representative of families of constraints that govern the proper treatment of child nodes and mother nodes, given the representational assumptions made here. As the basic syllable theory develops, FILL will be articulated into a pair of constraints:

**FILL\text{Nuc}**: Nucleus positions must be filled with underlying segments.

**FILL\text{Mar}**: Margin positions (Ons and Cod) must be filled with underlying segments.

Since unfilled codas are never optimal under syllable theory alone, shown below in §6.2.3 (141), p.104, **FILL\text{Mar}** will often be replaced by **FILL\text{Ons}** for perspicuity.
At this point, it is reasonable to ask whether there is any interesting difference between our claim that constraints like ONS and \(\neg\)COD can be violated under domination and the more familiar claim that constraints can be turned off — simply omitted from consideration. The Factorial Jakobson Typology, as simple as it is, contains a clear case that highlights the distinction. Consider the language \(\sum^{CV(C)}\). Since onsets are not required and codas are not forbidden, the Boolean temptation would be to hold that both ONS and \(\neg\)COD are merely absent. Even in such a language, however, one can find certain circumstances in which the force of the supposedly nonexistent structural constraints is felt. The string CVCV, for example, would always be parsed .CV.CV. and never .CVC.V. Yet both parses consist of licit syllables; both are entirely faithful to the input. The difference is that .CV.CV. satisfies ONS and \(\neg\)COD while .CVC.V. violates both of them. We are forced to conclude that (at least) one of them is still active in the language, even though roundly violated in many circumstances. This is the basic prediction of ranking theory: when all else is equal, a subordinate constraint can emerge decisively. In the end, summary global statements about inventory, like Jakobson’s, emerge through the cumulative effects of the actual parsing of individual items.

6.2 The Faithfulness Interactions

Faithfulness involves more than one type of constraint. Ranking members of the Faithfulness family with respect to each other and with respect to the structural markedness constraints ONS and \(\neg\)COD yields a typology of the ways that languages can enforce (and fail to enforce) those constraints. We will consider only the Faithfulness constraints PARSE and FILL (the latter to be distinguished by sensitivity to Nucleus or Ons); these are the bare minimum required to obtain a contentful, usable theory, and we will accordingly abstract away from distinctions that they do not make, such as between deleting the first or second element of a cluster, or between forms involving metathesis, vocalization of consonants, de-vocalization of vowels, and so on, all of which involve further Faithfulness constraints, whose interactions with each other and with the markedness constraints will be entirely parallel to those discussed here.

6.2.1 Groundwork

To make clear the content of the Basic Syllable Structure Constraints ONS, \(\neg\)COD, PARSE, and FILL, it is useful to lay out the Galilean arena in which they play. The inputs we will be considering are
CV sequences like CVVCC; that is, any and all strings of the language \{C,V\}*. The grammar must be able to contend with any input from this set: we do not assume an additional component of language-particular input-defining conditions; the universal constraints and their ranking must do all the work (see §9.3 for further discussion). The possible structures which may be assigned to an input are all those which parse it into syllables; more precisely, into zero or more syllables. There is no insertion or deletion of segments C, V.

What is a syllable? To avoid irrelevant distractions, we adopt the simple analysis that the syllable node σ must have a daughter Nuc and may have as leftmost and rightmost daughters respectively the nodes Ons and Cod.52 The nodes Ons, Nuc, and Cod, in turn, may each dominate C’s and V’s, or they may be empty. Each Ons, Nuc, or Cod node may dominate at most one terminal element C or V.

These assumptions delimit the set of candidate analyses. Here we list and name some of the more salient of the mentioned constraints. By our simplifying assumptions, they will stand at the top of the hierarchy and will be therefore unviolated in every system under discussion:

**Syllable form:**

(119) NUC

*Syllables must have nuclei.*

(120) *COMPLEX

*No more than one C or V may associate to any syllable position node.*53

**Definition of C and V,** using M(argin) for Ons and Cod and P(eak) for Nuc:

(121) *M/V

*V may not associate to Margin nodes (Ons and Cod).*

(122) *P/C

*C may not associate to Peak (Nuc) nodes.*

The theory we examine is this:

(123) **Basic CV Syllable Theory**

*Syllable structure is governed by the Basic Syllable Structure constraints*  
ONS, –COD, NUC, *COMPLEX, *M/V, *P/C; PARSE, and FILL.
• Of these, ONS, −COD, PARSE, and FILL may be relatively ranked in any domination order in a particular language, while the others are fixed in superordinate position.
• The Basic Syllable Structure Constraints, ranked in a language-particular hierarchy, will assign to each input its optimal structure, which is the output of the phonology.

The output of the phonology is subject to phonetic interpretation, about which we will here make two assumptions, following familiar proposals in the literature:

(124) **Underparsing Phonetically Realized as Deletion**
An input segment unassociated to a syllable position (‘underparsing’) is not phonetically realized.

This amounts to ‘Stray Erasure’ (McCarthy 1979, Steriade 1982, Itô 1986, 1989). Epenthesis is handled in the inverse fashion:

(125) **Overparsing Phonetically Realized as Epenthesis**
A syllable position node unassociated to an input segment (‘overparsing’) is phonetically realized through some process of filling in default featural values.


The terms ‘underparsing’ and ‘overparsing’ are convenient for referring to parses that violate Faithfulness. If an input segment is not parsed in a given structure (not associated to any syllable position nodes), we will often describe this as ‘underparsing’ rather than ‘deletion’ to emphasize the character of our assumptions. For the same reason, if a structure contains an empty syllable structure node (one not associated to an input segment), we will usually speak of ‘overparsing’ the input rather than ‘epenthesis’.

Suppose the phonology assigns to the input /CVVCC/ the following bisyllabic structure, which we write in three equivalent notations:

(126) **Transcription of Syllabic Constituency Relations**, from /CVVCC/

a. \[ \sigma \]
   \[
   \begin{array}{c}
   \text{Ons} \\
   \text{C}
   \end{array} \quad \begin{array}{c}
   \text{Nuc} \\
   \text{V}
   \end{array} \quad \begin{array}{c}
   \text{Ons} \\
   \text{C}
   \end{array} \quad \begin{array}{c}
   \text{Nuc} \\
   \text{V}
   \end{array} \quad \begin{array}{c}
   \text{Cod} \\
   \text{C}
   \end{array}
   \]

b. \([_\sigma [\text{Ons} \ C] [\text{Nuc} \ V]] [_\sigma [\text{Ons}] [\text{Nuc} \ V] [\text{Cod} \ C]] \ C\]

c. \(._{\text{C}}^\text{V} ._{\text{C}}^\text{V} \text{C}._{\text{C}}\)
Phonetic interpretation ignores the final C, and supplies featural structure for a consonant to fill the onset of the second syllable.

The dot notation (126c) is the most concise and readable; we will use it throughout. The interpretation is as follows:

(127) **Notation**
- **a.** \( .X. \) ‘the string X is a syllable’
- **b.** \( +x \) ‘the element x has no parent node; is free (unparsed)’
- **c.** \( \square \) ‘a node Ons, Nuc, or Cod is empty’
- **d.** \( \checkmark \) ‘the element x is a Nuc’

In the CV theory, we will drop the redundant nucleus-marking accent on \( \checkmark \). Observe that this is a ‘notation’ in the most inert and de-ontologized sense of the term: a set of typographical conventions used to refer to well-defined formal objects. The objects of linguistic theory — syllables here — are not to be confused with the literal characters that depict them. Linguistic operations and assessments apply to structure, not to typography.

We will say a syllable ‘has an onset’ if, like both syllables in the example (126), it has an Ons node, whether or not that node is associated to an underlying C; similarly with nuclei and codas.

The technical content of the Basic Syllable Structure Constraints (114–117) above can now be specified. The constraint ONS (114) requires that a syllable node \( \sigma \) have as its leftmost child an Ons node; the presence of the Ons node satisfies ONS whether empty or filled. The constraint \( \neg \text{COD} \) (115) requires that syllable nodes have no Cod child; the presence of a Cod node violates \( \neg \text{COD} \) whether or not that node is filled. Equivalently, any syllable which does not contain an onset in this sense earns its structure a mark of violation \( \ast \text{ONS} \); a syllable which does contain a coda earns the mark \( \ast \neg \text{COD} \).

The PARSE constraint is met by structures in which all underlying segments are associated to syllable positions; each unassociated or free segment earns a mark \( \ast \text{PARSE} \). This is the penalty for deletion. FILL provides the penalty for epenthesis: each unfilled syllable position node earns a mark \( \ast \text{FILL} \), penalizing insertion. Together, PARSE and FILL urge that the assigned syllable structure be faithful to the input string, in the sense of a one-to-one correspondence between syllable positions and segments. This is Faithfulness in the basic theory.

### 6.2.2 Basic CV Syllable Theory

We now pursue the consequences of our assumptions. One important aspect of the Jakobson Typology (113) follows immediately:

(128) **THM. Universally Optimal Syllables**

No language may prohibit the syllable .CV. Thus, no language prohibits onsets or requires codas.
To see this, consider the input /CV/. The obvious analysis .CV. (i.e., \([o \{\text{Ons} C\} \{\text{Nuc} V\}]\)) is *universally optimal* in that it violates *none* of the universal constraints of the Basic CV Syllable Theory (123). No alternative analysis, therefore, can be more harmonic. At worst, another analysis can be equally good, but inspection of the alternatives quickly rules out this possibility.

For example, the analysis .CV \(\square\) violates \(\sim\text{COD}\) and FILL. The analysis .C \(\square\) \(\sim\) V. violates ONS in the second syllable and FILL in the first. And so on, through the infinite set of possible analyses—\([.C/V.], [.C \(\square\).
\(\sim\) V.], [.\(\square\).C \(\square\).
\(\square\).V.], *etc. ad inf.* No matter what the ranking of constraints is, a form that violates even one of them can never be better than a form, like .CV., with no violations at all.

Because every language has /CV/ input, according to our assumption that every language has the same set of possible inputs, it follows that .CV. can never be prohibited under the Basic Theory.

### 6.2.2.1 Onsets

Our major goal is to explicate the interaction of the structural constraints ONS and \(\sim\text{COD}\) with Faithfulness. We begin with onsets, studying the interaction of ONS with PARSE and FILL, ignoring \(\sim\text{COD}\) for the moment. The simplest interesting input is /V/. All analyses will contain violations; there are three possible one-mark analyses:

\[(129) \quad /V/ \rightarrow \]

\[a. \quad .V. \quad \text{i.e.}, \quad [o \{\text{Nuc} V\}]\]

\[b. \quad \langle V \rangle \quad \text{i.e.}, \quad \text{no syllable structure}\]

\[c. \quad .\square V. \quad \text{i.e.}, \quad [o \{\text{Ons} \}] \{\text{Nuc} V\}]\]

Each of these alternatives violates exactly one of the Basic Syllable Structure Constraints (114–117).

\[(130) \textbf{Best Analyses of} /V/\]

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Interpretation</th>
<th>Violation</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>.V.</td>
<td>(\sigma) lacks Ons</td>
<td>(\ast\text{ONS})</td>
<td>satisfies FILL, PARSE</td>
</tr>
<tr>
<td>(\langle V \rangle)</td>
<td>null parse</td>
<td>(\ast\text{PARSE})</td>
<td>satisfies ONS, FILL</td>
</tr>
<tr>
<td>.(\square V.)</td>
<td>ONS is empty</td>
<td>(\ast\text{FILL})</td>
<td>satisfies ONS, PARSE</td>
</tr>
</tbody>
</table>

Every language must evaluate all three analyses. Since the three candidates violate one constraint each, any comparison between them will involve weighing the importance of different violations. The optimal analysis for a given language is determined precisely by whichever of the constraints ONS, PARSE, and FILL is *lowest* in the constraint hierarchy of that language. The lowest constraint incurs the least important violation.

Suppose .V. is the optimal parse of /V/. We have the following tableau:
The relative ranking of FILL and PARSE has no effect on the outcome. The violations of PARSE and FILL are fatal because the alternative candidate .V. satisfies both constraints. Of interest here is the fact that the analysis .V. involves an onsetless syllable. When this analysis is optimal, then the language at hand, by this very fact, does not absolutely require onsets. The other two inferior analyses do succeed in satisfying ONS: +V, achieves this vacuously, creating no syllable at all; □V creates an onsetful syllable by positing an empty Ons node, leading to epenthesis. So if .V. is best, it is because ONS is the lowest of the three constraints, and we conclude that the language does not require onsets. We already know from the previous section, Thm. (128), that onsets can never be forbidden. This means the following condition holds:

(132) If PARSE, FILL >> ONS, then onsets are not required.

(The comma’d grouping indicates that PARSE and FILL each dominate ONS, but that there is no implication about their own relative ranking.)

On the other hand, if ONS is not the lowest ranking constraint, — if either PARSE or FILL is lowest — then the structure assigned to /V/ will be consistent with the language requiring onsets. The following two tableaux lay this out:

(133) Enforcement by Overparsing (Epenthesis)
These lucubrations lead to the converse of (132):

(135) If ONS dominates either PARSE or FILL, then onsets are required.

There is an important difference in status between the two ONS-related implications. To prove that something is *optional*, in the sense of ‘not forbidden’ or ‘not required’ in the inventory, one need merely exhibit one case in which it is observed and one in which it isn’t. To prove that something is *required*, one must show that everything in the universe observes it. Thus, formal proof of (135) requires considering not just one trial input, as we have done, but the whole (infinite) class of strings on \{C,V\}* which we are taking to define the universal set of possible inputs for the Basic Theory. We postpone this exercise until the appendix; in §8 we will develop general techniques which will enable us to extend the above analysis to arbitrary strings, showing that what is true of /V/ and /CV/ is true of all inputs.

The results of this discussion can be summarized as follows:

(136) **Onset Theorem.**

Onsets are not required in a language if ONS is dominated by both PARSE and FILL. Otherwise, onsets are required.

In the latter case, ONS is enforced by underparsing (phonetic deletion) if PARSE is the lowest ranking of the three constraints; and by overparsing (phonetic epenthesis) if FILL is lowest.

If FILL is to be articulated into a family of node-specific constraints, then the version of FILL that is relevant here is FILL\textsuperscript{Ons}. With this in mind, the onset finding may be recorded as follows:

<table>
<thead>
<tr>
<th>Lowest constraint</th>
<th>Onsets are ...</th>
<th>Enforced by ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONS</td>
<td>Not required</td>
<td>N/A</td>
</tr>
<tr>
<td>PARSE</td>
<td>Required</td>
<td>V ‘Deletion’</td>
</tr>
<tr>
<td>FILL\textsuperscript{Ons}</td>
<td>Required</td>
<td>C ‘Epenthesis’</td>
</tr>
</tbody>
</table>
6.2.2.2 Codas

The analysis of onsets has a direct parallel for codas. We consider the input /CVC/ this time; the initial CV provides an onset and nucleus to meet the ONS and NUC constraints, thereby avoiding any extraneous constraint violations. The final C induces the conflict between –COD, which prohibits the Cod node, and Faithfulness, which has the effect of requiring just such a node. As in the corresponding onset situation (130), the parses which violate only one of the basic syllable structure constraints are three in number:

(137) **Best Analyses of /CVC/**

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Interpretation</th>
<th>Violation</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>.CVC.</td>
<td>σ has Cod</td>
<td>*–COD</td>
<td>satisfies FILL, PARSE</td>
</tr>
<tr>
<td>.CV(C)</td>
<td>No parse of 2nd C</td>
<td>*PARSE</td>
<td>satisfies ONS, FILL</td>
</tr>
<tr>
<td>.CV.C□</td>
<td>2nd Nuc is empty</td>
<td>*FILL</td>
<td>satisfies ONS, PARSE</td>
</tr>
</tbody>
</table>

The optimal analysis of /CVC/ in a given language depends on which of the three constraints is lowest in the domination hierarchy. If .CVC. wins, then the language must allow codas; –COD ranks lowest and violation can be compelled. If .CVC. loses, the optimal analysis must involve open (codaless) syllables; in this case –COD is enforced through empty nuclear structure (phonetic V-epenthesis) if FILL is lowest, and through non-parsing (phonetic deletion of C) if PARSE is the lowest, most violable constraint. In either case, the result is that open syllables are required. This is a claim about the optimal parse in the language of every string, and not just about /CVC/, and formal proof is necessary; see the appendix.

The conclusion, parallel to (136), is this:

(138) **Coda Theorem.**

Codas are allowed in a language if –COD is dominated by both PARSE and FILL\textsuperscript{Nuc}. Otherwise, codas are forbidden.

In the latter case, –COD is enforced by underparsing (phonetic deletion) if PARSE is the lowest ranking of the three constraints; and by overparsing (epenthesis) if FILL\textsuperscript{Nuc} is the lowest.

The result can be tabulated like this:
It would also be possible to break this yoke by having two separate PARSE constraints, one that applies to C and another to V. Basic syllable structure constraints that presuppose a C/V distinction, however, would not support the further development of the theory in §8, where the segment classes are derived from constraint interactions.

Motivation for distinguishing the constraints FILL\textsuperscript{Ons} and FILL\textsuperscript{Nuc} is now available. Consider the languages Σ\textsuperscript{CV} in which only CV syllables are allowed. Here ONS and \textminus COD each dominate a member of Faithfulness group. Enforcement of the dominant constraints will be required. Suppose there is only one FILL constraint, holding over all kinds of nodes. If FILL is the lowest-ranked of the three constraints, we have the following situation:

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
\textbf{Input} & \textbf{Optimal Analysis} & \textbf{Phonetic} \\
\hline
/V/ & .□V. & .CV. \\
\hline
/CVC/ & .CV.C□. & .CV.CV. \\
\hline
\end{tabular}
\caption{Triumph of Epenthesis}
\end{table}

The single uniform FILL constraint yokes together the methods of enforcing the onset requirement (\textquote{C-epenthesis}) and the coda prohibition (\textquote{V-epenthesis}). There is no reason to believe that languages Σ\textsuperscript{CV} are obligated to behave in this way; nothing that we know of in the linguistic literature suggests that the appearance of epenthetic onsets requires the appearance of epenthetic nuclei in other circumstances. This infelicitous yoking is avoided by the natural assumption that FILL takes individual node-classes as an argument, yielding FILL\textsuperscript{Nuc} and FILL\textsuperscript{Ons} as the actual constraints. In this way, the priority assigned to filling Ons nodes may be different from that for filling Nuc nodes.\textsuperscript{54}

It is important to note that onset and coda distributions are completely independent in this theory. Any ranking of the onset-governing constraints \{ONS, FILL\textsuperscript{Ons}, PARSE\} may coexist with any ranking of coda-governing constraints \{\textminus COD, FILL\textsuperscript{Nuc}, PARSE\}, because they have only one constraint, PARSE, in common. The universal factorial typology allows all nine combinations of the three onset patterns given in (136) and the three coda patterns in (138). The full typology of interactions is portrayed in the table below. We use subscripted \textit{del} and \textit{ep} to indicate the phonetic consequences of enforcement; when both are involved, the onset-relevant mode comes first.

\textsuperscript{54} It would also be possible to break this yoke by having two separate PARSE constraints, one that applies to C and another to V. Basic syllable structure constraints that presuppose a C/V distinction, however, would not support the further development of the theory in §8, where the segment classes are derived from constraint interactions.
(140) **Extended CV Syllable Structure Typology**

<table>
<thead>
<tr>
<th>Onsets</th>
<th>required</th>
<th>not required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONS, FILL\textsuperscript{Ons} \gg PARSE</td>
<td>ONS, PARSE \gg FILL\textsuperscript{Ons}</td>
<td>PARSE, FILL\textsuperscript{Ons} \gg ONS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Codas</th>
<th>forbidden</th>
<th>(-\text{COD, FILL}\textsuperscript{Nuc} \gg \text{PARSE})</th>
<th>(\sum CV_{del,del})</th>
<th>(\sum CV_{ep,del})</th>
<th>(\sum (CV)_{del})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(-\text{COD, PARSE} \gg \text{FILL}\textsuperscript{Nuc})</td>
<td>(\sum CV_{del,ep})</td>
<td>(\sum CV_{ep,ep})</td>
<td>(\sum (CV)_{ep})</td>
<td></td>
</tr>
<tr>
<td>allowed</td>
<td>PARSE, FILL\textsuperscript{Nuc} \gg \text{COD}</td>
<td>(\sum CV(C)_{del})</td>
<td>(\sum CV(C)_{ep})</td>
<td>(\sum (CV(C)))</td>
<td></td>
</tr>
</tbody>
</table>

If we decline to distinguish between the Faithfulness constraint rankings, this simplifies to the Jakobson Typology of (118).

### 6.2.3 The Theory of Epenthesis Sites

The chief goal of syllabification-driven theories of epenthesis is to provide a principled account of the location of epenthetic elements (Selkirk 1981, Broselow 1982, Lapointe and Feinstein 1982, Itô 1986, 1989). Theories based on manipulation of the segmental string are capable of little more than summary stipulation on this point (e.g. Levin 1985:331; see Itô 1986:159, 1989 for discussion). The theory developed here entails tight restrictions on the distribution of empty nodes in optimal syllabic parses, and therefore meets this goal. We confine attention to the premises of the Basic CV syllable structure theory, which serves as the foundation for investigation of the theory of epenthesis, which ultimately involves segmental and prosodic factors as well.

There are a few fundamental observations to make, from which a full positive characterization of syllabically-motivated epenthesis emerges straightaway.

(141) **Prop. 1.** \([\text{\textsubscript{Cod}}]\)  

Coda nodes are never empty in any optimal parse.

Structures with unfilled Cod can never be optimal; there is always something better. To see this, take a candidate with an unfilled Cod and simply remove that one node. This gives another candidate which has one less violation of \(-\text{COD}\) and one less violation of FILL. Since removing the node has no other effects on the evaluation, the second candidate must be superior to the first. (To show that something is non-optimal, we need merely find something better: we don’t have to display the best.)
We know from the earlier discussion that Ons and Nuc must be optimally unfilled in certain parses under certain grammars. So the remaining task is to determine the conditions under which these nodes must be posited and left empty.

(142) **Prop. 2.** *.(*□)C

A whole syllable is never empty in any optimal parse.

The same style of argument applies. Consider a parse that has an entirely empty syllable. Remove that syllable. The alternative candidate thereby generated is superior to the original because it has (at least) one less \( \text{FILL}^\text{Nuc} \) violation and no new marks. The empty syllable parse can always be bested and is therefore never optimal.

Of course, in the larger scheme of things, whole syllables can be epenthesized, the canonical examples being Lardil and Axininca Campa (Hale 1973, Klokeid 1976, Itô 1986, Wilkinson 1988, Kirchner 1992a; Payne 1981, 1982, Spring 1990, Black 1991, McCarthy & Prince 1993). In all such cases, it is the impact of additional constraints that forces whole-syllable epenthesis. In particular, when the prosody/morphology interface constraints like \( \text{LX} ≃ \text{Pr} \) are taken into account, prosodic minimality requirements can force syllabic epenthesis, as we will see for Lardil in §7 below.

(143) **Prop. 3.** *.(*□)C.

No syllable can have \( \text{Cod} \) as its only filled position.

Any analysis containing such a syllable is bested by the alternative in which the content of this one syllable (namely ‘C’) is parsed instead as \(.C\). This alternative incurs only the single mark \( \text{FILL}^\text{Nuc} \), but the closed-syllable parse \(.(*□)C\). shares this mark and violates \( \text{Cod} \) as well. (In addition, the closed-syllable parse must also violate either \( \text{Ons} \) or \( \text{FILL}^\text{Ons} \).)

Such epentheses are not unknown: think of Spanish //slavo// → eslavo and Arabic //ḥmarar// → ḥmarar. We must argue, as indeed must all syllable theorists, that other constraints are involved (for Arabic, see McCarthy & Prince 1990b).

(144) **Prop. 4.** *[ ] *

Adjacent empty nodes cannot occur in an optimal parse.

Propositions 1, 2 and 3 entail that \[ ] cannot occur inside a syllable. This leaves only the intersyllabic environment \(.C\). This bisyllabic string incurs two marks, \( \text{FILL}^\text{Nuc} \) and \( \text{FILL}^\text{Ons} \).

Consider the alternative parse in which the substring \(/CV/\) is analyzed as tautosyllabic \(.CV\). This eliminates both marks and incurs no others. It follows that two adjacent epentheses are impossible.

We now pull these results together into an omnibus characterization of where empty nodes can be found in optimal parses.
(145) **Fill Violation THM. Location of possible Epenthesis Sites.**
Under the basic syllable structure constraints, epenthesis is limited to the following environments:

a) Onset, when Nucleus is filled:
   \[ \Box V. \]
   \[ \Box V C. \]

b) Nucleus, when Onset is filled:
   \[ .C \Box. \]
   \[ .C \Box C. \]

Furthermore, two adjacent epentheses are impossible, even across syllable boundaries.

We note that this result will carry through in the more complex theory developed below in §8, in which the primitive C/V distinction is replaced by a graded sonority-dependent scale.