1 Introduction

In this squib, we explore the generative consequences of candidate omission for constraint-based grammars. In principle, the candidate set for an input form $i$ is the range of $Gen(i)$ for a given generating function $Gen$ (see Prince and Smolensky 1993:17). In practice, however, analyses often cover just a few candidates that are deemed relevant. The consequences of this practice are the focus of this squib.

Whether the omission is by oversight or by principle, a range of defects can readily arise because there are situations where, even though an analysis might include the observed/intended optimum along with a range of viable competitors, it is still possible to draw erroneous inferences about languages and typologies owing to the omission of a single candidate.

Aside from the—unfortunately familiar—problem of discovering an omitted competitor that breaks one’s current analysis because it is more harmonic than the intended/observed form, there are three other risks associated with candidate omission that we discuss here:

1. Omission of certain candidates—even suboptimal ones—can lead to flawed generalizations about how observed patterns extend to novel forms.
2. This can occur even for intentional “principled” omission if it is derived post hoc, rather than via the generative mechanism of the grammar.
3. These errors can multiply explosively when generating typologies.

For the sake of simplicity, the examples and discussion (up until section 5) are couched in terms of the ranked constraint interactions of Optimality Theory (OT; Prince and Smolensky 1993, McCarthy and Prince 1995, Kager 1999). However, our conclusions are equally applicable to grammars with weighted constraints such as those in Goldsmith 1989, 1993, 1994, Goldsmith and Larson 1990, Legendre, Miyata, and Smolensky 1990, Bosch 1991, Wiltshire 1992, Legendre, Sorace, and Smolensky 2006, Pater 2009, and Potts et al. 2009.
2 Candidate Omission

In this section, we illustrate a range of ways in which the omission of candidates can corrupt an analysis. The basic problem can be illustrated with an extremely simple pair of schematic examples, shown in (1) and (2).

Table: (1) $T_1$:

<table>
<thead>
<tr>
<th></th>
<th>$i_1$</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

Rankings:

- $a$ wins if $C1 \gg C2$
- $b$ wins if $C2 \gg C1$

Table: (2) $T_2$:

<table>
<thead>
<tr>
<th></th>
<th>$i_2$</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>d</td>
<td></td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

Rankings:

- $c$ wins if $C1 \gg C2$
- $d$ wins if $C2 \gg C1$

In this schematic example, each constraint ranking produces a distinct outcome (i.e., candidates $a$ and $c$ are optimal if $C1 \gg C2$ but $b$ and $d$ are optimal if $C2 \gg C1$). Consider, however, the consequences of omitting one of the candidates, as in (2).

Table: (2) $T_2'$:

<table>
<thead>
<tr>
<th></th>
<th>$i_2$</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td></td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

Rankings:

- $c$ wins under any ranking

With the omission of candidate $d$, $T_1$ and $T_2'$ still allow two distinct outcomes, but now the ranking $C2 \gg C1$ selects candidates $b$ and $c$ (instead of $b$ and $d$).

We will refer to each of these outcomes as a language. Specifically, we say that language $L = \{(T_1, a), (T_2', c)\}$ is generated by ranking $R = C1 \gg C2$ over tableaux $\{T_1, T_2'\}$ if $R$ makes $a$ and $c$ optimal in their respective tableaux. We assume each $T_n$ contains the candidate set for one input form. Ideally, this set should be the complete output of Gen, but this definition also covers languages implied by candidate sets that fall short of the ideal. We will say that a given candidate has been omitted from an analysis if (a) its absence has not been motivated by Gen or an undominated constraint and (b) there is some ranking/weighting under which it would have been optimal.

There are three general cases where omitting a candidate like $d$ in example (1) can be problematic. The first—and somewhat trivial—case arises when a form like $d$ is among those that the analyst intends the grammar to generate. This error can be readily avoided if a tableau is created for each intended optimum.

The second case arises when the analyst intends the grammar to generate a form like $c$. In this case, omitting $d$ leads to the conclusion
that \( c \) is generated by a stratified hierarchy with \( C_1 \) and \( C_2 \) in the same stratum despite the fact that \( C_1 \) must dominate \( C_2 \) to rule out \( d \). In this case, we say that the analysis is incomplete.

The third—and most pernicious—case arises when the analyst intends the grammar to generate a pair of forms like candidates \( b \) and \( c \). In this case, omitting \( d \) leads to the conclusion that ranking \( C_2 \gg C_1 \) generates the intended outcome. Here, we say that the analysis is inconsistent because there is, in fact, no ranking that makes \( b \) the optimal output for \( i_1 \) and also makes \( c \) the optimal output for \( i_2 \).

Cases like this often come to light when a helpful reviewer or audience member inquires about a relevant candidate that seems to have been left out of an analysis. Such observations are often met with the assertion that there is actually some third, unmentioned yet highly ranked constraint \( C_3 \) that rules out candidates like \( d \).

Continuing our running example, a schematic case in which this response would be valid is given in (3).

\[
\begin{array}{c|c|c|c}
T_1: & i_1 & C_3 & C_1 & C_2 \\
\hline
a. & & * & & \\
b. & & & & *
\end{array}
\]

\[
\begin{array}{c|c|c|c}
T_2: & i_2 & C_3 & C_1 & C_2 \\
\hline
c. & & * & & \\
d. & & & & *
\end{array}
\]

Though (3) shows that candidate omission can sometimes be justified by appeal to constraints that have themselves been omitted (i.e., \( b \) and \( c \) are co-optimal under \( R = C_3 \gg C_2 \gg C_1 \)), suppressed premises of this sort are dangerously fraught with the potential for unforeseen consequences (a point we return to in section 4).

### 3 Typological Consequences of Omission

Given a set of tableaux \( T \) and constraints \( Con \), each ranking of the constraints in \( Con \) generates a language by selecting one candidate as optimal in each tableau (we take outputs with the same violation profile to constitute a single candidate). We refer to this set of languages as the typology generated by \( Con \) over \( T \).

The set of languages in the typology that \( Con \) generates over \( T \) is a subset of the range of logically possible ways to choose one candidate per tableau. Given this fact, one might intuitively expect that omission would generally impoverish the typology by dint of removing logically possible candidate combinations.
In some instances, this is indeed what happens (e.g., omitting \(b\) from (6) in section 4). More surprisingly, the opposite can also occur; there are some situations where the omission of a candidate introduces novel, possibly spurious distinctions that can actually inflate the typology. For a schematic example that shows how omission can cause typological inflation, consider the pair of tableaux in (4).

<table>
<thead>
<tr>
<th>(4) (T_1):</th>
<th>(i_1)</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(a) wins if (C3 \gg {C1,C2,C4})</td>
</tr>
<tr>
<td>(b)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(b) wins if (C2 \gg {C1,C3,C4})</td>
</tr>
<tr>
<td>(c)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(c) wins if (C1) or (C4 \gg {C2,C3})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(4) (T_2):</th>
<th>(i_2)</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(d) wins if (C1 \gg {C2,C3,C4})</td>
</tr>
<tr>
<td>(e)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(e) wins if (C4 \gg {C1,C2,C3})</td>
</tr>
<tr>
<td>(f)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(f) wins if (C2) or (C3 \gg {C1,C4})</td>
</tr>
</tbody>
</table>

What is the typology of these two tableaux? Of the 9 pairings of candidates from \(T_1\) and \(T_2\), only 4 can be generated. For example, the pair \((c,d)\) is generated if \(C1\) is top-ranked, denoted \(((c,d) \leftrightarrow C1 \gg \{C2,C3,C4\})\), and the other three languages are \(((b,f) \leftrightarrow C2 \gg \{C1,C3,C4\})\), \(((a,f) \leftrightarrow C3 \gg \{C1,C2,C4\})\), and \(((c,e) \leftrightarrow C4 \gg \{C1,C2,C3\})\).

The mechanism by which candidate omission can inflate a typology becomes clear upon considering the rankings under which \(d\) and \(e\) are optimal in \(T_2\), and the way these change in \(T_2'\), where \(f\) has been omitted. This is illustrated in (5).

<table>
<thead>
<tr>
<th>(5) (T_2'):</th>
<th>(i_2)</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(d) wins if (C1 \gg C4)</td>
</tr>
<tr>
<td>(e)</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>(e) wins if (C4 \gg C1)</td>
</tr>
</tbody>
</table>

With candidate \(f\) gone, the relative merits of \(d\) and \(e\) become relevant in rankings with \(C3\) or \(C2\) in the top spot (i.e., those that selected \(f\) in \(T_2\)). This allows 6 pairs of optima: \((c,d)\) and \((c,e)\) are generated as before, but \((a,f)\) is replaced by \(((a,d) \leftrightarrow C3 \gg \{C1,C2\} \& C1 \gg C4)\), \(((a,e) \leftrightarrow C3 \gg \{C4,C2\} \& C4 \gg C1)\) and \((b,f)\) is replaced by \(((b,d) \leftrightarrow C2 \gg \{C1,C3\} \& C1 \gg C4)\), \(((b,e) \leftrightarrow C2 \gg \{C3,C4\} \& C4 \gg C1)\).
This shows that even though omission results in fewer logically possible ways to choose optima across tableaux (e.g., omitting $f$ cuts the possible combinations down from 9 to 6), it can nonetheless inflate the typology because languages $(a, f)$ and $(b, f)$ are replaced by $\{(a, d), (a, e), (b, d), (b, e)\}$ to yield a 6-language-typology that is larger than, but not a superset of, the original 4-language typology.

4 Runners-Up and Covert Constraints

The accidental omission of candidates can sometimes be relatively harmless. For instance, an incomplete analysis that fails to rule out an omitted competitor can readily be fixed by including that competitor or by including another tableau that provides ranking information sufficient to rule it out.

The real danger comes when omission makes an analysis inconsistent in the sense that no ranking generates the intended forms if the omitted candidate is present. Incompleteness can be remedied by additional data, but inconsistency can only be resolved in favor of the desired optimum by changing the constraints.

Henceforth, we assume that $Con = \{\text{Onset}, *\text{Coda}, \text{DepV}, \text{DepC}, \text{Max}\}$. The first two constraints respectively penalize syllables that lack onsets and have codas, the next two penalize epenthesis of vowels and consonants, and the last penalizes deletion. Following further in the spirit of the CV-syllable theory of Prince and Smolensky 1993: chap. 6, we assume that Gen freely allows deletion and insertion of C, V, and “.” (the syllable boundary symbol) but excludes clusters and excludes syllables without exactly one V (e.g., no output contains CC, VV, .C., or VCV).

In (6), we give two tableaux, $\{T_1, T_2\}$, in which each candidate is annotated with the size of the factorial typology that would have resulted if it had been omitted. Complete descriptions of these typologies can be found in Bane and Riggle 2012.

(6) $T_1$: $\text{/VC/} \quad \text{DepC} \quad \text{DepV} \quad \text{Max} \quad \text{*Coda} \quad \text{Onset}$

<table>
<thead>
<tr>
<th></th>
<th>DepC</th>
<th>DepV</th>
<th>Max</th>
<th>*Coda</th>
<th>Onset</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. CVC</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. V</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. θ</td>
<td></td>
<td>**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. CV</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>e. CV.CV</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f. VC</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>g. V.CV</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h. CV</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

removal $\mapsto$ 12 lgs.
removal $\mapsto$ 11 lgs.
removal $\mapsto$ 12 lgs.
removal $\mapsto$ 12 lgs.
removal $\mapsto$ 13 lgs.
removal $\mapsto$ 12 lgs.
removal $\mapsto$ 13 lgs.
removal $\mapsto$ 11 lgs.
These tableaux have respectively 8 and 4 candidates, but of the $8 \times 4 = 32$ candidate pairs, only 12 are in the typology of $\text{Con over } \{T_1, T_2\}$. Each candidate occurs in at least one of the 12 pairs, so each omission changes the typology in a different way.

For example, in the full typology, $b$ only occurs in $(b, k)$, but $k$ also occurs in $(c, k)$ and $(f, k)$. When $b$ is omitted, the typology shrinks because the rankings for $(b, k)$ are divided between $(c, k)$ and $(f, k)$. Candidate $e$ has a similar status in the full typology; it only occurs in $(e, j)$, while $j$ has two other partners in $(d, j)$ and $(g, j)$. If $e$ is omitted, some of its rankings go to $(d, j)$ and $(g, j)$ but others yield entirely new pairings, $(a, j)$ and $(h, j)$, thereby enlarging the typology. As a final example, omitting $c$ reshuffles the typology over $\{T_1, T_2\}$, with $d$ taking the place of $c$ for some rankings and thereby yielding a typology of the same size.

If, however, candidates like $c$ are omitted in all tableaux, the consequences are more striking. For instance, suppose one wanted to categorically ban the null candidate $\emptyset$ by disallowing candidates in which the entire underlying form is deleted. (See Orgun and Sprouse 1999 and Wolf and McCarthy 2010 for contrasting views on constraints against null candidates.) Consider the results of omitting $\emptyset$ when a third tableau such as $T_3$ in (7) is added to the analysis of the tableaux in (6).

### (7) $T_3$:

<table>
<thead>
<tr>
<th></th>
<th>/CC/</th>
<th>DepC</th>
<th>DepV</th>
<th>Max</th>
<th>*Coda</th>
<th>Onset</th>
</tr>
</thead>
<tbody>
<tr>
<td>m. CV.CV</td>
<td></td>
<td>**</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n. CVC</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>o. $\emptyset$</td>
<td></td>
<td></td>
<td>**</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The removal of candidate $o$ would simplify the ranking conditions as in (8).

### (8) $T_3'$:

<table>
<thead>
<tr>
<th></th>
<th>/CC/</th>
<th>DepC</th>
<th>DepV</th>
<th>Max</th>
<th>*Coda</th>
<th>Onset</th>
</tr>
</thead>
<tbody>
<tr>
<td>m. CV.CV</td>
<td></td>
<td>**</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n. CVC</td>
<td></td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
However, this would be a serious misstep because candidate $o$ cannot be discarded without taking into account the status of $p$ relative to the other candidates in (9).

<table>
<thead>
<tr>
<th>(9) $T_3''$:</th>
<th>/CC/</th>
<th>DepC</th>
<th>DepV</th>
<th>Max</th>
<th>*Coda</th>
<th>Onset</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$. CV.CV</td>
<td></td>
<td>**</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n$. CVC</td>
<td></td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$o$. $\emptyset$</td>
<td></td>
<td></td>
<td></td>
<td>**</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p$. CV</td>
<td></td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If candidate $o$ were present, then $p$ would have been collectively harmonically bounded by candidates $m$ and $o$ (i.e., bounded neither by $m$ nor by $o$ individually, but by their combined ranking entailments; see Samek-Lodovici and Prince 2002 for discussion of various kinds of harmonic bounding). However, with $o$ gone, some rankings that would have selected $o$ should select $p$; but if omitted $o$ is not replaced by formerly bounded $p$, then what was intended as a ban on $\emptyset$ will act as if it were some kind of *tableau-specific* constraint penalizing the candidate CV.

Extending the post hoc ban on $\emptyset$ to more tableaux has even more convoluted consequences. Consider tableaux for all inputs in $\{C, V\}^n$ (i.e., all strings of Cs and/or Vs of length zero through $n$), and in particular the case of $\{C, V\}^3$. Though Gen defines infinite candidate sets for each of the 14 input forms of $\{C, V\}^3$, only 55 of the candidates are contenders (i.e., not harmonically bounded). The typology that Con generates over the 14 tableaux in $\{C, V\}^3$ consists of 12 languages—this is the same 12-way distinction that was realized by pairs of optimal candidates drawn from $T_1$ and $T_2$ in (6).

Of the 14 tableaux, 9 contain the null candidate $\emptyset$. If these candidates are discarded, the typology over the 46 candidates that remain will have 26 languages. If, however, an undominated constraint against $\emptyset$ is included in the generation of candidates, then the tableaux for $\{C, V\}^3$ will contain 58 contenders (because some $\emptyset$ are replaced by multiple runners-up) and the typology will have 20 languages.

The contrast between the post hoc omission of $\emptyset$ and a properly implemented restriction on candidate sets provides another case where a smaller set of logically possible candidate combinations yields a larger typology.

### 5 Ranking, Weighting, and Inflation

As a final example, we examine a hypothetical case of candidate omission with weighted rather than ranked constraints. For a model of weighted constraints, we will consider Harmonic Grammar (HG; Legendre, Miyata, and Smolensky 1990). Optimization in HG selects...
candidates so as to minimize the weighted sum (or ‘cost’) of their violations according to a given weighting of the constraints.

For any upper bound \( n \) on violations, the typology generated by weighting is a superset of that generated by ranking over the same constraints and tableaux. This is so because the behavior of ranking can be simulated using weights whose differences are exponential in \( n \) (Prince and Smolensky 1993:219).

The typology generated by \( Con \) over \( \{C, V\}^3 \) under weighting has 23 languages, and these are a superset of the 12 languages generated by OT. For 12 of the 14 input forms in \( \{C, V\}^3 \), the set of contenders is identical regardless of the ranking/weighting distinction. (Prince (2002) calls this an ‘anything goes’ relation in the candidate set.) However, for two of the inputs, /VVC/ and /CCC/, there are candidates that are optimal under weighting but that cannot win under any ranking. We will say that these candidates are HG contenders but not OT contenders.

(10)\[
\begin{array}{|c|c|c|c|}
\hline
& \text{DepV} & \text{Max} & \text{*Coda} \\
\hline
a. CVC.CV & \star & \star & \star \\
b. CV.CV.CV & \star \star \star & \star \star \star & \\
c. \emptyset & \star \star \star & \star \star \star & \\
d. CVC & \star & \star & \star \\
\hline
\end{array}
\]
\[ \rightarrow \text{Collectively bounded in OT} \]

No ranking can make \( d \) optimal in (10) because it alone violates all 3 constraints. There are, however, constraint weightings such as \((\text{DepV}:4, \text{Max}:3, \text{*Coda}:1)\), which will assign candidates \( \{a, b, c, d\} \) the respective costs of \([9, 12, 9, 8]\).

The other input in \( \{C, V\}^3 \) with novel HG contenders is /VVC/, which allows 2 novel HG contenders. Thus, there are only 3 HG contenders for \( \{C, V\}^3 \) above and beyond the 55 candidates that are contenders in both OT and HG.

Though these extra contenders mark an obvious difference between ranking and weighting, they are not pivotal to the distinction between the two frameworks. That is, even if /VVC/ and /CCC/ are ignored, all of the OT/HG distinctions are preserved by differences in the combinations of optima that OT and HG generate over the other 12 tableaux (i.e., those with the same contenders in both models).

Consider the following thought experiment: What would happen if, instead of forms like /VVC/ and /CCC/ being ignored entirely, just the novel HG contenders were ignored and typologies were built using only forms allowed by both models?

We illustrate the results in table 1 for sets of inputs ranging from \( \{C, V\}^2 \) to \( \{C, V\}^5 \). Columns OTHG and HGHG give the respective sizes of the OT and HG typologies, and the column labeled HGOT gives the sizes of HG-style typologies constructed using only candidates that
are contenders under both HG and OT. Full detail for the \{C, V\}³ case is presented in Bane and Riggle 2012. As with the post hoc removal of ∅, omitting the HG contenders for forms like /VVC/ and /CCC/ introduces a patchwork of spurious distinctions that reflect differences among runners-up. The most striking thing about table 1 is the way these distinctions can compound to inflate the typology. Though the omitted candidates admit a simple description—those harmonically bounded under ranking—their exclusion introduces structure very different from that introduced by a normal constraint.

This last example might seem slightly absurd. No one has suggested that HG typologies should be constructed from OT contenders, nor does it seem reasonable to expect the results of such an enterprise to be sound. Our intention, however, is merely to show that the results of omission can be unpredictable and that patterns of omission guided by a fairly simple principle can nonetheless fail to correspond in any obvious way to an undominated constraint or restriction on Gen.

### 6 Algorithmic Candidate Generation

The omission of a single relevant candidate can introduce spurious typological distinctions and hide critical inconsistencies in an analysis. This suggests that it is important to include all relevant candidates. The potential complexity of this task argues strongly for algorithmic generation of tableaux.

Algorithms for generating complete sets of contenders within the framework of OT are provided by Tesar (1995) for syllable structures, by Prince (2010) for metrical structures, and by Riggle (2004) for the general case of structures evaluated by grammars in which the constraints are representable with finite automata.

Riggle (2009) presents a contender-generating algorithm whose complexity is linear in the length of input forms, modulo the number of resulting contenders (which can be quite large). This algorithm is formulated in the semiring parsing framework (Goodman 1998, Mohri 2002) and thus extends transparently to cases with context-free constraints. Furthermore, we elsewhere (Bane and Riggle 2010, to appear) extend contender generation to grammars with weighted constraints and provide algorithms for building typologies with ranked and weighted constraints in the software package PyPHON, available online at code.google.com/p/clml/.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>OT</th>
<th>HG</th>
<th>HGOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>{C, V}²</td>
<td>11</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>{C, V}³</td>
<td>12</td>
<td>23</td>
<td>38</td>
</tr>
<tr>
<td>{C, V}⁴</td>
<td>12</td>
<td>23</td>
<td>58</td>
</tr>
<tr>
<td>{C, V}⁵</td>
<td>12</td>
<td>23</td>
<td>109</td>
</tr>
</tbody>
</table>
7 Conclusions

In this squib, we have shown that the omission of candidates can give rise to faulty grammatical inferences and to spurious typological distinctions that are not justified by constraints in the model under consideration and, in the worst case, that cannot be justified by the addition of any imaginable constraint.

These problems can thus invalidate not only analyses of individual languages, but also larger arguments about the merits of alternative (sets of) constraints and modes of constraint interaction such as ranking and weighting. These problems are especially disconcerting in cases where the arguments rest on claims about specific details of the typological predictions of the models under consideration.

Given the inherent difficulty of hand-generating complete sets of contenders, the importance of methods for systematically generating candidates is paramount. To highlight one positive example, the typological analyses of stress systems in Gordon 2002, Tesar 2004, and Heinz 2007 suggest (parts of) a “gold standard” for analytical clarity in terms of explicit formal detail. We say this because, in these cases, it has been possible for independent researchers to exactly replicate each other’s typologies of hundreds and even thousands of predicted languages owing to clear specifications of Gen, clear definitions of the constraints, and clear descriptions of the candidate sets under consideration in the various analyses.

In the case of stress systems, this clarity has undoubtedly been facilitated by the restricted scope of the analyses, the finitude of the candidate sets (i.e., there is no epenthesis), and the simplicity of the candidates’ representations. To facilitate this kind of specificity for more elaborate and open-ended sets of phenomena, there is a clear need for development of more tools like those described in section 6 to automate the generation and checking of predictions made by optimization-based grammars.

References


Bosch, Anna. 1991. Phonotactics at the level of the phonological word. Doctoral dissertation, University of Chicago, Chicago, IL.


