An Approach to the Automatic Acquisition of Phonotactic Constraints

Anja Belz
School of Cognitive and Computing Sciences
University of Sussex
Brighton BN1 9QH, UK
anjab@cogs.susx.ac.uk

Abstract

This paper describes a formal approach and a practical learning method for automatically acquiring phonotactic constraints encoded as finite automata. It is proposed that the use of different classes of syllables with class-specific intra-syllabic phonotactics results in a more accurate hypothesis of a language's phonological grammar than the single syllable class traditionally used. Intra-syllabic constraints are encoded as acyclic finite automata with input alphabets of phonemic symbols. These automata in turn form the transitions in cyclic finite automata that encode the inter-syllabic constraints of word-level phonology. A genetic algorithm is used to automatically construct finite automata from training sets of symbol strings. Results are reported for a set of German syllables and a set of Russian bisyllabic feminine nouns.

1 Background

1.1 Phonotactic Description

In recent years, phonology — partly under the influence of computational models — has moved away from procedural, rule-based approaches towards explicitly declarative statements of the constraints that hold on possible phonological forms. Such statements form sets of constraints that apply at a given level of description, and ill-formedness is often defined as constraint violation.

Phonotactic descriptions state the constraints that hold for possible sequences of phonetic or phonemic features or symbols, usually at the level of the syllable (more rarely for onset, peak and coda separately). Phonological words are defined as sequences of at least one syllable.

A phonotactic description is typically thought to be adequate only if it generalises beyond the set of phonological forms that exist in a language to a superset of possible forms that also includes forms that could exist but do not. This distinction between non-existent but possible forms on the one hand, and non-existent and impossible forms on the other, is often described in terms of accidental vs. systematic gaps (e.g. Carson-Berndsen (1993), Gibbon (1991), and originally Chomsky (1964)).

Carson-Berndsen (1993) lists five encoding schemes for phonotactic description at the syllable level found in the literature: *templates* that merely state the number of consonants permitted in the onset and coda, *distribution matrices* with a separate matrix for each type of consonant cluster, *enhanced templates* which add the notion of phoneme classes, *feature-based phonotactic networks* using feature bundles, natural classes, variables, defaults and underspecification, and *phrase-structure rules* which have the same (potential) representative power as feature-based phonotactic networks.

In the approach presented here, finite state automata (FSA) encoding is preferred over these other schemes, since they can all equivalently be represented by FSAs — what is described is always a regular language — in the most general sense of the term, and since finite-state machinery in itself does not have the disadvantage of necessarily overgenerating forms (as do templates), or of excluding the possibility of multi-tier description (as do most of the above).

1.2 FSAs and Phonotactic Description

FSAs have been used to encode syllable phonotactics e.g. to reduce the search space for lexical hypotheses (Carson-Berndsen, 1993) and to detect unknown words (Jusek et al., 1994) in

speech recognition, but are usually constructed in a painstaking manual process. The aim of the research presented here is to develop a completely automatic method for constructing phonotactic descriptions. This requires a formal theoretical approach to the task as well as a practical automatic inference method. The former is outlined in the following section, while Section 3 describes the genetic algorithm developed for the latter. The remainder of this section briefly summarises the standard FSA notation and definitions used in this paper as well as some non-standard usages.

Following (Hopcroft and Ullman, 1979, p.17), a deterministic finite-state automaton A is a 5-tuple (S, I, δ, s_0, F), in which S is a finite set of states, I is a finite input a alphabet, δ is the state transition function mapping $S \times I$ to S, $s_0 \in S$ is the initial state, and $F \subseteq S$ is the set of final states. For every state $s \in S$ and every symbol $a \in I$, $\delta(s, a)$ is a state. Ordinarily, the δ-notation (sometimes $\hat{\delta}$) is also used for the input of strings $x \in I^*$, such that δ maps $S \times I^*$ to S. The language L accepted by A, denoted $L(A)$, is $\{x | \delta(s_0, x) \in F\}$.

The transition function $\delta(s, a)$ is often represented as a 2-dimensional state transition matrix, and (in contrast to most related research which uses sets of production rules of the form $s_1 \rightarrow as_2$) this matrix is used to represent FSAs in the learning method described in Section 3.

The term FSA is taken to refer to n-level transducers, where the input alphabet consists either of individual symbols $a \in I$ or of strings $x \in I^*$. Although only experiments for 1-level transducers with labels $a \in I$ have been carried out so far, the approach will be extended to the general case, permitting multi-tier phonological description. Another type of FSA is used which can be considered a further generalisation step, i.e. FSAs where the transitions are themselves FSAs.

2 Formal Approach to the Automatic Acquisition of Phonotactics

2.1 Syllable Classes

The phonological word is usually defined as a sequence of syllables, in fact not taking this general approach would mean ignoring a basic phonological regularity (the standard arguments in favour of the syllable are summarised e.g. in Blevins (Blevins, 1994)). Phonological description has, as a rule, described syllables in terms of a single structure consisting of smaller units of description (usually onset, peak and coda) on which certain constraints hold, and words as sequences of one or more occurrences of this structure, on which by assumption no further constraints hold. In many languages, however, word-initial and/or word-final consonant clusters differ from other consonant clusters with regard to (co-)occurrence constraints. Goldsmith (1990, p. 107ff) lists several examples from different languages. This has resulted in the use of the notion of extrasyllabicity to account for 'extra' consonantal segments at the beginnings and the ends of words. Similar problems occur with regard to tonal and metrical regularities, where the first and/or the last vowels in words are often referred to as 'extratonal' and/or 'extrametrical'[1].

There are two problems here. The first is that if a phonological theory assumes a single syllable class for a language and if the language has idiosyncratic word-initial and word-final phonotactics, then the set of possible words that the theory hypothesises is necessarily too large, and includes words that form systematic (rather than accidental) gaps in a language.

The second problem is that if extrasyllabicity is used to reduce the first problem, then the resulting theory of syllable structure fails to account for everything that it is intended to account for, and is forced to integrate extrasyllabic material directly at the word level.

Furthermore, it is likely that all languages display some phonological idiosyncracy at the beginnings and/or ends of phonological words. For these reasons, it seems more practical to make the general assumption that a word is of the form $S_I S_M{}^* S_F$ (where S_I stands for initial syllable, S_M for medial syllable, and S_F for final syllable[2]. These basic syllable classes with different associated sets of phonotactic constraints enable the integration *at the syllable level* of seg-

[1] E.g. in the case of Kirundi, where words with an initial vowel have no tone assigned to the first vowel by word-level phonology, and in Central Siberian Yup'ik where final syllables are never stressed (Goldsmith, 1990, p. 29 and p.179 respectively).

[2] I am grateful to John Goldsmith for advice on this matter.

Total number of word forms: *408,603*

Uniquely occurring syllables:

TOTAL:	9,851
Initial:	3,851
Medial:	3,858
Final:	7,119
Monosyllabic words:	5,120

Intersections:

	Medial	*Final*	*Mono*
Initial	2,626	1,478	1,394
Medial		1,946	1,187
Final			3,877

Initial ∩ *Medial* ∩ *Final* : 1092

Initial ∩ *Medial* ∩ *Final* ∩ *Mono* : 728

Hypothesised sets of possible German words:

length	S^+	$S_I S_M{}^* S_F$ (S_{mono})	*CELEX*
1	9,851	5,120	5,120
2	$9.7 * 10^7$	$2.74 * 10^7$	54,754
3	$9.55 * 10^{11}$	$1.05 * 10^{11}$	105,031
4	$9.41 * 10^{15}$	$4.08 * 10^{14}$	90,278

Figure 1: German syllable statistics (from CELEX).

ments traditionally accounted for by extrasyllabicity, and result in a more accurate hypothesis of a language's set of possible words. Monosyllabic words — often highly idiosyncratic[3] — may have to be accounted for separately, by a syllable class S_{mono}.

Consider as an example the syllable statistics from the German part of the lexical database CELEX (Baayen et al., 1995) shown in Figure 1. The statistics of set sizes and intersections suggest that 4 syllable classes are needed for German (initial, medial, final and monosyllables). Hypotheses for possible German words based on a single syllable class (S^+) would arrive at much larger word sets than a hypothesis based on 4 syllable classes, and because it overgenerates, the theory would not reflect some of the phonotactic constraints that the statistics suggest hold for German.

In addition to word-initial and word-final position, syllables may have idiosyncratic phonotactics as a result of tone and stress effects[4]. It therefore seems natural to propose language-specific **syllable class systems** where each class has its own set of phonotactic constraints (intra-syllabic constraints) assigned to it. Words can then be defined as sequences of syllables, where language-specific 'syllable-tactics' (inter-syllabic constraints) constrain the possible combinations of syllables from different classes, and hence the possible phonological forms of words[5].

2.2 Syllabic Sections

For an automatic method of constructing phonotactic descriptions, the syllable as a unit of description is problematic in that the methods available for syllabification have recourse to morphological knowledge, an underlying, more abstract, underspecified level of description, and/or involve the notion of extrasyllabicity, and generally tend to require an amount of prior knowledge of language-specific phonotactics that is unacceptable where the aim is to discover these very constraints automatically.

The main problems with syllabification arise from difficulties in assigning consonantal segments to exactly one syllable, or drawing unambiguous syllable boundaries between adjacent codas and onsets. Locating syllable peaks, or dividing lines between vocalic and consonantal segments (distinguishable in the acoustic signal) is less problematic, and the approach to word segmentation proposed here involves utilising the relative ease with which peak boundaries can be located[6]. This requires the introduction of the term **syllabic section** to describe a grouping of phonological segments consisting of a peak and the consonantal material between it and either the preceding or the following peak. While the resulting sections are not syllables in the traditional sense, they are syllabic in that they form single stress and tone-bearing units.

[3] Dafydd Gibbon, personal communication.

[4] An example from Russian is that the vowel /o/ only occurs in the peak of stressed syllables.

[5] Stress and tone effects are ignored in this paper, as they are the subject of ongoing research.

[6] Ambiguous material such as glides on peak boundaries poses no problem as long as it is consistently grouped either with the peak or with the surrounding consonantal material.

2.3 Learning Task

Phonological words are thus analysed in terms of intra-syllabic and inter-syllabic constraints as described in Section 2.1, while the traditional syllable is replaced by syllabic sections for reasons outlined in the last section.

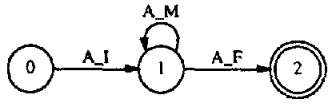
In some languages (such as German, e.g. in the analysis from (Jusek et al., 1994) shown in Figure 3) only peak and coda constrain each other, while in other languages (such as Russian) only onset and peak are mutually constrained (e.g. Halle, (1971)). The third possibility is that both types of constraints occur in the same language.

In order to allow for all three possibilities, the following approach is taken: each word in a given training sample is scanned and segmented in two ways, once by division before the peak and once after. This two-way word segmentation results in the following two analyses:

Initial	Medial	Final
$(c_I{}^*)$	$(v_M{}^+c_M{}^*)^*$	$(v_F{}^+c_F{}^*)$
$(c_I{}^*v_I{}^+)$	$(c_M{}^*v_M{}^+)^*$	$(c_F{}^*)$

In both cases, the initial and final sections together contain three subsections that can be interpreted as the onset, peak and coda of a traditional syllable, which makes it possible to use the same analysis to account for words of arbitrary length, including monosyllables if appropriate. This approach also has the advantage that it can incorporate constraints that cross the boundaries of traditional syllables, such as assimilation phenomena.

For a given training sample of words, in the first scan, all initial syllabic sections resulting from the word segmentation described above are grouped together in data set D1, all final sections in data set D3, and all remaining sections (regardless of how many result from each word) in D2. The same process results in data sets D4–D6 for the second scan. The learning task is then to automatically construct an acyclic FSA on the basis of each data set, resulting in six automata A1–A6. Two cyclic automata C1 and C2 are then constructed (corresponding to the two scans) that have the following structure, where A1 and A4 correspond to A_I, A2 and A5 to A_M and A3 and A6 to A_F:



The final result is a hypothesis of the (word-level) phonological grammar of a given language, based on a given training sample, encoded by the intersection of C1 and C2 (i.e. a word has to be accepted by both in order to be considered well-formed). The present discussion is restricted to a basic syllable class system, but it is likely that descriptive accuracy can be further improved by extending this basic system to include tone and stress effects. This would of course result in more complex automata C1 and C2 (trivially inferrable here).

3 Learning Method

3.1 Background

The Grammatical Inference Problem Generally, the problem considered here is that of identifying a language L from a fixed finite sample $D = (D^+, D^-)$, where $D^+ \subseteq L$ and $D^- \cap L = \emptyset$ (D^- may be empty). If D^- is empty, and D^+ is structurally complete with regard to L, the problem is not complex, and there exist a number of reliable inference algorithms. If D^+ is an arbitrary strict subset of L, the problem is less clearly defined. Since any finite sample is consistent with an infinite number of languages, L cannot be identified uniquely from D^+. "...the best we can hope to do is to infer a grammar that will describe the strings in D^+ and predict other strings that in some sense are of the same nature as those contained in D^+", (Fu and Booth, 1986, p.345).

To constrain the set of possible languages L, the inferred grammar is typically required to be as small as possible, in accordance with a more general principle of machine learning which holds that a solution should be the shortest or the most economical description consistent with all examples, as e.g. suggested in Michalski (1983). However, the problem of finding a minimal grammar consistent with a given sample D was shown to be NP-hard by Gold (1978). Li & Vazirani (1988), Kearns & Valiant (1989) and Pitt & Warmuth (1993) have added

nonapproximability results of varying strength. In the special case where D contains all strings of symbols over a finite alphabet I of length shorter than k, a polynomial-time algorithm can be found (Trakhtenbrot and Barzdin, 1973), but if even a small fraction of examples is missing, the problem is again NP-hard (Angluin, 1978).

Genetic Search Given the nature of the inference problem, a search algorithm is the obvious choice. Genetic Algorithms (GAs) are particularly suitable because search spaces tend to be large, discontinuous, multimodal and high-dimensional. The power of GAs as general-purpose search techniques derives partly from their ability to efficiently and quickly search large solution spaces, and from their robustness and ability to approximate good solutions even in the presence of discontinuity, multimodality, noise and highdimensionality in the search space. The most crucial difference to other general-purpose search and optimisation techniques is that GAs sample different areas of the search space simultaneously and are therefore able to escape local optima, and to avoid poor solution areas in the search space altogether.

Related Research A number of results have been reported for inference of regular and context-free grammars with evolutionary techniques, e.g. by Zhou & Grefenstette (1986), Kammeyer & Belew (1996), Lucas (1994), Dupont (1994), Wyard (1989) and (1991). Results concerning the inference of stochastic grammars with genetic algorithms have been described by Schwehm & Ost (1995) and Keller & Lutz (1997a) and (1997b) describe. Much of this research bases inference on both negative and positive examples, and no real linguistic data sets have been used. Genotype representation is always based on sets of production rules, and knowledge of the target grammar is often utilised. Of these, Zhou & Grefenstette is the one approach directly comparable to the present method, and some comparative results are given in Section 4.

3.2 The Genetic Algorithm

The present algorithm[7] maintains a population of individuals represented by genotypes of varying length which are initialised to random gene values and length. In the iteration typical of GAs, individuals are (1) *evaluated* according to the fitness function, (2) *selected* for reproduction by a process that gives fitter individuals a higher chance at reproduction, (3) offspring are created from two selected individuals by *crossover* and *mutation*, and (4) weaker parents are replaced by fitter offspring. These steps are repeated until either the population converges (when the genotypes in the population have reached a degree of similarity beyond which further improvement is impossible), or the nth generation is reached.

The remainder of this section outlines the fitness function (corresponding to the evaluation function common to all search techniques), and describes how generalisation over the training set is achieved. Full details of the GA can be found in Belz & Eskikaya (1998).

Fitness Evaluation The fitness of automata is evaluated according to 3 fitness criteria that assess (C_1) *consistency* of the language covered by the automaton with the data sample, (C_2) *smallness* and (C_3) *generalisation* to a superset of the data sample. For the evaluation of C_1, the number of strings in the data sample that a given automaton parses are counted. Partial parsing of prefixes of strings is also rewarded, because the acquisition of whole strings by the automata would otherwise be a matter of chance. Size (C_2) is assessed in terms of number of states, the reward being higher the fewer states an FSA has. This criterion serves as an additional pressure on automata to have few states, although the number of states and, more explicitly, the number of transitions is already kept low by crossover and mutation. Generalisation (C_3) is directly assessed only in terms of the size of the language covered by an automaton, where the reward is higher the closer language size is to a specified target size (expressing a given degree of generalisation).

When the goodness of a candidate solution to a problem, or the fitness of an individual, is most naturally expressed in terms of several criteria, the question arises how to combine these criteria into a single fitness value, or, alternatively, how to compare several individuals according to several criteria, in a way that accurately reflects the goodness of a candidate solu-

[7] The algorithm described here was developed in collaboration with Berkan Eskikaya, School of Cognitive and Computing Sciences, University of Sussex.

tion. In the present context, trial runs showed that the structural and functional properties of solution automata are very directly affected by each of the three fitness criteria described above. Therefore, it was most natural to normalise the three criteria to make up one third of the fitness value each, but to attach weights to them which can be manipulated (increased and decreased) to affect the structural and functional characteristics of resulting automata.

Raising the weight on a fitness criterion (increasing its importance relative to the other criteria) has very predictable effects, in that the criterion with the highest weight is most reliably satisfied. Lowering the weight on C_3 towards 0 has the result that language size becomes unpredictable, while lowering the weight on C_2 simply increases the average size of the resulting automata. The weight on C_1 tends to have to be increased with increasing sample size.

Generalisation There are two main parameters that influence the degree of generalisation a given population achieves: the fitness criteria of size (C_2) and degree of overgeneration (C_3). C_2 encourages automata to be as small as possible, which — in the limit — leads to universal automata that parse all strings $x \in I^*$. This is counterbalanced by C_3 which limits the number of strings not in the training set which automata are permitted to overgenerate. To control the quality of generalisation, transitions that are not used by any member of the training set are eliminated, because automata would otherwise accept arbitrary strings in addition to training set members to make up the required target language size.

The overall effect is that a range of generalisation can be achieved over the training set, from precise training set coverage towards universal automata, while meaningless overgeneration of strings is avoided. When $L(A) =$ training set, only symbols $a \in I$ with identical distributions in the data set can be grouped together on the same transition between 2 states. As the required degree of generalisation increases, symbols with the most similar distributions are grouped together first, followed by less similar ones.

Figure 2 shows an example of what effects can be achieved in the limit. The bottom diagram is part of the best automaton discovered for the second half of the German reduced syllable set, shown in Figure 4. Here, the degree of overgeneration was set to 1 (i.e. $L(A) =$ training set), and the size criterion C_2 had a small weight. This resulted in generalisation being completely absent, i.e. the automaton generates only nasal/consonant combinations that actually occur in the data set.

The top diagram in Figure 2 shows the effect of having a large weight on the size criterion, and increasing target language size. The nasals were consistently grouped together under these circumstances, because there is a higher degree of distributional similarity (in terms of the sets of phonemes that can follow) between `m`, `n`, `N` than between these and other phonemes. This achieves the effect that strings not in the data set can be generated in a linguistically useful way, but also may have the side-effect that rarer phoneme combinations (`m[pf]`, `n[ts]`, etc.) are not be acquired, an effect that is described in (Belz, 1998).

4 Results

This section summarises the results that have been achieved for complete presentation of data for finite languages (Section 4.1), and incomplete presentation of data for finite (Section 4.3) and infinite languages (Section 4.2).

The last example (Section 4.3) illustrates how the GA method can be used in conjunction with the formal approach described in Section 2 to automatically discover word-level phonotactic descriptions from raw data sets of phoneme strings.

4.1 German Syllables

Here, the aim was to discover an FSA that accepts a known finite language precisely, so that (following preliminary small-scale tests) the efficiency of the algorithm could be assessed against a medium-sized known language. The data set was generated with the finite-state automaton used by Jusek et al. (1994) to represent the phonotactics of reduced German syllables (shown in Figure 3, double circles indicate final states). This automaton accepts a language of around 11,000 syllables, and because a training set of this size is computationally infeasible, the automaton was divided in half (where the first half covers phonemes preceding the syllable

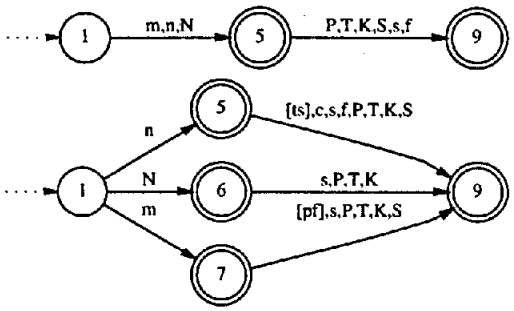


Figure 2: Effect of generalisation on phoneme groupings.

peak, and the second half covers the remainder), and the corresponding strings were used as separate training sets (the training set for the first half contained 127 strings, that for the second 82).

Results are summarised in Table 1, and the best automaton that was found is shown in Figure 4. Algorithms based on Hopcroft & Ullman (1979, equivalence proofs pp. 26–27 and 22–23, minimisation algorithm p.70) to eliminate ϵ-transitions, determinise and then minimise the original manually constructed automaton showed that the automatically discovered FSAs are the minimal deterministic[8] equivalents of the two halves of the original automaton, and therefore represent optimal solutions.

These results show that the GA is able to locate optimal solutions (although it is not guaranteed to find them), and that once an optimum has been found, the population also converges on it, rather than on another, suboptimal solution.

4.2 Some Non-linguistic Examples

In order to assess the performance of the GA on known infinite languages, and to compare it to another GA-based technique with similar aims, experiments were carried out for four previously investigated learning tasks (Zhou and Grefenstette, 1986). Task 1 was to discover the language (10)*, Task 2 was 0*1*0*1*, Task 3 was "all strings with an even number of 0's and an even number of 1's", and Task 4 was "all strings such that the difference between the number of 1's and 0's is 3 times n (where n is an integer)". Zhou & Grefenstette used both positive and negative examples, and moreover had to "modify the training examples to make the genetic algorithm 'understand' what a perfect concept should be" (p.172). The present approach used positive data only, consisting of all strings up to a certain length L, resulting in datasets of varying size S. L was incremented from 0 until the first value was found for which 5 random runs produced the target automaton in under 200 generations.

Task	'L'	'S'	Target found at generation		Z&G Trials
			Best	Avg	
1	4	2	1	2	980
2	7	161	23	29	1027
3	6	42	87	110	2820
4	4	10	69	90	2971

Table 2: Results for non-linguistic examples.

In all four cases, the task was to generalise over the data set, by discovering the recursive nature of the target language from a small subset of examples. Results are summarised in Table 2. Zhou & Grefenstette measure the amount of time it takes for a target automaton to be discovered in terms of 'number of trials', but fail to explain exactly what this refers to. It can probably be assumed to refer to the number of generations, as this is the way genetic search performance is usually measured. Given this interpretation, the present method outperformed the earlier approach in all cases. However, the main point is that the method described here discovers the target FSAs without reference to negative examples or manipulation of the data samples. This second set of results shows that

[8]The 2-dimensional transition matrices that are used as genotypes encode only deterministic FSAs.

		Consistency	Overgeneration (Number of strings)	States	Transitions	Best found at generation
Front	Best	127(100%)	0	3	36	78
	Average	127(100%)	1	2.6	34.0	93
Back	Best	82(100%)	0	9	61	360
	Average	82(100%)	0	9.7	61.5	780

Table 1: Results for set of German reduced syllables.

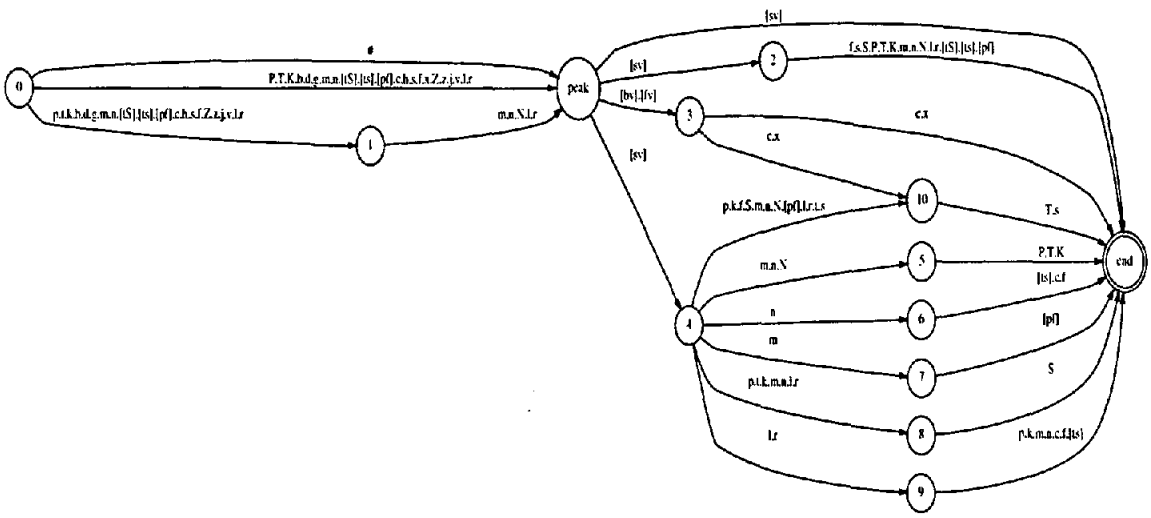


Figure 3: Manually constructed automaton for German reduced syllables.

the GA can find optimal solutions for infinite languages.

4.3 Russian Data

For the third experiment the words in a data set of 450 bisyllabic feminine Russian nouns were divided into sections in the way described in Section 2.3. This resulted in five learning sets D1–D5 (because the set of final consonants is empty in this training set). For each of the five learning sets, five automata were inferred that precisely generate the learning set. On the basis of these, the degree of generalisation that could be expected given the training sets was estimated, and five automata A1–A5 generalising to between 1.5 and 2.5 times the size of their respective learning sets were then evolved. Finally, two automata C1 and C2 were constructed to encode inter-syllabic constraints, with labels A1–A5 representing the automata that encode intra-syllabic constraints:

The resulting phonological grammar hypothesis (the intersection of the two automata encoding inter-syllabic constraints) accepted all words from the original learning set of 450 Russian bisyllabic feminine nouns, generalised to a total set of words of around 10 times this size, and accepted ca. 85% of nouns from a testing set of 200 different such nouns. Generalisation was almost always meaningful in that the greater the similarity between two phonemes (in terms of the phoneme sequences that can follow them), the higher was the likelihood that they were grouped together on the same transition.

5 Conclusion and Further Research

This paper introduced a formal theoretical approach to the automatic acquisition of phonotactic constraints encoded as finite-state automata, and described a genetic-search method for the construction of such automata. Results show that the method is reliably successful in constructing FSAs that accurately cover training samples and allow a range of generalisa-

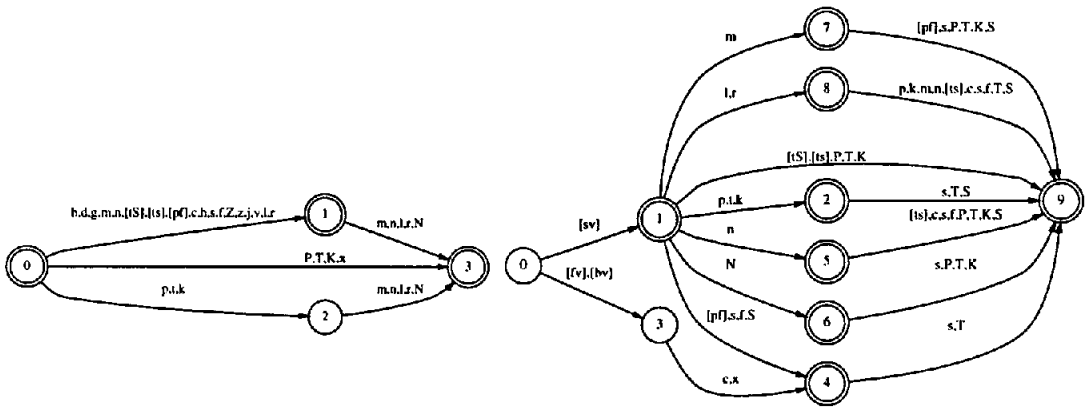


Figure 4: Best automatically discovered automata for German reduced syllables.

tion over the learning samples. The approach to phonotactic description involving several syllable classes that is proposed in this paper is likely to enable a more accurate account of possible phonological forms in a language than approaches that assume a single syllable class. Future research will focus on developing word-level phonotactic descriptions for larger datasets of German and Russian words, and extending the approach to descriptions incorporating tone and stress effects.

References

D. Angluin. 1978. On the complexity of minimum inference of regular sets. *Information and Control*, 39:337–350.

R. H. Baayen, R. Piepenbrock, and L. Gulikers, editors. 1995. *The CELEX Lexical Database (CD-ROM).* Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

A. Belz and B. Eskikaya. 1998. A genetic algorithm for finite-state automaton induction. Cognitive Science Research Paper 487, School of Cognitive and Computing Sciences, University of Sussex.

A. Belz. 1998. A few English words can help improve your Russian. In Henri Prade, editor, *ECAI98: 13th European Conference on Artificial Iintelligence.* John Wiley & Sons.

J. Blevins. 1994. The syllable in phonological theory.

J. Carson-Berndsen. 1993. An event-based phonotactics for German. Technical Report ASL-TR-29-92/UBI, Fakultaet fuer Linguistik und Literaturwissenschaft, University of Bielefeld.

N. Chomsky. 1964. *Current Issues in Linguistic Theory.* Mouton, The Hague.

P. Dupont. 1994. Regular grammatical inference from positive and negative samples by genetic search: the GIG method. In *Grammatical Inference and Applications, Second International Colloquium, ICGI-94, Proceedings*, Berlin. Springer.

K. S. Fu and T. L. Booth. 1986. Grammatical inference: Introduction and survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:343–375. (Reprinted from 1975.).

D. Gibbon. 1991. Lexical signs and lexicon structure: Phonology and prosody in the ASL-lexicon. Technical Report ASL-MEMO-20-91/UBI, University of Bielefeld.

E. M. Gold. 1978. Complexity of automaton identification from given data. *Information and Control*, 37:302–320.

John A. Goldsmith. 1990. *Autosegmental and Metrical Phonology.* Blackwell, Cambridge, Mass.

M. Halle. 1971. *The Sound Pattern of Russian.* Mouton, The Hague.

J. E. Hopcroft and J. D. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, Mass.

A. Jusek, H. Rautenstrauch, G. A. Fink, F. Kummertand G. Sagerer, J. Carson-Berndsen, and D. Gibbon. 1994. Detektion

unbekannter Woerter mit Hilfe phonotaktischer Modelle. In *Mustererkennung 94, 16. DAGM-Symposium.*

T. E. Kammeyer and R. K. Belew. 1996. Stochastic context-free grammar induction with a genetic algorithm using local search. Technical Report CS96-476, Cognitive Computer Science Rsearch Group, Computer Science and Engineering Department, University of California at San Diego.

M. Kearns and L. G. Valiant. 1989. Cryptographic limitations on learning boolean formulae and finite automata. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 433–444, New York. ACM.

B. Keller and R. Lutz. 1997a. Evolving stochastic context-free grammars from examples using a minimum description length principle. Nashville, Tennessee, July. Paper presented at the Workshop on Automata Induction Grammatical Inference and Language Acquisition, ICML-97.

B. Keller and R. Lutz. 1997b. Learning SCFGs from corpora using a genetic algorithm. In *ICANNGA97.*

M. Li and U. Vazirani. 1988. On the learnability of finite automata. In *Proceedings of the 1988 Workshop on Computational Learning Theory*, pages 359–370, San Mateo, Ca. Morgan-Kaufmann.

S. Lucas. 1994. Context-free grammar evolution. In *First International Conference on Evolutionary Computing*, pages 130–135.

R. S. Michalski. 1983. A theory and methodology of inductive learning. In R. Michalski, K. Carbonell, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach Vol. 1*, pages 83–143. Tioga, Palo Alto, CA. Also published by Springer in 1994.

L. Pitt and M. K. Warmuth. 1993. The minimum consistent DFA problem cannot be approximated within any polynomial. *Journal of the Association for Computing Machinery*, 40(1):95–142.

M. Schwehm and A. Ost. 1995. Inference of stochastic regular grammars by massively parallel genetic algorithms. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 520–527. Morgan Kaufmann.

B. B. Trakhtenbrot and Ya. Barzdin. 1973. *Finite Automata.* North Holland, Amsterdam.

P. Wyard. 1989. Representational issues for context free grammar induction using genetic algorithms. Technical report, Natural Language Group, Systems Research Division, BT Laboratories, Ipswich, UK.

P. Wyard. 1991. Context free grammar induction using genetic algorithms. In Richard K. Belew and Lashon B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 514–518, San Diego, CA. Morgan Kaufmann.

H. Zhou and J. J. Grefenstette. 1986. Induction of finite automata by genetic algorithms. *Proceedings of the 1986 International Conference on Systems, Man and Cybernetics*, pages 170–174.