Chapter 7

Word Structure and Syllable Structure with SA-OT

This chapter enlarges further the scope of techniques that the Simulated Annealing Optimality Theory Algorithm offers us. Two phenomena are dealt with, both related to syllabification and syllable structure. First, the results of Judit Gervain’s psycholinguistic experiments on the cliticisation of the definite article in Hungarian are modelled, and then Prince and Smolensky (1993)’s well-known basic syllable structure theory is implemented using SA-OT.

The topologies of the candidate sets in both models are similar to that presented in section 6.5. Recall the infinite search space used there (Fig. 6.3 on page 169), and the fact that we distinguished between radial and tangential moves. In fact, the search space had a centre, an origin in which candidates had no epenthetic segments. Centrifugal moves involved applying epenthesis recursively, centripetal moves undid epenthesis, whereas tangential moves corresponded to other operations on the candidate string, which did not change the number of epenthetic elements, that is, the “distance” from the origin. As a typical candidate had four neighbours, and each of them was assigned the same a priori probability, therefore performing a tangential step had an a priori probability of 50%, whereas centripetal and centrifugal steps had 25% each (with the remark at the end of footnote 17 on page 181).

In the case of both phenomena to be introduced, we shall start with a similar topology, but then make them more complex. Some of the conclusions to be drawn can also be applied to the description of phenomena discussed earlier.

7.1 Th’ article in Hungarian

7.1.1 The behaviour of the definite article

Let us now review a model accounting for the behaviour of the definite article in Hungarian, which is similar to the model we have just discussed (section 6.5),
but which displays further features of SA-OT. The definite article has two allomorphs, and the choice between them depends on whether the next word begins with a consonant or with a vowel. For instance:

\[ \text{az alma} \quad \text{‘the apple’}, \]
\[ \text{a szalma} \quad \text{‘the straw’}. \] (7.1)

The pause between the \( \text{az} \) allomorph of the article and the subsequent word is prone to omission, resulting in the cliticising of the article. The hypothesis has been that cliticising is more frequent with an acceleration of the speech rate (Kiefer, 1994). In order to support the hypothesis, Judit Gervain has performed a series of controlled psycholinguistic experiment measuring the frequency of the phenomenon (so far unpublished, reported by Biró and Gervain, 2006). Her experiments confirmed this hypothesis by measuring the presence and the overall length of pauses in critical minimal pairs (e.g. \( \text{az ár} \ ‘\text{the price}’ \) as opposed to \( \text{a zár} \ ‘\text{the lock}’ \) excised from test sentences pronounced by four female native speakers in three conditions.

She reports that for the \( \text{a} \) allomorph, cliticisation is the default, irrespective of speed. However, \( \text{az} \) cliticises (i.e., the length of pause is less than 3 msec) only in 1 case out of 12 (8.3%) at a slow speaking rate. This proportion grows to 4 cases out of 12 (33.3%) at a medium rate, and to 8 cases out of 12 (66.7%) at a fast rate. Furthermore, detailed results show that the length of the pauses correlates inversely with speed, the average length of the pause being significantly shorter at a medium speech rate than in slow speech. A first explanation based on the intuition of the native speakers could be that if the \( \text{a} \) allomorph cliticises, the syllable boundaries still align with the morpheme boundaries; if, however, the \( \text{az} \) allomorph cliticises, the segment \( \text{[z]} \) is resyllabified into the onset of the subsequent word, resulting in a violation of the relevant alignment constraint, which should be avoided at slower speech rate.

### 7.1.2 Constructing a model

The model to be presented resembles the one employed to account for the magic square-type phenomena, such as Dutch voice assimilation. This model is also based on the infinity of the search space, even if its structure is slightly different. Moreover, we shall tune the frequencies again by having the random walker rove away from the origin due to large \( K_{\text{max}} \) values.

The candidates to be considered for an input such as \( \text{az ár} \) will have the form \([az^n\#^mE]\); between the [a] segment of the article and the arbitrary initial vowel E of the subsequent word, segment [z] of length \( n \) is followed by a pause of length \( m \) \((n, m \geq 0)\). Exponents \( n \) and \( m \) can also be thought of as time units, for instance given in msec. The initial candidate, from which the simulations are launched, will always be \([az\#E]\), that is \( n = m = 1 \), and basic steps alter the values of \( n \) and \( m \). Thus, candidates of the form \([a\#^mz^nE]\) \((n, m > 0)\) are never reached, even though these would come into play if the input were something like \( \text{a zebra} \ ‘\text{the zebra}’ \) or \( \text{a zár} \ ‘\text{the lock}’ \). The pause between the article and the subsequent word is considered to be omitted if the exponent \( m \)

\(^{1}\text{The present section builds upon Biró and Gervain (2006), but presents a more elaborate model for the same phenomenon.}\)
7.1. Th’ article in Hungarian

of the pause symbol # is less than three, corresponding to Gervain’s definition of cliticising (measuring a pause shorter than 3 msec).\footnote{As correctly remarked by Paul Boersma, the exponents of [z] and of [#] cannot have both the interpretation of measuring the respective durations in msec, because the outputs are going to be candidates with n = 1, while the duration of a segment [z] in reality is about 100 milliseconds. Hence, this asymmetry has to be accounted for, either in the definition of the exponents, or otherwise.}

A basic step consists of changing both n and m by 1— one may not change only n or m. This stipulation may sound \textit{ad hoc}, but the success of the model will depend on it, and has the advantage of creating a simple topology.\footnote{Readers criticised this stipulation, similarly to the decision in Chapter 5 of not allowing the insertion and deletion of a bisyllabic foot as a basic step. As proposed there, too, the general principle might be that the set of basic steps should be minimal in the sense that leaving out one of them would create a topology in which not all candidates can be reached from any other candidate. Now, as both n and m should change, the candidates with an odd \(n + m\) value cannot be reached from candidates with an even \(n + m\) value. And yet, I have the impression (which will not be shared by most readers) that leaving out the “odd half” of the candidate set does not really influence the general structure, while including further basic steps breaks the logic of the structure to be explained soon. Changing n only (or m only) corresponds to the idea of a “diagonal” step that could be decomposed into a radial and a tangential one. Nonetheless, I am also open to alternative proposals.} A general candidate \([az^n \#^m E]\) has four neighbours only: \([az^{n-1} \#^m E]\), \([az^n \#^{m+1} E]\), \([az^{n-1} \#^{m+1} E]\) and \([az^{n+1} \#^{m-1} E]\). Obviously, if n or m is 0, the candidate has fewer neighbours. In other words, one may shorten the candidate by 2, one may lengthen the candidate by 2, and one may change the proportions of \([z]\) and the pause without changing the overall length.

The resulting topology, presented in Fig. 7.1, has a similar structure to the topology dealt with in section 6.5 (Fig. 6.3 on page 169). There are radial steps (shortening the candidate is a centripetal step, while lengthening the candidate is taking a centrifugal step), as well as tangential steps, perpendicular to the radial ones, which change the difference of the number of \([z]\)’s and \(#\)’s. If changing only \(n\) without changing \(m\), and changing only \(m\) without changing \(n\) were also allowed, then diagonal steps could be also possible. In some sense, radial moves change the “quantity”, and tangential steps change the “quality” of the candidate, and combining the two is not possible within one basic step. In the present case, permitting \(n\) or \(m\) not to change in a basic step, that is, having for instance \([az^{n+1} \#^m E]\) as a neighbour of \([az^n \#^m E]\), might be viewed as a diagonal move in Fig. 7.1, which would be a combination of both qualitative and quantitative changes in the candidate. Such diagonal moves would, however, prevent the candidates that should be returned by the algorithm to become local optima.

The markedness constraints to be used are very simple:

\[
\begin{align*}
C_1(w) &= \text{KEEPSHORT}([az^n \#^m E]) = n + m \\
C_0(w) &= \text{KEEPSEGMENTSHORT}([az^n \#^m E]) = n
\end{align*}
\]

Both reflect some principles of economy: \textsc{KeepShort} punishes long strings in general, whereas \textsc{KeepSegmentShort} disprefers long segments. Importantly, “pronouncing” the pause requires negligible energy, so no separate constraint \textsc{KeepPauseShort}— whose value on candidate \([az^n \#^m E]\) would be \(m\)— is needed; alternatively, such a constraint should be ranked (universally) lower for the same rationale. Observe, furthermore, that these constraints follow also
the logic of the search space: \textit{KeepShort} is the constraint that briddles recursive insertions of \([z\#]\), while \textit{KeepSegmentShort} influences the tangential movements. In Fig. 7.1, the first constraint forces the system to stay left, and the second constraint to stay as close to the bottom as possible.

Each of the four possible basic steps involves a well-defined change in the violation level of each constraint, so there is no need to re-evaluate the candidates at every iteration of the algorithm. The difference of the violation profiles follows directly from the basic step chosen by the algorithm. This strong connection between the structure of the candidates, the topology and the (markedness) constraints improves the speed of the algorithm, and is an illustration of what I refer to as the SA-OT implementation being built organically upon the underlying traditional OT model.

Our goal is to have the system return candidates \([az\#^{2k+1}E]\): the consonant of the article is kept always short, while the pause might have different lengths. The special case \(k = 0\) corresponds to cliticisation, because the pause is so short that it is unperceivable (our system is unable to return candidate \([azE]\)), whereas a larger \(n = 2k + 1\) corresponds to a (shorter or longer) audible pause. In order to make these candidates local optima, we still need to disqualify the bottom left-most candidates \([a\#^{m}E]\), which are more harmonic than their neighbours for the constraints introduced so far. That is a simple task, once we observe that the candidates to be disqualified miss the \([z]\) segment of the input. The following constraint will do the work:

\[
C_2(w) = \text{Faithfulness}(az^n\#^{m}E) = \begin{cases} 
1 & \text{if } n = 0 \\
0 & \text{if } n \geq 1
\end{cases} \quad (7.3)
\]

It is only by ranking this latter constraint above both markedness constraints that candidates \([az\#^{2k+1}E]\) become local optima in our topology:

\[
\text{Faithfulness} \gg \text{KeepShort} \gg \text{KeepSegmentShort} \quad (7.4)
\]

The expected behaviour of this system is similar to that of the one analysed in section 6.5. After being launched from candidate \([az\#E]\), the random walker may freely rove in the initial phase of the simulation. The larger the \(K_{max}\),
7.1. The article in Hungarian

Figure 7.2: Frequencies of [az#”E] employing different $K_{\text{max}}$ values, as a function of $m$, with a linear (left box) and a logarithmic (right box) frequency axis. The different graphs correspond to $K_{\text{max}} = 1$ (solid line), $K_{\text{max}} = 3$ (dotted line), $K_{\text{max}} = 5$ (dot-dashed line) and $K_{\text{max}} = 7$ (dashed line) respectively. $T_{\text{step}} = 0.1$.

the farther it gets. Once temperature reaches the domain of constraint Keep-Short, insertions are prohibited, and the random walker slowly gravitates back towards the origin. In this second phase, approximately half of the time steps are employed to perform “tangential” moves, that is, vertical ones in Fig. 7.1.

As soon as the temperature cools down to the domain of constraint Keep-SegmentShort, candidates [az#2$k+1$E] become traps wherein the random walker may get stuck. The “competition” analysed in the previous section translates here to a competition between the centripetal moves leading back to candidate [az#E] and the tangential moves trapping the system to another candidate [az#2$k+1$E]. The difference between the two models is that the [bd] channel was only a trap to the tangential moves and “channelled the water” towards [bd] in a centripetal direction, whereas now candidates [az#2$k+1$E] are local optima, for centripetal steps are also prohibited by the high-ranked constraint Faithfulness. The “water channel” is replaced here by a series of “water reservoirs”. Even without working out the formal analysis that would be similar to the one presented in the previous section, we expect the chance of returning candidate [az#2$k+1$E] ($k > 0$) to increase as parameter $K_{\text{max}}$ grows larger.

Running the simulation under the “usual” conditions results in Fig. 7.2. Constraints were assigned the indices 2, 1 and 0. The parameters were $T_{\text{max}} = 3$, $T_{\text{min}} = 0$, $T_{\text{step}} = 0.1$ and $K_{\text{step}} = 1$. Furthermore, instead of employing $K_{\text{min}}$, the algorithm was run each time until the random walker had not moved for 30 iterations: in the case of four neighbours, the likelihood of having a more harmonic neighbour but not finding it in 30 trials is $0.75^{30} < 0.0002$. This technique corresponds to measuring the “specific heat” in standard simulated annealing: there, the stopping condition is that the specific heat—the decrease in the target function divided by the decrease in temperature—drops below a certain value, that is, much is expected to be gained if going further. Finally, the simulation was run 25000 times using each of four different $K_{\text{max}}$ values,
Chapter 7. Word Structure and Syllable Structure with SA-OT

and Fig. 7.2 presents the frequency of being returned candidate [az#\(^m\)E] as a function of \(m\) (only odd \(m\)'s appear on the graphs).

The graphs confirm our prediction. While the probability of \(m \leq 25\) is around 99.5\% if \(K_{\text{max}} = 5\) and around 95\% if \(K_{\text{max}} = 10\), increasing the length of the initial phase results in only 84\% for \(K_{\text{max}} = 20\), and 68\% for \(K_{\text{max}} = 40\). Candidates with a probability above 1\% are the candidates [az#\(^m\)E] with \(m < 21\) for \(K_{\text{max}} = 5\), \(m < 29\) for \(K_{\text{max}} = 10\), \(m < 39\) for \(K_{\text{max}} = 20\), and \(m < 53\) for \(K_{\text{max}} = 40\). In brief, the larger \(K_{\text{max}}\), the broader the distribution of the outputs. As a larger \(K_{\text{max}}\) requires a longer running time, our model correctly predicts longer pauses at slower speech rates.

7.1.3 Refining the model by changing the topology

This result does not satisfy us, however. These distributions are, namely, centred around the origin, and the candidate with \(m = 1\) (omission of the pause) has always the highest chance; whereas experimental results observed a longer gap in most of the cases for slower speech rates, with the pause being almost never omitted. Therefore, we need a model that produces a distribution similar to those in Fig. 7.3: in fast speech, the length of the pauses has a distribution around zero, but at a slower rate, the distribution is centred around some larger \(m\). The slower the rate, the further is the distribution shifted to the right. How can we produce such a distribution?

Observe that so far the random walker has had an equal chance to move to the centripetal and to the centrifugal direction in the initial phase. Hence, both in the section on voice assimilation and in the present model, the distribution of the random walker’s position at the end of the initial phase has been centred around the origin. It has been true that larger \(K_{\text{max}}\) values increase the chance of reaching a more remote region of the search space, but the region with the highest probability has always remained the centre.

It is by introducing a bias into the random walk that one can force the random walker to leave the central region. As pointed out in footnote 19 on page 183, the expected position of the random walker in an asymmetric, one-dimensional Brownian motion is proportional to the difference of the probability of moving left and moving right. Consequently, if the \(a\ priori\) probability of lengthening the candidate is increased, and the \(a\ priori\) probability of shortening the candidate is decreased, a drift is introduced into the system, and the random walker’s position by the end of the initial phase will be some distribution centred around a more remote point in the search space. Then, even if lengthening the candidate becomes impossible after the temperature has reached constraint \text{KEEPSHORT}, and the system starts gravitating backwards, the final distribution is not necessarily centred around the origin. If the most probable position of the random walker at the end of the initial phase is far enough, then the chance is very low for the random walker to get back to the origin, and most probably it will be stuck in some farther local optimum.

Figure 7.3 has been obtained by introducing a simple change into the \(a\ priori\) probabilities. Earlier, the exponent of [\(z\)] was increased or decreased by one with a probability of 0.5 each, and the same applied, independently, to the exponent of [\(#\)] (whenever possible). Now, we first toss a coin, and with a chance of 0.5, we lengthen the candidate (both exponents are increased by 1), and with a chance of 0.5, we apply the earlier algorithm. Thereby, the
7.1. Th’ article in Hungarian

Figure 7.3: Frequencies of [az#mE] employing different $K_{\text{max}}$ values, as a function of $m$, with the altered $\textit{a priori}$ probabilities. The frequency axis is linear on the left box, and logarithmic on the right one. The different graphs correspond to $K_{\text{max}} = 5$ (solid line), $K_{\text{max}} = 10$ (dotted line), $K_{\text{max}} = 20$ (dotted line) and $K_{\text{max}} = 40$ (dashed line) respectively. $T_{\text{step}} = 0.1$.

earlier $\textit{a priori}$ probabilities of 0.25 each (in the general case) have been altered: $P_{\text{choice}}([az^{n+1}#m+1E]| [az^n#mE]) = 5/8$, whereas the $\textit{a priori}$ probabilities of the three other neighbours have been reduced to 1/8 (in the case of $n, m > 0$). This technique enables us to have the model fit the empirical observations better. But it also demonstrates the important role that a further component of the SA-OT algorithm, namely, the $\textit{a priori}$ probabilities, play in determining the output frequencies.

7.1.4 Refining the model by demoting constraints

Judit Gervain’s psycholinguistic experiment has also confirmed the different behaviour of the $a$ allomorph from that of the $az$ allomorph. So far, our model accounts for the speech-rate dependent cliticisation of the $az$ allomorph, but are we also able to include the fact that the $a$ allomorph almost always cliticises (the case of $a$ zebra ‘the zebra’)? The solution demonstrates a further dimension of SA-OT models.

Figure 7.4 shows the search space analogous to the previous one (Fig. 7.1), but for the $a+$zebra case. As our constraints have been insensitive to the order of the pause and the segment [z] within a candidate, the two models should display exactly the same behaviour. (The definitions in (7.2) and in (7.3) can be easily generalised to a candidate of the form [$a#mz^nE$].) Which is not what we aim at, because a certain parameter setting is supposed to be characteristic for a particular speaker, a particular speech situation or a particular speech rate, so our goal is to predict significantly different frequencies for the two types ($a+zár ‘the lock’, as opposed to $az+ár ‘the price’) using the same parameter setting.

It has been already mentioned that in the $a+zár$ case, a support or permission to cliticisation might be that the concatenation creates a sequence in which the well-formed syllable structure preserves the morphological structure. In the $az+ár$ case, however, either the syllables are suboptimal (by including a
Chapter 7. Word Structure and Syllable Structure with SA-OT

coda and missing an onset), or the resyllabified structure does not align with the morphological boundaries. Can we introduce this difference into our model?

A simple proposal could be to reinterpret constraint KeepSegmentShort. Without having to change its definition in (7.2) for our purposes, let us regard it as the well-known constraint *ComplexCoda in the az'ar case, and constraint *ComplexOnset if a+zar is the input:

\[
C_0(w) = \ast \text{ComplexCoda}([az^n\#^m{z}^kE]) = n
\]
\[
C_{-1}(w) = \ast \text{ComplexOnset}([az^n\#^m{z}^kE]) = k
\]  
(7.5)

Both constraints disfavour a higher complexity on the respective edge of a syllable. These two constraints should be ranked with respect to each other, thus:

\[
\text{Faithfulness} \gg \text{KeepShort} \gg \ast \text{ComplexCoda} \gg \ast \text{ComplexOnset}
\]  
(7.6)

Arguments for the relative ranking of constraints *ComplexCoda and *ComplexOnset should be brought traditionally from the syllabification of words with an internal consonant cluster such as asztronómia 'astronomy', in which one can either avoid having a complex onset or a complex coda. The native speaker is, however, uncertain about the syllabification of such cases, and is unavoidably influenced by orthographic rules learned in school. However, a different type of argument will be brought for the ranking of these two constraints based on our modelling of cases where these two constraints do not even conflict seemingly.

In short, if constraint Faithfulness is associated with domain 2 and KeepShort with domain 1, then *ComplexCoda can be assigned index (domain)

\footnote{Prince and Smolensky (2004, p. 108) introduce the constraint *Complex, which prescribes that “[n]o more than one C or V may associate to any syllable position node”, and they add in a footnote that “[t]he constraint *Complex is intended as no more that a cover term for the interacting factors that determine the structure of syllable margins”. Nevertheless, several variants of constraint *Complex are widespread in contemporary phonological literature, each of which apply only to a certain domain, especially to codas. For examples, see Wheeler (2005) or Bye (2005).}
7.1. The article in Hungarian

Figure 7.5: The effect of demoting constraint *KeepSegmentShort*: The frequency of an output with a pause of length \( m \) among \( 10^5 \) outputs is displayed as a function of \( m \), for \( T_{\text{step}} = 0.1 \) (upper left box), for \( T_{\text{step}} = 0.05 \) (upper right box), and for \( T_{\text{step}} = 0.01 \) (lower boxes, with linear and logarithmic frequency axes respectively; notice the change of the horizontal scale, as well). In each box, constraint *KeepSegmentShort* is associated either with domain 0 (solid line, rightmost), or with domain −2 (dotted line), −4 (dotdashed line), −6 (shortdashed line), or −8 (longdashed line, leftmost, only in the lower boxes). \( K_{\text{max}} = 3 \).

0, while the index of *ComplexOnset* should be −1 or even lower (if further constraints intervene).

In the \( az+\ddagger r \) case, the vacuously fulfilled, low-ranked *ComplexOnset* does not interfere with our previously presented simulation results. Similarly, constraint *ComplexCoda* is vacuously fulfilled in the \( a+\ddagger z\ddagger r \) case, but it does influence the frequencies. Namely, it lengthens the phase during which the random walker gravitates back to the origin, and candidates \([a\#2^k+i\ddagger z\ddagger E]\) do not act yet as a trap. This idea corresponds to increasing \( n_1 \) in (6.19) on page 184: the random walker has a larger probability of reaching the origin \([a\#z\ddagger E]\), or at least of getting stuck in a local optimum \([a\#^mz\ddagger E]\) with a small \( m \).

The plots in Fig. 7.5 presents experimental results on demoting constraint *KeepSegmentShort* to lower domains, while constraint *Faithfulness* is kept associated with index (domain) 2, and *KeepShort* with 1. In each box, the solid line represents what we have had so far (cf. Figure 7.3, but \( K_{\text{max}} = 3 \),
that is, constraint \texttt{KeepSegmentShort} is associated with domain 0. This situation corresponds to the case when \texttt{KeepSegmentShort} is in fact constraint \texttt{*ComplexCoda} for input \([az+E]\). As discussed in the previous subsection, the distribution of the outputs is shifted to the right due to the bias introduced into the topology. If, however, constraint \texttt{KeepSegmentShort} is demoted and associated with a lower domain (that is, the lower ranked \texttt{*ComplexOnset} is acting on input \([a+zE]\)), while no effective constraint is associated with domain 0, then this distribution is shifted back to the left. In the extreme case, if \texttt{KeepSegmentShort} is ranked very low, then the normal-like distribution centred around some \(m\) turns into a quickly decreasing distribution.

For \(T_{step} = 0.1\) and constraint \texttt{KeepSegmentShort} demoted to domain \(-6\), output \(m = 1\) is returned in 20% of the cases, and \(m = 3\) in another 11%. The same values are 50% and 5%, if \(T_{step} = 0.01\) and constraint \texttt{KeepSegmentShort} is demoted to domain \(-8\). Besides, this figure also demonstrates the influence of \(T_{step}\) on this model: lower \(T_{step}\) involves more steps in the initial phase, therefore the distribution is shifted further to the right, and is broader.

Here, we make crucial use of the way temperature decreases through the domains of the constraints. The length of the pause in \(a+z\bar{a}\) or in \(a+z\bar{e}\)bra can be tuned (or an observable pause can be practically eliminated) by introducing further domains between \texttt{*ComplexCoda} and \texttt{*ComplexOnset}—either by inserting other constraints in between, or by simply associating constraint \texttt{*ComplexOnset} with a lower index. Hence, the temperature has to cross empty domains (domains not associated with any constraint or with any relevant constraint) before it reaches \texttt{*ComplexOnset}.

Here, we touch upon the role of parameter \(K_{step}\), so far absolutely neglected, since by varying this parameter it is also possible to tune the model. Without changing the indices of constraints \texttt{*ComplexCoda} and \texttt{*ComplexOnset}, but by decreasing parameter \(K_{step}\), you can also introduce empty domains between these constraints. If, for instance, \(K_{step} = 0.5\), then a phase is added to the simulation during which the first component of the temperature is \(-0.5\), and which phase does not differ from the case when the first component of the temperature is \(-1\), but \texttt{*ComplexOnset} is associated with index \(-2\).

Finally, this solution also has a consequence for traditional OT. In a traditional approach, based on the data presented, one would not be able to rank constraints \texttt{*ComplexCoda} and \texttt{*ComplexOnset} relative to each other, because for some inputs \texttt{*ComplexCoda} is vacuously satisfied, whereas in other cases \texttt{*ComplexOnset} does not influence the computation. In SA-OT, however, the hierarchy must be such as in (7.6), because this is the hierarchy that guarantees that allomorph \(a\) cliticises much more often than allomorph \(az\). The reversed ranking would reverse the frequency of cliticisation.

### 7.1.5 Conclusion

In the present section, a model has been presented to account for the resyllabification or cliticisation of the Hungarian article. Similarly to the extended magic square model advanced in section 6.5, this model also exploits the infinity of the search space, but has an infinite number of local optima. The major difference is therefore the fact that the local optima, the candidates \([az\#\"E]\), are not neighbours of each other, so they do not form a “channel” any more.

Explaining the different behaviour of the two allomorphs was possible by
employing two different constraints, ranked into different domains. Both constraints measured the number (or length) of the segment $[z]$, but temperature reached that domain later in the $[a+zE]$ case. In other words, an empty domain (the domain of a vacuously satisfied constraint) is introduced for $[a+zE]$ inputs. Consequently, not only does this model demonstrate the proper role of domains to be traversed by temperature above the highest ranked constraint, it also shows that empty domains between the constraints can also influence the model. Moreover, introducing empty domains can be replaced by employing $K_{step} \leq 0.5$, too, thus even this parameter of the algorithm might influence the output.

Finally, if lengthening the candidate is assigned a higher \textit{a priori} probability than shortening it, we introduce a bias in the random walk, that is, a drift, which helps us better reproducing experimental observations. This alternation of the topology already leads us to discussing the next model, the goal of which is to see what happens if the \textit{a priori} probabilities vary in a given interval.

### 7.2 Syllabification (CVT) theory

In the following section, we implement the basic CV Theory for syllabification using simulated annealing. There are a few reasons for doing that. Besides being the most known example of an OT grammar since Prince and Smolensky (1993), it is also a model that has been implemented using different technologies. As already mentioned in section 1.2, Tesar and Smolensky (2000, Chapter 8) employ dynamic programming (chart parsing) for this task, whereas Gerdemann and van Noord (2000) demonstrate the matching approach to Finite-State OT exactly on syllabification. Furthermore, it offers us the possibility to discuss further the role of the topology in SA-OT.

To begin with, the search space generated is infinite due to the possibility of inserting epenthetic vowels and consonants in a recursive way. Not only that, but the number of neighbours will grow with the length of the candidate, since a longer candidate can be altered at more places. The topology will, therefore, turn more complex than any other topology discussed in my dissertation, even though its basic structure consisting of radial and tangential moves is the same as the structure of the topologies discussed so far. Additionally, the constraints also operate on each substring of a candidate independently, which results in a high number of local optima and, hence, in a drop in the precision of SA-OT.

By enlarging the set of basic transformations in an \textit{ad hoc} way, the performance of the model can be improved; nevertheless, further work is required to create a really elegant model. As it is the case for SA-OT in general, the precision of SA-OT lags behind the precision of other techniques (dynamic programming, finite state OT); but I conjecture that SA-OT will be able to account for typical performance phenomena, such as the drop (underparsing) of syllable parts in fast speech (\textit{partij} ‘political party’ becoming \textit{p’tij} in Dutch, \textit{azt hiszem} ‘I think’ shortening to \textit{asszem} in Hungarian, as well as numerous examples from other languages). Indeed, a major claim of my thesis is that human performance lags also behind the precision of dynamic programming and finite state OT.
7.2.1 Basic CV Theory

First, we repeat Prince and Smolensky (1993)’s syllable theory, which follows Jakobson’s typology.

The input is a string of segments, such as abab. What GEN does is to parse its segments into a syllable structure. A candidate is a series of syllables, each syllable containing a nucleus, preceded by an optional onset, and followed by an optional coda. Additionally, the string may contain underparsed (deleted) segments, which we shall return to soon. For the sake of simplicity, we assume that the phonemes of the language can be divided into two distinct sets, vowels may appear only as nuclei, and consonants only as onsets or codas.\(^5\)

Hence, an underlying vowel can be either underparsed or parsed as a nucleus, and an underlying consonant can be either underparsed or parsed as an onset or as a coda. Additionally, overparsing may insert onsets, nuclei and codas that do not contain underlying segments, but epenthetic (default) material, such as a schwa or a [t]. Ignoring the underparsed segments, which are not pronounced, the result must be a well-formed word, that is, a sequence of well-formed syllables (exactly one nucleus, preceded optionally by an onset and followed optionally by a coda). The candidate set corresponding to an input is the set of all possible candidates whose underlying (i.e., not epenthetic) material forms that input, by also keeping the linear order of the segments.\(^6\)

For the sake of convenience, we do not allow complex (branching) onsets and codas, that is, a syllable may contain at most one consonant as onset, and at most one consonant as coda. This constraint applies only to certain languages, but now our only aim is to keep the model as simple as possible.

Following the notation of Gerdemann and van Noord (2000), let N[a] denote an underlying element (the phoneme /a/ in this case) parsed as a nucleus; moreover, O[b] and D[b] refer to the consonant /b/ parsed as onset and coda respectively. By X[a] or X[b] we represent the underparsing (deletion) of some underlying material, whereas N[,] O[,] and D[,] shows the insertion of the default epenthetical vowel or consonant in the position of nucleus, onset or coda. Not surprisingly, X[,] is avoided: deleting a previously inserted element is realised by simply removing it from the string.

For example, if the underlying representation is ba, possible candidates include O[b]N[,]O[,]N[a], O[,]N[,]O[b]N[a], and X[b]N[a]N[ ]. However, O[b]O[,]N[a] or O[b]D[,]N[a] are not valid candidates, since the first one includes a branching onset, while the first syllable of the second one lacks a nucleus. The otherwise well-formed candidates N[a]D[b], O[b]N[a]D[b] or O[d]N[e] are not element of the candidate set either, for they correspond to different inputs.

Following Prince and Smolensky (1993), and most work in their footsteps, we use the following five constraints:

- Onset (ons): the number of nuclei in the candidate that are not preceded immediately by an onset.\(^7\)

\(^5\)Syllabic consonants are therefore seen as vowels. What we ignore is the possibility of changing the syllabicity of a segment, such as turning a vowel into a glide or a plain consonant into a syllabic consonant. In that respect, we follow in the footsteps of the earlier work referred to, as our first goal is to illustrate SA-OT, and not to account exhaustively for specific linguistic phenomena.

\(^6\)Consequently, this model does not allow metathesis and reduplication.

\(^7\)Immediate precedence is understood in the surface form. That is, underparsed segments
7.2. Syllabification (CVT) theory

- **NoCoda (noc)**: the number of codas (Ds) in the candidate.
- **Parse (prs)**: the number of underlying segments that is underparsed in the candidate (the number of Xs).
- **FillOnset (fio)**: the number of onsets in the candidate that are not parses of an underlying segment (the number of O[s]).
- **FillNucleus (fin)**: the number of nuclei in the candidate that are not parses of an underlying segment (the number of N[s]).

Motivated by Jakobson’s typology, and found in Prince and Smolensky (2004, p. 106), constraint **Onset** requires each syllable to have an onset, and constraint **NoCoda** prefers each syllable not to have a coda. Unlike these markedness constraints, the last three are faithfulness constraints: they punish any difference between the input and the output.

### 7.2.2 Syllabification with simulated annealing I.

From the building blocks of the SA-OT Algorithm, we have introduced the constraints, so let us now concentrate on the definition of the topology. Through what basic step shall we construct a neighbour from the actual candidate, the present position of the random walker? Unlike so far, instead of defining formally the set of neighbours $\text{Neighb}(w)$ and the a priori probability distribution on it, now we rather advance a procedure that constructs some neighbour $w'$ which the present candidate $w$ will be compared to.

The proposed algorithm first checks if there is any intervocalic consonant that can be reparsed (turn an onset into a coda or a coda into an onset). If there is at least one, the algorithm generates a random number $r$ between 0 and 1 with an equal distribution, and if $r < P_{\text{reparse}}$, the basic step to be performed is reparsing. In this case, one of the possible loci for reparsing is chosen with equal probability, where subsequently reparsing takes place.

If no such locus exists, or if $r \geq P_{\text{reparse}}$, then the word is lengthened or shortened. The next decision to be made is whether to lengthen or to shorten the candidate. If shortening is not possible (the random walker is located in the centre of the search space), then lengthening takes place. Otherwise, each has a chance ($P_{\text{centrifugal}}$ and $P_{\text{centripetal}}$) of 50%, because if shortening would be preferred over lengthening, then the random walker would stay around the origin (the candidate with no epenthetical position and all underlying segments underparsed). Increasing the probability $P_{\text{centrifugal}}$ of lengthening might be an option for future research, analogous to the model producing Fig. 7.3 (on page 201, in subsection 7.1.3), but does not seem to be very promising: here—unlike there—falling into distant local optima is not attested in speech. In fact, the 50%-chance-each model parallels the earlier model described in section 7.1, as well as the model in section 6.5: moving away from the centre has the same a priori probability as moving backwards ($P_{\text{centrifugal}} = P_{\text{centripetal}}$). This stipulation allows us to concentrate on further parameters of the topology, whereas the role of parameter $P_{\text{centrifugal}}$ has already been analysed in subsection 7.1.3.

Intervening between an onset and a nucleus do not cause the candidate to violate this constraint.
Chapter 7. Word Structure and Syllable Structure with SA-OT

Subsequently, once it has been decided that the candidate will be shortened, then all possibilities of shortening the candidate are listed, and one of the possibilities is chosen with equal chance. A possibility involves performing one of the following operations at a given locus of the candidate string:

1. Underparse an underlying consonant (parsed as an onset or coda).
2. Delete an epenthetic onset (O[ ] ) or coda (D[ ]).
3. Underparse an underlying vowel (parsed as a nucleus), supposing that what remains is a well-formed candidate.\footnote{A nucleus can be deleted if it is the first parsed element of the candidate and the next parsed element is not a coda; if the previous or the next parsed element is a nucleus; or if the previous parsed element is a coda and the next one is not a coda.}
4. Delete an epenthetic nucleus (N[ ] ), supposing that what remains is a well-formed candidate.

For instance, candidate O[b]N[a]N[ ]D[c]N[a] can be shortened at five different points, so each possibility has a chance of 20%.

Similarly, if the candidate’s fate is to be lengthened, then one is picked from the possibilities, where a possibility is rewriting a single locus using one of the following operations:

1. Insert an epenthetic nucleus (N[ ] ), which is possible everywhere (between any two parsed, underparsed or overparsed elements, at the beginning and at the end of the string).
2. Insert an epenthetic onset (O[ ] ), supposing that the previous parsed (or overparsed; if there is one) element is not an onset, and the next one is a nucleus.
3. Insert an epenthetic coda (D[ ] ), supposing that the previous parsed (or overparsed) element is a nucleus, and the next one (if there is one) is not a coda.
4. Turn an underparsed vowel into a nucleus.
5. Turn an underparsed consonant into an onset, supposing that the previous parsed (or overparsed; if there is one) element is not an onset, and the next one is a nucleus.
6. Turn an underparsed consonant into a coda, supposing that the previous parsed (or overparsed) element is a nucleus, and the next one (if there is one) is not a coda.

So far, we have a topology similar to those in sections 6.5 and 7.1, but more complex. The similarity is that the search space has a centre, the candidate whose length is minimal in pronunciation. Furthermore, possible moves are either radial or tangential with respect to this centre. The present model is more complex, however, for a candidate string can be lengthened at any point. Not only that, but we have also introduced a parameter, \( P_{\text{reparse}} \), that determines the probability of the tangential moves. In the earlier models, the probability...
of considering a tangential move was 50%, because each neighbour had equal \textit{a priori} probabilities, and most often two neighbours out of four represented a tangential move.

Additionally, we render our model even more complex by introducing further neighbours. The reason for that is that if you run a simulation with the present model, you will most often be stuck in some local optimum, similarly to the search spaces in section 7.1. But unlike that case, one is unhappy now if this happens, because local optima are non-attested in speech. These local optima include cases such as epenthetical syllables: if the substring \(O[N[c]]\) is followed by a consonant or by the end of the word, then deleting the nucleus is impossible, while deleting the onset makes the candidate worse if \textsc{Onset} \(\gg\) \textsc{FillOnset}. Another type of local optima is formed by candidates with a substring such as \(N[v][a]\) if \textsc{Onset} \(\gg\) \textsc{Parse}: deleting the epenthetical nucleus might bring to an ill-formed string, whereas reparsing the vowel \([a]\) first increases the violations of constraint \textsc{Onset}.

Therefore, an additional parameter \(P_{\text{postproc}}\) is introduced in the definition of the topology. After having performed exactly one basic operation (resyllabification, lengthening or shortening), some post-processing may also occur. Each substring \(O[N[c]]\) is considered, and if the next parsed element is not a coda, then this substring is deleted with probability \(P_{\text{postproc}}\). Similarly, each substring \(N[v][X[c]]\), \(X[v]N[l]\), \(O[l][X[c]]\) and \(X[c]O[l]\) (where \(v\) stands for any vowel and \(c\) for any consonant) is collapsed into \(N[v]\) or into \(O[c]\) with the same probability. These operations help to avoid certain traps, but the analysis of the experiments performed demonstrate that further operations should also be allowed in the future.\(^9\) Note finally that due to the irreversibility of these post-processing operations, the neighbourhood relation is not symmetric anymore.

In sum, two parameters determine the \textit{a priori} probabilities of the topology, \(P_{\text{reparse}}\) and \(P_{\text{postproc}}\). How do they influence the precision of the algorithm? The experiments display huge differences in function of the input string and of the hierarchy employed, but also of \(T_{\text{step}}\). The role of the latter here is similar to its role in section 6.5: lower values allow for the random walker to move farther away from the origin in the initial stage of the simulation.

Here, I report on some short experiments performed with the hierarchy

\[
\text{noc} \gg \text{prs} \gg \text{ons} \gg \text{fin} \gg \text{fio}
\]

and with initial form \(0[1]N[a]D[b]O[d]N[a]D[k]\) (from input /ladbak/). The optimal candidate is \(0[1]N[a]0[b]N[l]O[d]N[a]O[k]N[l]\) with two epenthetical nuclei, as epenthesis is preferred to deletion and to having codas. The constraints were associated with ranks 0 to 4, and the parameters of the algorithm were: \(T_{\text{max}} = 3, T_{\text{min}} = 0, T_{\text{step}} = 0.01, K_{\text{max}} = 5, K_{\text{min}} = -2\) and \(K_{\text{step}} = 1\). The simulations were run 750 times for a certain \((P_{\text{reparse}}, P_{\text{postproc}})\) pair, so that the mean and the standard deviation \((\sigma(N - 1))\) of the frequencies

\(^9\)These include deleting sequences of \(N[l][O[c]],\) as well as reparsing whole syllables: turning \(X[c]X[v]\) into \(O[c]N[v]\), and \(X[c]\) into \(O[c]N[l]\) in one go. The gradual inclusion of such \textit{ad hoc} operations lessens the simplicity of the model, and future research will hopefully propose a more elegant solution. Another direction has also been advanced, namely, to prevent \textsc{GEN} from generating candidates with similar redundant or verbose substructures.
Table 7.1: Varying the parameters of the *a priori* probabilities: The frequencies of the optimal form and of the most frequently returned non-global local optimum are reported in function of the parameters $P_{\text{reparse}}$ and $P_{\text{postproc}}$. See text for more details. In the upper table $P_{\text{postproc}} = 0.5$, while in the lower table $P_{\text{reparse}} = 0.3$.

could be calculated based on the values measured in three groups of 250 runs each.

The results appear in Table 7.1 and Fig. 7.6. The differences in the precision across different parameter combinations are significant in most of the cases, demonstrating how important role the *a priori* probabilities play in the SA-OT algorithm. Moreover, an interesting observation has been that the second most frequent candidate, $01[a]0[b]0[d]0[a]0[k]$, has a much more stable chance to be returned, even though there seems to be a major jump between $P_{\text{reparse}} = 0.3$ and $P_{\text{reparse}} = 0.5$. For $P_{\text{reparse}} = 0.9$, this candidate is the most frequent one, but there are also further non-optimal candidates that emerge more frequently than the optimal one.

However, precision varies enormously with hierarchy and input. The next subsection (Tables 7.3 and 7.4) exemplifies the precision’s dependence on the constraint ranking, and preliminary experiments not reported here demonstrated the dependence upon the input. Even for the same hierarchy and input, a different $T_{\text{step}}$ value results in a very different behaviour of the system. For instance, if $T_{\text{step}} = 0.1$, the random walker in the model just discussed is unable to get far enough from the origin in the initial stage of the simulation, therefore heavily underparsed candidates (such as $X[1]X[a]X[b]X[d]X[a]X[k]$) are returned most often. By introducing further post-processing steps, more local optima can be avoided, but the model becomes more complex. It is only to be hoped that a more elegant model will emerge from future research.

### 7.2.3 Syllabification with simulated annealing II.

Now, let us turn to a few further interesting lessons that might be learnt from early, preliminary experiments. These experiments were performed using a
slightly different topology, so first let us describe it.\footnote{This topology involves too many decision points, involving too many (hidden) parameters. Furthermore, it may run into an infinite loop, because it checks the possibility of an operation only after having performed it. These are the reasons why this topology was revised. However, I did not have the time to reproduce all experiments reported here, using the topology described earlier.}

Similarly to the previous model, we first have to decide whether we want to change the length of the candidate (by inserting, deleting or underparsing segments), or to reparse the present structure, that is, to move the syllable borders. Let $P_{\text{reparse}}$ denote again the probability of the latter, whenever possible; whereas $1 - P_{\text{reparse}}$ is the probability of changing the word’s length.

If we have chosen to reparse, we move one syllable border, that is, we reparse randomly one of the intervocalic consonants: if it has been an onset, we reparse it as a coda, and vice versa (for instance, we turn $\text{O[b]N[a]O[b]D[\ldots]}$ into $\text{O[b]N[a]D[b]N[a]D[\ldots]}$).

Alternatively to reparsing, a basic step can either insert an epenthetical vowel or consonant, or delete a phoneme.

Both insertion and deletion of a segment are chosen with a probability of 50%, similarly to an unbiased, one-dimensional random walk. Hence, after $t$ time steps, the expected value of the distance of a random walker’s position from its origin is proportional to $t^{1/2}$. Therefore, if we want our simulated annealing algorithm to reach even remote regions of the search space that correspond to $k$ insertions, we must allow a number of steps proportional to $k^2$ in the first phase of the simulation, when temperature is high and any move is allowed with transition probability 1.

The positions of insertion or deletion are chosen with equal probability again. But from this point onwards come a few differences compared to the topology presented in the previous subsection.

Insertion means only overparsing randomly either an onset, or a nucleus, or a coda, that is, to insert an $\text{O[\ldots]}$, a $\text{N[\ldots]}$ or a $\text{D[\ldots]}$. The three options are chosen with a probability of 40%, 40% and 20%, respectively, making the chance
Table 7.2: The role of \( P_{\text{reparse}} \) and \( P_{\text{postproc}} \) in CV Theory: Percentage of simulations returning the correct parse \( 0[\text{\textup{\textit{N}[\text{\textup{\textit{N}}]}]}][\text{\textup{\textit{t}}}][\text{\textup{\textit{N}}}[\text{\textup{\textit{a}}}]}] \), for \textit{Uranta} and hierarchy \textit{Onset} \( \gg \) \textit{FillNucleus} \( \gg \) \textit{Parse} \( \gg \) \textit{FillOnset} \( \gg \) \textit{NoCoda}. Each \((P_{\text{reparse}}, P_{\text{postproc}})\) parameter combination was run 10 times. The left panel shows the results for different \( P_{\text{reparse}} \) values (average over all possible \( P_{\text{postproc}} \) values), whereas the right panel presents the role of \( P_{\text{postproc}} \) (averaged over all possible values of \( P_{\text{reparse}} \)).

<table>
<thead>
<tr>
<th>( P_{\text{reparse}} )</th>
<th>0.00</th>
<th>0.05</th>
<th>0.10</th>
<th>0.20</th>
<th>0.30</th>
<th>0.40</th>
<th>0.50</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{\text{postproc}} )</td>
<td>0.00</td>
<td>0.65</td>
<td>0.60</td>
<td>0.70</td>
<td>1.00</td>
<td>0.60</td>
<td>0.65</td>
</tr>
<tr>
<td>Location</td>
<td>19</td>
<td>18</td>
<td>15</td>
<td>15</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Probability</td>
<td>0.00</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
</tbody>
</table>

of inserting a consonant slightly higher than inserting a vowel. I acknowledge that these values are fully arbitrary, and further research may investigate the role of that choice. Once it has been decided that we want to overparse an onset (similarly in the case of overparsing a nucleus or a coda), a random position is chosen, and the insertion takes place (unless the result is an ill-formed string). Sometimes, forced reparsing takes place in order to obtain a valid candidate.

If deletion is the operation to perform, then a random element of the candidate is chosen, and depending on its present parse, we alter its status. If it has been a parsed, underlyingly existing element, then it gets underparsed (turned into \( X[.] \)). If it is an epenthetical element, then it gets deleted. Lastly, and differently from the previous model, if it has been underparsed so far, then it gets reparsed (vowels as nuclei, consonants as onsets or codas). This last operation belonged to insertion (lengthening the candidate) in the previous subsection.

Finally, we can “cheat” again by allowing some post-processing, that is some greater changes in the string, larger steps in the search space that improve the word. Let \( P_{\text{postproc}} \) denote again the probability of these changes, whenever possible. These steps help the system to escape from trivial local minima. In our case, we have considered deleting \( 0[\text{\textup{\textit{N}[\text{\textup{\textit{N}}]}]}] \) substrings (whole epenthetical syllables), furthermore, contracting adjacent epenthetical nuclei with underparsed vowels, as well as epenthetical onsets with underparsed consonants. The right panel of Table 7.2 shows how increasing \( P_{\text{postproc}} \) has improved our results.

After having defined the basic step, applying simulated annealing is straightforward. The algorithm presented in Figure 2.8 requires some values for \( T_{\text{max}} \), \( T_{\text{min}} \) and \( T_{\text{step}} \). Now, we used 4, 0 and 0.1 respectively.

Note that \( P_{\text{reparse}} \) and \( P_{\text{postproc}} \) are parameters of the model that determine the a priori probabilities of the topology. The left panel on Table 7.2 reports on some experiments on the role of parameter \( P_{\text{reparse}} \). Significant difference was found only for the highest \( P_{\text{reparse}} \) values. Seemingly, a low value for this parameter does not affect the results too much, because a number of insertions and deletions, involving also some forced reparses (in order to make the string a valid word), can replace reparsing. However, increasing \( P_{\text{reparse}} \) too much will prevent the model from performing the insertions and deletions required in the optimal candidate.
Finally, one may ask whether precision depends on the hierarchy, besides—as we have seen—parameters of the topology (such as $P_{\text{reparse}}$ or $P_{\text{postproc}}$) and of the cooling schedule (such as $T_{\text{step}}$ or $K_{\text{max}}$).

Therefore, simulated annealing was run for the input anta, with parameters $P_{\text{reparse}} = 0.60$ and $P_{\text{postproc}} = 0.95$, and with all the possible 120 rankings. By comparing the outputs to the correct output produced by a finite state technique (Gerdemann and van Noord, 2000), 17 out of the 20 outputs for ranking \text{PARSE} \gg \text{FILLONSET} \gg \text{NOCODA} \gg \text{FILLNUCLEUS} \gg \text{ONSET} were found correct, whereas only 3 for \text{PARSE} \gg \text{FILLONSET} \gg \text{FILLNUCLEUS} \gg \text{ONSET} \gg \text{NOCODA}. For more results, see Tables 7.3 and 7.4.

<table>
<thead>
<tr>
<th>Ranking</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>prs fio fin noc ons</td>
<td>31</td>
</tr>
<tr>
<td>prs fio fin ons noc</td>
<td>26</td>
</tr>
<tr>
<td>prs fio noc fin ons</td>
<td>78</td>
</tr>
<tr>
<td>prs fio noc ons fin</td>
<td>84</td>
</tr>
<tr>
<td>prs fio ons fin noc</td>
<td>14</td>
</tr>
<tr>
<td>prs fio ons noc fin</td>
<td>72</td>
</tr>
<tr>
<td>prs lin fio noc ons</td>
<td>38</td>
</tr>
<tr>
<td>prs lin fio ons noc</td>
<td>25</td>
</tr>
<tr>
<td>prs lin noc fio ons</td>
<td>30</td>
</tr>
<tr>
<td>prs lin noc ons fio</td>
<td>23</td>
</tr>
</tbody>
</table>

Table 7.3: Percentage of correct outputs for different rankings, out of 100 simulations for each.

### 7.2.4 Conclusion

The present chapter aimed at realising the potential in the model introduced in section 6.5. This model had assigned equal \textit{a priori} probabilities to tangential and radial moves, and within the second, to centripetal and to centrifugal moves ($P_{\text{tangential}} = 0.5$, $P_{\text{centripetal}} = 0.25$, $P_{\text{centrifugal}} = 0.25$).\footnote{As mentioned in footnote 17 on page 181, one should reconsider the \textit{a priori} probabilities for unpenthesised candidates in order to fully meet this definition.} Subsection 7.1.3 increased the chance of centrifugal moves, whereas the present section demonstrated the role of $P_{\text{tangential}}$, called here $P_{\text{reparse}}$. As mentioned, not all possibilities have been tried out yet, for only $P_{\text{centripetal}} = P_{\text{centrifugal}}$ has been considered in the present section.

A further connection between the models in sections 7.1 and 7.2 is the high number of local optima. However, it was exactly our goal to have the algorithm be stuck in them in section 7.1, and thereby to account for the observed pause following the definite article in Hungarian; whereas these local optima are not attested in speech in the case of CV-Theory. Thus, we had to introduce some post-processing steps, that is, to enlarge the set of neighbours of some candidates, in order not to make them local optima. Interestingly, these post-processing steps are not reversible, so the new neighbourhood relation is not symmetric any more. The role of parameter $P_{\text{postproc}}$, determining the chance of applying these post-processing steps, has also been analysed.
Table 7.4: For the input anta, the number of correct outputs for different rankings, out of 100 runs. In each hierarchy, the highest ranked constraint appears first. The third column shows the results with parameters $P_{\text{reparse}} = 0.60$ and $P_{\text{postproc}} = 0.95$, while the results in the fourth column was obtained with parameters $P_{\text{reparse}} = 0.60$ and $P_{\text{postproc}} = 0.30$. The correct outputs were calculated using finite state techniques. The precision does not depend on the output either, for there are significant differences between rankings that yield the same optimal candidate.
7.2. Syllabification (CVT) theory

It has been mentioned that implementing the CV Theory for syllabification allows us to compare the SA-OT Algorithm to other computational approaches to Optimality Theory. It should not be surprising by now that SA-OT has a lower computational complexity than most of its competitors, but in exchange, it has a lower precision. It is to be hoped that a new and more elegant topology for CV Theory will be able to increase precision and account for fast speech phenomena, such as dropping of syllables or syllable parts.

Indeed, SA-OT is polynomial in time: quadratic in the number of constraints, and approximately cubic in the length of the word. Evaluating candidates is linear in the number of constraints, and linear in the length of the input word (however, as candidates blow up with inserted materials, the evaluation becomes longer). The more constraints, the higher $K_{max}$, i.e. the interval traversed by $T$. Furthermore, the longer the word, the larger the region to walk through, i.e. the more insertions and deletions to try out. A good estimate for the number of steps required is proportional to the square of the distance we want to walk.

How can one make use of simulated annealing, if it sometimes returns the correct output only in 20% of the cases? One can run more simulations in parallel, and then choose the output returned the most often. This solution would work if the erroneous outputs have an even lower probability, which is true only in part of the SA-OT models.

Another possibility is to compare the few different outputs obtained by parallel simulations, using the classical OT evaluation methods. Although this solution seems to returning to the original methods, it is not the case, for now we only need to compare a few candidates, and not the entire boundless candidate set. In fact, if our algorithm returns the optimal candidate only with a probability of 20%, but we run it ten times, then the optimal candidate will be returned at least once with a chance of almost 90%. However, once the optimal candidate appears in the output set, it will win using traditional tableau comparison of the ten outputs. Running 20 parallel simulations will increase the precision to almost 99%.

It is also possible to combine the previous two solutions: run many simulations in parallel, choose the most frequent outputs, and compare them using a tableau.

If SA-OT is indeed an adequate model of language production, then the observation about the differences in precision across different rankings also has an important consequence for OT’s claim on factorial typology. The traditional approach in a Chomskyan style is that attested and non attested language types result from what the human brain (the Universal Grammar) is able or is unable to realise—that is, the idea of factorial typology in Optimality Theory. Jäger (2003a) has proposed however that some gaps in factorial typology (i.e., a language type predicted by some constraint ranking, but not attested among the languages of the world) may be explained by it being unstable during evolution (across generations). Boersma (2004a) has raised a second option: some constraint rankings may turn out to be not learnable. Now, we may add a third, even more trivial possibility: some constraint rankings might not appear in attested languages, just because they are not producible. That is, SA-OT has only a very low chance to find the correct output. From the observation that some constraint rankings are much more advantageous than other ones, we predict that hard-to-compute rankings are less likely to be attested in the
What happens if the global optimum is relatively hard to reach (because it is located in a narrow valley), whereas some other local optima are returned relatively often? If this is the case for most words, we expect this language not to be stable: the next generation will learn another ranking. But if this phenomenon happens only to a few words, we may predict the surfacing of the sub-harmonic form, seen as either irregularity, or free alternation, or slip of the tongue.