

*author*

**Michal Měchura**

*assignment title*

**Why English may be a regular language after all**

*degree*

**M.Phil. in Speech and Language Processing**

*module*

**LI 7873 Computational Theories of Grammar and Meaning**

*term*

**Hilary 2008**

*word count*

**2,501 words**

# Why English may be a regular language after all

*Michal Měchura*

This essay tackles the question whether English is or is not a regular language, both weakly and strongly, where the term “regular language” is understood to mean a language that can be generated and recognized by a finite-state machine, and equivalently, a language belonging to the Type-3 family of languages in the Chomsky Hierarchy (Chomsky 1959). First, I will review proofs of the non-regularity of English that are based on embedding and recursion. Then, I will observe that human memory limitations largely render such proofs inconclusive and provide evidence for the hypothesis that English is, after all, weakly regular. However, the main thesis of this essay is that, even though English may be weakly regular, it is not strongly regular. In other words, even if a regular grammar might in principle be written that could generate and recognize all and only the grammatical sentences of English, such a grammar could never generate descriptively adequate syntax trees. Last but not least, the arguments presented here can be extended easily to other natural languages.

## **1. Arguments against the regularity of English**

All arguments against the regularity of English make use of recursion in one way or another. For example, the fact that English clauses can contain embedded clauses is extended into the assumption that this embedding may continue indefinitely, and this serves as evidence for the proof that English is not regular. The following two subsections demonstrate two such proofs.

### *1.1. Embedding*

Partee *et al.* (1993, pp. 477–478) present a classic example of a proof that English is not regular. They state that English sentences can contain embedded clauses that

can contain further embedded clauses, as in this gradually expanding series of examples:

- (1) (a) The dog died.
- (b) The dog the cat chased died.
- (c) The dog the cat the mouse hated chased died.

This behaviour can be modelled as a language  $L$ , specified in the following way:

$$(2) \quad \{L = a^n b^{n-1} [died] \mid n \geq 1\}$$

where  $a \in A$ ,  $A = \{[the\ cat],[the\ dog], \dots\}$   
 $b \in B$ ,  $B = \{[chased],[hated], \dots\}$

The language  $L$  is the result of intersecting English ( $E$ ) with the regular language  $R$ :

$$(3) \quad L = E \cap R, \quad R = \{a^* b^* [died]\}$$

If  $L$  is a regular language, then so must be  $E$  (English), because regular languages are closed under intersection. Consequently, if  $L$  is not a regular language, the neither is English.  $L$  can be demonstrated to be non-regular by the pumping lemma for regular languages.

The pumping lemma for regular languages states that, if  $L$  is a regular language, then there is a constant  $n$  such that for every string  $w$  in  $L$  which is longer than  $n$ , it is possible to break  $w$  into three substrings  $w = xyz$  such that  $y \neq \varepsilon$ ,  $|xy| \leq n$ , and for every  $k \geq 0$ , the string  $xy^k z$  is also in  $L$  – that is, the substring  $y$  can be “pumped” (Hopcroft *et al.* 2007, p. 128).

We can prove that the pumping lemma does not hold for  $L$  by imagining that we have found a string  $w$  longer than  $n$  and then trying to break  $w$  into the three substrings  $xyz$  while satisfying the conditions. This proof follows Partee *et al.* (1993, pp. 470–471). If we break  $w$  into  $xyz$ , what would  $y$  consist of? There are only three possibilities:

- $y$  consists of some number of  $a$ 's, that is, elements from the set

$A = \{[the\ cat],[the\ dog], \dots\}$ . If we now pump  $y$ , the resulting string will contain a larger number of  $a$ 's than the number of  $b$ 's minus one, and will therefore not be in  $L$ .

- $y$  consists of some number of  $b$ 's, that is, elements from the set  $B = \{[chased],[hated],...\}$ . If we now pump  $y$ , the resulting string will contain a larger number of  $b$ 's than the number of  $a$ 's plus one, and will therefore not be in  $L$ .
- $y$  consists of some number of  $a$ 's followed by some number of  $b$ 's. If we now pump  $y$ , the resulting string will contain some occurrences of  $b$ 's that precede  $a$ 's, and will therefore not be in  $L$ .

Thus, whichever way we subdivide  $w$ , it cannot be pumped in the way the pumping lemma for regular languages requires. The pumping lemma has the logical form  $R \rightarrow P$ , meaning "if the language is regular, then it can be pumped". By natural deduction, if a language cannot be pumped, then it is not regular:  $\neg P \rightarrow \neg R$ . Since the language  $L$  cannot be pumped, it is not regular. Finally, since  $L$  has been obtained by intersecting English with a regular language, then English is not regular either.

## 1.2. Dependencies

A feature of English and other languages invoked by Chomsky (Chomsky 1956 p. 109) in support of an argument against the regularity of natural languages is the presence of nested dependencies between various words, for example between *if* and *then* and between *either* and *or*:

(4) If you either phone or e-mail us by Friday, then we will give you a discount.

This behaviour can be modelled as the language  $L$  in which the  $i$ th symbol from the left must match the  $i$ th symbol from the right:

$$(5) \quad L = \{xx^R \mid x \in \{a,b\}^*\}$$

If we now intersect  $L$  with the regular language  $\{aa^*bbaa^*\}$  we obtain a new language  $L' = \{a^n b^2 a^n \mid n \geq 1\}$ . This language can be proven to be non-regular by the pumping lemma for regular languages (the details of the proof will not be presented here). Since  $L'$  is not a regular language, and since regular languages are closed under intersection, then  $L$  is not a regular language either.

Note that this merely proves that  $L$ , as specified above in (5), is not regular. Strictly speaking, it does not prove that English is not regular. To prove that, one would need to find a relationship between English and  $L$ , such as an intersection or a homomorphism, that could not be disturbed by the fact that words like *if* and *or* have many uses, some of which do not require them to be paired with *then* and *either*. Nonetheless, it constitutes strong evidence in favour of the non-regularity of English.

Note as well that dependencies do not need to be nested in order to provide this sort of evidence. Crossing dependencies have been demonstrated to exist in several languages, including Swiss German (Shieber 1985). A crossing dependency can be modelled by a language of the form:

$$(6) \quad L = \{(a | b)^* \mid \text{there are as many } a\text{'s as there are } b\text{'s}\}$$

Such a language can also be proved to be non-regular (in fact, it can even be proven to be non-context-free, see Shieber 1985). Thus, the existence of any dependency combined with recursion can provide evidence against the regularity of a language.

## 2. Recursion and human memory

Both the proofs outlined above are valid – but are they based on true premises? They both presume that recursion can be applied indefinitely. However, human short-term memory places some limitations on the number of recursive steps that humans can handle in a single sentence.

### 2.1. Recursive embedding

Recursive embedding is a term I will use here to refer to the two phenomena described in the previous section: the presence of embedded clauses as in (1) and of paired dependencies as in (4). What they have in common is that they both introduce dependencies that the parser needs to keep track of while processing the sentence. For example, while processing the sentence *The dog the cat chased died* from start to end, the human language parser probably places the item *dog* in some

form of short-term memory and keeps it there until it reaches its dependent item *died*, with which it can be paired up. The more levels of embedding there are, the more items need to be kept in short-term memory simultaneously. Presumably, there is a limit to the number of items that can be kept in such a memory.

With recursive embedding, human memory appears to have a limit of just one embedded clause: sentences containing more than one embedded clause are hard to understand. Let's take another look at a sequence of sentences which demonstrates that:

- (7) (a) The dog died.
- (b) The dog the cat chased died.
- (c) # The dog the cat the mouse hated chased died.
- (d) # The dog the cat the mouse the cow squashed hated chased died.

That (7a) and (7b) are grammatical is beyond doubt. Whether (7c) and (7d) are grammatical is less clear and depends on exactly how one defines the notion of grammaticalness. In the generative tradition of syntax, sentences like (7c) and (7d) are considered grammatical, but **unacceptable**. Acceptability and grammaticality are separate qualities. Importantly, a sentence can be ungrammatical but still acceptable for some pragmatic reason, such as when a foreign language learner utters an ungrammatical sentence like (8) whose intended meaning is nonetheless reconstructable, at least approximately:

- (8) \* The dog was died.

Thus, we see that a sentence can be ungrammatical but acceptable at the same time. Is the opposite possible too, that is, can a sentence be grammatical but unacceptable at the same time? Sentences (7c) and (7d) certainly seem like good candidates. There is a pre-theoretic intuition that they can be judged grammatical if one takes the time and effort to analyze them, such as by performing some kind of pen-and-paper analysis where one would draw arrows or brackets to ascertain which verbs “go with” with which nouns and that no dependencies are left undischarged.

However, in actual every-day language use, humans do not do such analyses. Kornai (1985) mentions experiments in which native speakers responded in exactly the same way to questionably grammatical sentences like (7c) and (7d) as they did to definitely ungrammatical sentences like (9) where there is a mismatch between the number of verbs and nouns (you will probably need to count them to notice that):

(9) \*# The dog the cat the mouse the cow squashed hated died.

That is, the speakers did not even attempt to analyze the sentences, they objected to them merely on the basis of their complexity. The question of grammaticalness did not enter into it.

## 2.2. Recursive iteration

Recursive iteration is a term I will use here to refer to phenomena which are sometimes called *left branching* and *right branching*. Recursive iteration stands in opposition to recursive embedding in that it does not introduce dependencies. The sets of sentences in (10) and (11) illustrate left branching and right branching, respectively.

- (10) (a) The brother has arrived.  
(b) The sister's brother has arrived.  
(c) The husband's sister's brother has arrived.  
(d) The mother's husband's sister's brother has arrived.
- (11) (a) The cow squashed the dog.  
(b) The cow squashed the dog which chased the cat.  
(c) The cow squashed the dog which chased the cat which troubled the mouse.  
(d) The cow squashed the dog which chased the cat which troubled the mouse which ate the cheese.

Unlike recursive embedding, recursive iteration does not seem to be limited by any obvious short-term memory constraints in the human mind. Speakers might agree that sentences involving a lot of iterations are long-winded, but they remain

acceptable and can be readily judged grammatical or not. For example, the sentence (12) can be recognized as ungrammatical at first reading and without recourse to any form of pen-and-paper analysis, even though it contains many levels of recursion:

(12) \* The cow squashed the dog which chased the cat which troubled the mouse which the cheese.

An important fact is that recursive iteration can be modelled easily by regular languages. The specifications in (13) and (14) demonstrate that:

(13)  $L = \{[The]a*[brother\ has\ arrived]\}$   
 where  $a \in A$ ,  $A = \{[sister's],[brother's],[father's],...\}$

(14)  $L = \{[The\ cow\ squashed\ the\ dog]a*\}$   
 where  $a \in A$ ,  $A = \{[which\ chased\ the\ cat],[which\ troubled\ the\ mouse],...\}$

### 3. English as a weakly regular language

It is interesting that human memory imposes limitations on exactly those features that make English non-regular. Recursive embedding introduces dependencies into the sentence. As the human language parser processes a sentence from start to end, it needs to have access to some form of a stack in which it keeps a record of dependencies that have not been discharged yet. The need for a stack is precisely the quality that makes the language non-regular. Machines called *push-down automata* (Hopcroft *et al.* 2007 ch. 6) also use stacks to generate and recognize non-regular languages. Recursive iteration, on the other hand, introduces no such dependencies and does not require a stack for parsing – therefore, the existence of infinitely recursive iteration does not affect the status of a regular language as regular.

As multiple-level embedding like in (7c) and (7d) is unacceptable in actual English usage because it is too complex for humans to parse, the argument can be made that there is no value in building computational grammars of English that generate and accept such sentences. Arguably, a more realistic model of English

would be obtained by declaring that recursive embedding is limited to a depth of one, such as in the following updated version of the language  $L$  from (2):

$$(15) \quad L = \{a^n b^{n-1} [died] \mid 1 \leq n \leq 2\}$$

where  $a \in A$ ,  $A = \{[the\ cat], [the\ dog], \dots\}$   
 $b \in B$ ,  $B = \{[chased], [hated], \dots\}$

Embedding is here limited to a maximum of one embedded clause. That is, the updated language in (15) includes the sentences (7a) and (7b) but not (7c) and (7d). This yields a language which is a closer match to the abilities of humans. Essentially, this approach changes the definition of language from a set of grammatical sentences to a set of grammatical **and acceptable** sentences. That may seem like a bold move to make, but it is a worthwhile option to consider. Importantly, a language thus constrained is weakly regular because the number of patterns is finite and can be enumerated, as in (16):

$$(16) \quad L = \{a[died], aab[died]\}$$

where  $a \in A$ ,  $A = \{[the\ cat], [the\ dog], \dots\}$   
 $b \in B$ ,  $B = \{[chased], [hated], \dots\}$

#### 4. English as a strongly regular language

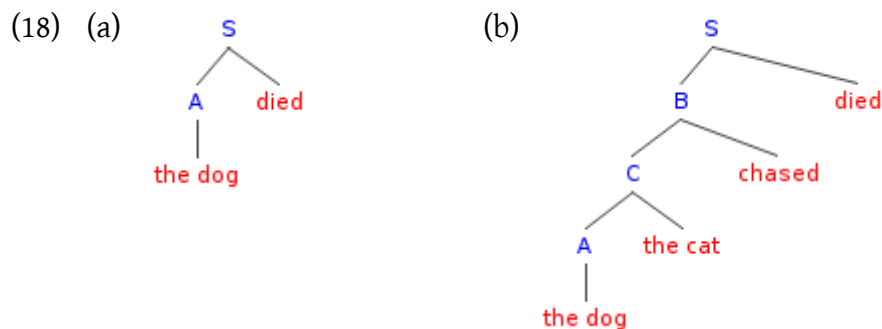
Having established that English can be modelled as a weakly regular language if performance constraints are taken into account, the next obvious question to ask is whether English is strongly regular, too – that is, whether the syntax trees assigned to English sentences by a regular grammar are descriptively adequate (Borsley 1991 pp. 37-38): whether they capture pre-theoretic intuitions about constituent structure, whether they capture all linguistically relevant generalizations, whether they support a compositional semantics, and so on. Unsurprisingly, the answer to that question is no.

It appears that dependencies of the type illustrated in (1a), (1b) and (4) above cannot be modelled by a regular grammar in a way that would display a

descriptively adequate connection between verbs and their subjects. A regular grammar that generates the language specified in (16) would like (17).

- (17)  $S \rightarrow A \text{ [died]} \mid B \text{ [died]}$   
 $B \rightarrow C \text{ [chased]} \mid C \text{ [hated]} \mid \dots$   
 $C \rightarrow A \text{ [the cat]} \mid A \text{ [the dog]} \mid \dots$   
 $A \rightarrow \text{[the cat]} \mid \text{[the dog]} \mid \dots$

For one thing, there is some duplication in this grammar, which is necessary because a regular grammar does not allow more than a single non-terminal symbol on the right-hand side of a rule. That, however, is nothing that could not be solved by some form of rule condensing, similar to the way phrase-structure rules that use features are condensed. A more important negative effect is that the grammar generates descriptively inadequate trees, such as those in (18).



As descriptively adequate is not a formally defined concept, it is possible to argue that trees like (18b) are, in fact, descriptively adequate. Indeed, Reich (1969) seems to be making the point that structural descriptions generated by a regular grammar are in fact adequate. This is an intriguing avenue to explore, however I will not explore it here and conclude with the more intuitive observation that syntax trees generated by a regular grammar are not descriptively adequate and therefore, English is not a strongly regular language.

## 5. Conclusion

This essay has demonstrated that, when human performance limitations are taken into account, English (and possibly all natural languages) is weakly regular, but not strongly regular.

## References

- Borsley, R. (1991) *Syntactic Theory: a unified approach* London: Edward Arnold
- Chomsky, N. (1956) 'Three models for the description of language' in *IRE Trans. Inform. Theor.* IT-2. Reprinted in R. Duncan Luce; R. R. Bush; E. Galanter (eds.) *Readings in Mathematic Psychology, Volume II* London: John Wiley
- Chomsky, N. (1959) 'On certain formal properties of grammars' in *Inform. & Control* 1. Reprinted in R. Duncan Luce; R. R. Bush; E. Galanter (eds.) *Readings in Mathematic Psychology, Volume II* London: John Wiley
- Hopcroft, J. E.; R. Motwani; J. D. Ullman (2007) *Introduction to Automata Theory, Languages, and Computation (3rd edition)* London: Pearson Education
- Kornai, A. (1985) 'Natural languages and the Chomsky Hierarchy' in M. King (ed.) *Proceedings of the second conference on European chapter of the Association for Computational Linguistics* Geneva: ACL. Also available on-line:  
<<http://www.kornai.com/Papers/nlch.pdf>> accessed 23 March 2008
- Partee, B. H.; A. ter Meulen; R. E. Wall (1993) *Mathematical Methods in Linguistics* London: Kluwer Academic Publishers
- Reich, P. A. (1969) 'The finiteness of natural language' in *Language* 45, pp. 831–843
- Shieber, A. M. (1985) 'Evidence against the context-freeness of natural language' in *Linguistics and Philosophy* 8, pp. 333–343