# Towards a new empiricism for linguistics

John Goldsmith
The University of Chicago

## 1 Introduction

### 1.1 The goal of this chapter

The view of linguistics which we will consider in this chapter is empiricist in the sense explored in Chapter One of this book:[1] it is epistemologically empiricist, rather than psychologically empiricist; in fact, it is a view that is rather agnostic about psychology — ready to cooperate with psychology and psychologists, but from a certain respectful distance. It is empiricist in the belief that the justification of a scientific theory must drive deep into the quantitative measure of real-world data, both experimental and observational, and it is empiricist in seeing continuity (rather than rupture or discontinuity) between the careful treatment of large-scale data and the desire to develop elegant high-level theories. To put that last point slightly differently, it is not an empiricism that is skeptical of elegant theories, or worried that the elegance of a theory is a sign of its disconnect from reality. But it is an empiricism that insists on measuring just how an elegant a theory is, and measuring how well it is (or isn't) in sync with what we have observed about the world. It is not an empiricism that is afraid of theories that leave observations unexplained, but it is an empiricism that insists that discrepancies between theory and observation are a sign that more work will be needed, and sooner rather than later. And it is an empiricism that knows that scientific progress cannot be reduced to mechanistic procedures, and even knows exactly why it cannot.

Thus this chapter has four points to make: first, that linguists can and should make an effort to measure explicitly how good the theoretical generalizations of their theories are; second, that linguists must make an effort to measure the distance between their theories's predictions and our observations; third, that there are actually things we working linguists could do in order to achieve those goals; and fourth, that many of the warnings to the contrary have turned out to be much less compelling than they seemed to be, once upon a time.

The perspective spelled out is thus *non*-cognitivist, though by no means anti-cognitivist, and I emphasize to the reader that our empiricism is not one that in any sense prefers data over theory. And I do not wish to argue that this is *the* way, the only way to do linguistics; there are many ways to do linguistics. But in recent decades, and especially within theoretical linguistics, a view has become so widespread that it passes now for uncontroversial: that the reality claimed by a linguistic theory is the same reality claimed by a psychological theory, and I wish to respectfully disagree with that view, and suggest that it is a serious oversimplification, at the very least.

The main focus of this chapter is the notion of Univeral Grammar: henceforth, *UG*. From a methodological point of view, UG is the set of assumptions we bring to the design of a grammar for a language. From a psychological point of view, UG is a model of the initial cognitive state of a language learner before any of the data from a particular language have been made available to the learner. I will focus on the methodological sense, for reasons that we will see, and I will offer my reasons for believing that an empirically empty version of UG is available to us, and may well be just what we need, at least for some parts of linguistics. That sounds a bit mysterious (after all, what could it mean to speak of an *empirically empty* UG?), but this will become clearer as we proceed. The two most important points that I shall argue are, first, that we need a quantitative account of theory-confirmation, and second, that we should not treat theoretical mechanisms that we put in Universal Grammars as cost-free. I will suggest, on the one hand, that probability can be understood as the *quantitative theory of evidence*, and, on the other hand, that probability theory offers us a more comprehensive and concrete way to understand the fundamental problem of induction, which is to say, how one passes from knowledge of a finite number of particulars to a generalization, which, typically, allows us to infer an infinite number of conclusions, almost all of which have not yet been tested. One of the consequences of this perspective is the perhaps surprising principle that the value of a theoretical innovation is neither more nor less than the amount of information it would take to (merely) stipulate its intended consequences.

[1] To appear as chapter 3 in *Empiricist Approaches to Language Learning*, co-authored with Alex Clark, Nick Chater, and Amy Perfors.

## 1.2 The problem of induction, and Universal Grammar

Chomsky's greatest contribution to our understanding of linguistics is the clarity with which he showed the deep connection between the modern problem of induction and the problem of explanation in linguistics. By *the problem of induction*, I mean the problem of justifying the move from knowledge of a finite number of particulars to knowledge of a generalization, especially a generalization which (like virtually all generalizations that interest us) has an infinite number of consequences which follow from it. From the philosopher's point of view, any such inference is tricky business, since there is a serious danger of making a mistake and drawing the *wrong* conclusion from the observations, not because of a logical error, but because any generalization and any prediction will be based on the categories that we use to encode the particular observations that we began with.

From the psychologist's point of view, understanding how people (or for that matter, rats and pigeons) draw inferences is a difficult task, but one which holds the promise of telling us something about otherwise scientifically inaccessible reaches of the human psyche.

From the linguist's point of view, the problem of induction is an abstract way to formulate the most central question of synchronic linguistic analysis: how do we justify the formulation of grammatical statements, valid for a potentially infinite class of representations, on the basis of a finite set of data? What I daresay is the most widespread view in linguistics currently, the principles and parameters approach, is based on the hope that only one element in a restricted class of grammars, those permitted by Universal Grammar, are consistent with the observed data. The alternative, empiricist account is that a careful study of the formal and statistical properties of observable data will lead to what we might think of as a *rating* of grammars which generate the data calculated from a probabilistic model, and that the most probable model is the best one. This paper is an attempt to spell that out in more detail.

Oddly enough, as we will see, what I will describe here is very much in the tradition of classical generative grammar—by which I mean, the research program described in detail in *The Logical Structure of Linguistic Theory* [**?**], but this line of thinking does make one quite skeptical about the principles and parameters approach to grammar. Partisans of principles and parameters should be prepared to jettison some beliefs.

Needless to say, I do not think that the last word in the solution to the problem of induction has been said, and puzzles and serious questions remain. My purpose is to shed a little bit of light on the connection between what linguists do and the worries that have arisen from thinking about the problem of induction; I think that the current emphasis on what Chomsky calls I-language is the result of a misplaced belief that the study of I-language is less beset by methodological and philosophical problems than the study of E-language is. I will try to show why I do not agree with that belief, and why I think that a respectable, E-language-oriented way of doing linguistics is fully justifiable, and leads to perfectly reasonable results: indeed, quite appealing results; see section 8 below.

## 1.3 Empiricism and linguistics

Linguists in the pre-generative period in American linguistics would happily have referred to themselves as empiricists. Foremost in their understanding of the term was the sense that empiricists were highly skeptical of what they viewed as metaphysical claims, and they shared the view that a rigorous method needed to be invoked in order to make scientific claims.

For that reason, linguists expended considerable time, energy, and attention discussing and developing notions of linguistic method. To many generative linguists today, this earlier emphasis on method, along with an apparent lack of concern for something else called "theory," makes empiricist views seem more mysterious than they really are. Empiricist methods have at their core two rough-and-ready principles: first, that the data are what they are, not what the linguist wants them to be, and second, that care must be taken to justify the positing of abstract entities in one's theoretical analysis—or to put the matter another way, while it is fine to be proud to have discovered an unseen object, the burden of proof remains heavy on the scientist who claims to have found one. A direct consequence of this is that alternative analyses in which abstract elements are not posited have to be thoroughly explored to be sure that none of them is as capable of accounting for the evidence.

The abstract element that I would like to skeptically rethink in this paper is a rich Universal Grammar—UG, for short. UG is a central concept in much current thinking about linguistics; it is sometimes conceptualized as the initial state of the human language acquisition device; it is, in any event, the conceptual substance necessary to bridge the gap between the linguistic data presented to the child-learner and the grammar that he ends up with as a fully competent adult native speaker. The question is not whether UG exists; it is rather to what extent UG should be thought of as conceptually rich. The empiricist assumption regarding human knowledge in the golden days of empiricism in the 17th and 18th century was that the mind starts off like a tabula rasa: a blank white-board, so to speak, on which drawings of any kind, and formulas in any language, could be written; UG, on this account, is relatively impoverished. The opposing picture today is one in which the mind starts off more like the control panel of a jet airliner, with a panoply of gauges and dials which need to be set, but whose settings only gain sense and meaning by virtue of the circuitry that lies behind the dashboard.

The main question I will attack in this paper is whether the role played by Universal Grammar can be assigned to a completely abstract and, we might say, platonic object, one that is based on algorithmic complexity, and unrelated to any particular biological characteristics of human beings. I will argue that such a project is feasible, given our present knowledge. In fact, the perspective I am describing here has a stronger claim to being called "universal grammar" than an information-rich UG does, in the sense that the empiricist position described here would be valid in any spot in the known universe, and is not a theory of the human genetic endowment.

In the next section, Section 2, we will introduce the idea of a probabilistic grammar, and sketch out how such a grammar can be thought of within a perspective that incorporates a notion of Universal Grammar, allowing us a Bayesian conception of linguistics. In Section 3, we will discuss the basic logic of classical generative grammar in a way that facilitates comparison with a Bayesian approach, and then return to the exposition of the Bayesian conception in Section 4. ...

# 2 Probabilistic grammars

## 2.1 The basic idea

Generative grammar has always been understood as making claims about a language by virtue of predicting what strings are in a language and what strings are not.[2] This has never been particularly controversial, though in the early years of generative grammar, there was more concern than there is nowadays about the fuzzy status that should be associated with semi-grammatical sentence. Still, most linguists felt that we could make a lot of scientific progress focusing just on the clear cases, the sentences whose grammaticality status was not in doubt (they are good, they are bad), and the goal of the grammar was to generate or enumerate the grammatical ones (typically an infinite set) and to fail to generate any of the ungrammatical ones.

Probabilistic grammars take a different tack: they say that there is another way that we can develop formal grammars and ensure that they come in contact with the real world of linguistic facts. A *probabilistic grammar* assigns a non-negative probability to every predicted outcome, in such a fashion that the probabilities sum to 1.0—neither more, nor less. In most cases of interest, the number of predicted outcomes is infinite, but the same condition holds: the sum of the probabilities of each outcome must be 1.0. For this condition to hold over an infinite set, it must be the case, first of all, that the probabilities get indefinitely small, and it must also be true that while we cannot test every outcome (since there are an infinite number of them), we can always find a finite number of outcomes whose total probability gets arbitrarily close to the probability of the whole set. Thus a probability measure assigned to an infinite set makes it *almost* as manageable as a finite set, while still remaining resolutely infinite.That is the heart of the matter.

We need to be clear right from the start that the use of probabilistic models does not require that we assume that the data itself is in a linguistic sense "variable," or in any sense fuzzy or unclear. I will come back to this point; it is certainly possible within a probabilistic framework to deal with data in which the judgments are non-categorical and in which a grammar predicts multiple possibilities. But in order to clarify the fundamental points, I will not assume that the data are anything except categorical and clear.

Assume most of what you normally assume about formal grammars: they specify an infinite set of linguistic representations, they characterize what is particular about particular languages, and at their most explicit they specify sequences of sounds as well as sequences of words. It is not altogether unreasonable, then, to say that a grammar essentially *is* a specification of sounds (or letters) particular to a language, plus a function that assigns to every sequence of sounds a real value: a non-negative value, with the characteristic that the sum of these values is 1.0. To make matters simpler for us, we will assume that we can adopt a universal set of symbols that can be used to describe all languages, and refer to that set as $\Sigma$.[3]

A grammar, then, is a function $g$ with the properties in (1).

$$\begin{aligned} g &: \Sigma^* \rightarrow [0,1] \\ \sum_{s \in \Sigma^*} & g(s) = 1 \end{aligned} \tag{1}$$

The grammar assigns a probability (necessarily non-negative, but not necessarily positive) to all strings of segments, and these sum to 1.[4]

[2] Putnam 1960 article.

[3] I do not really believe this is true, but it is much easier to express the ideas we are interested in here if we make this assumption. See Robert Ladd, *Handbook of Phonological Theory* vol. 2 for discussion.

[4] If you are concerned about what happened to trees and the rest of linguistic structure, don't worry. We typically assign a probability to a structure, and then the probability assigned to the string is the sum of the probabilities assigned to all of the structures that involve the same string.

A theory of grammar is much the same, at a higher level of abstraction. It is a specification of the set $\mathcal{G}$ all possible grammars, along with a function that maps each grammar to a positive number (which we call its probability), and the sum of these values must be 1.0, as in (2). We use the symbol $\pi$ to represent such functions, and each one is in essence a particular Universal Grammar.

$$\begin{aligned} &\pi : \mathcal{G} \rightarrow [0,1] \\ &\sum_{g \in \mathcal{G}} \pi(g) = 1 \end{aligned} \tag{2}$$

To make thing a bit more concrete, we can look ahead and see that the function $\pi$ is closely related to grammar complexity: in particular, the complexity of a grammar g is $-log\, \pi(g)$; likewise, the function $g$ is closely related to grammaticality; in particular, $-log\, g(s) + \sum_{w \in s} log\, pr(w)$ is a measure of the ungrammaticality of $s$.[5]

If we put no restrictions on the nature of the function $\pi$, then the class of models that we have described so far could include a good deal of what already exists in formal linguistics, once we have probabilized the grammars, so to speak—once we have organizerd our grammars $g_i$ in such a way that they not only generate sentences, they assign probabilities that sum to 1.0 for their sentences. With no constraints on the function $\pi$, most any theory of formal grammar could be described in such terms. Would this be an empiricist conception of linguistics? It would be empiricist to the degree that real scientific work was being done by virtue of testing the model's fit to reality by means of the computation of the probability of the data. But with no restrictions on the form of Universal Grammar (i.e., the choice of $\pi$), the approach could be as nativist as a researcher wanted it to be.

But we are suggesting something a good deal more specific than that. Our claim here is that, *as far as we can see at this point in time*, algorithmic complexity is all that is needed (or *most* of what is needed) in order to specify $\pi$. Thus algorithmic complexity plays the same role in the new empiricism that formal logic played in the old empiricism: it is what we can add to the observed data without becoming nativist, and if we or anyone else are to argue in favor of a nativist position, the argument must be made that the learning that is accomplished by the language learner cannot be accounted for by a combination of data and its analysis using the notions of algorithmic complexity. Or: the task of the grammarian, given a corpus of data $D$, is to find the most probable grammar $g$, and $g$'s probability, in this context, is directly proportional to its probability based on its algorithmic complexity, multiplied by the probability that it assigns to the data $D$. This is the basic idea, then, which we will try to describe in more detail below.

## 2.2 Probability evaluates the grammar, not the data

In order to make this new empiricist interpretation work, we need to understand the notion of a *probabilistic grammar*, developed first by Solomonoff in the 1950s (see Solomonoff 1997 [**?**]). Curiously, the notion is not widely known or appreciated in mainstream linguistics, and my impression is that most linguists think that probabilistic models make vague, soft, or non-categorical predictions. This is false; probabilistic grammars can be put to those purposes, but we will not do so, and there is nothing about probabilistic grammars that requires one to make fuzzy predictions. Rather, what makes a model probabilistic is much more formal and mathematical (see, e.g., Goldsmith 2007 [**?**]).

Like virtually any other formal device, a probabilistic grammar specifies a universe of possible representations for the domain it treats; but in addition, a probabilistic model associates with each representation a non-negative number, its probability, and a strict condition is associated with these probabilities: the sum of the probabilities of all of the representations must be 1.0—neither more nor less. Informally speaking, a probabilistic grammar can be thought of as possessing an infinitely dividable substance, referred to as probability mass, and it doles it out to all of the representations it generates. The goal is to find a grammar that assigns as much of that probability mass as possible to the data that was actually seen. In a sense, this is the crucial difference between the empiricist (and probabilistic) approach and the generative approach: the empiricist, like the rationalist, wants and needs to generate an infinite class of representations, but the empiricist measures the adequacy of the grammar on the basis of how well the grammar treats data that was naturalistically encountered (that is to say, data that was recovered from Nature in an unbiased fashion).

The condition that the sum of the probabilities of all generated representations be equal to 1.0 is trivial in the case where there are a finite number of representations, to be sure. But it is typically not a problem when the representations form an infinite set either. If the reader is uncertain how it can be that an infinite set of positive numbers sum to 1.0, imagine that all the representations are sorted alphabetically, in such a way that shorter ones come first (that is, by treating space as the first element of the alphabet), and then assign probability $2^{-n}$ to the $n^{th}$ word. A moment's thought will convince the reader that these numbers sum to 1.0 ( + $+\frac{1}{8}$ + ...)

[5] Note about normalizing for sentence length.

To repeat, then: the first goal is to find the grammar that maximizes the probability of the observed data (this will be modified slightly, in due time, to allow the simplicity of the grammar to play a role in the selection). Any reasonable theory will assign most of the probability mass to unseen events, that is to say, to sentences that have never been pronounced, and perhaps never will be. That's not a problem. The grammar will not be tested on the basis of those sentences, either: it will be tested on the basis of the probability that it assigns to the sentences that have already been seen.[6]

It should now be clear that the purpose of our insisting that a grammar be probabilistic has nothing to do with evaluating the probability of different sets of data. It would indeed be odd if we were to use a probabilistic grammar to decide what the probability was of various data that had in fact been observed. No; rather, the point of asking different grammars what probability they assign to a single, fixed set of data is to *evaluate the grammars, not the data*. If the data is naturalistic, then we know it exists, but what we care about is evaluating different candidate grammars to see how well they are able (so to speak) to decide which set of data actually exists, and we do this by seeing which grammar assigns the highest probability to the corpus.

We turn now to the subject of Bayesian analysis.

## 2.3 Bayesian analysis

A Bayesian approach to probabilistic modeling is one that takes into consideration not only the probability that is assigned to the data by a model (or as we linguists say, by a grammar), but also the probability of the model (i.e., the grammar). And this latter notion is one that takes us right into the heart of classical generative grammar, to the notion of an evaluation metric. But first we will look at the mathematical side of Bayes's rule, reminding the reader of some of the points we covered in Chapter One.[7]

Bayes's rule involves inverting conditional probabilities, although from a mathematical point of view it is a very simple algebraic manipulation. We need first to state what it means to speak of the probability of an event $X$, given another event $Y$, written $pr(X|Y)$. This means that we consider only those possible situations in which $Y$ is true, and within that set of situations, we calculate what $X$'s probability is. If we select a word at random from English, the probability will be about 8% that it is *"the"*, but if we look only at sentence-initial words, the probability of *"the"*, given that it occurs sentence-initially, is quite a bit higher. The probability that *"the"* occurs, given that it is in sentence-*final* position, is essentially nil.

To calculate such probabilities, when we already have in hand a system which assigns probability mass to all possible representations, we do the following. To determine the probability of $X$, given $Y$, we ask: how much probability mass altogether is assigned to all of the events in which both $X$ and $Y$ are true? And we divide this quantity by the probability mass that is assigned to all of the events in which $Y$ is true. If we want to know the probability of the word *"the"* sentence-initially, then we calculate the probability that *"the"* occurs sentence-initially, and divide by the probability that a random word selected is sentence-initial. That is:

$$pr(X|Y) = \frac{pr(X\,and\,Y)}{pr(Y)} \tag{3}$$

But it will often be the case that we want to invert the dependence, in the following sense. We can calculate the probability that the word *"the"* occurs in sentence-initial position: that is the probability of *"the"*, given that it's in word-initial position. But we may also be interested in knowing, for any given occurrence of the word *"the"*, what the probability is that it is sentence-initial. If the first is $pr(T|I)$, then the second is $pr(I|T)$. Bayes's rule is the formula that relates these two quantities.

Expression (3) can be rewritten as (4), and a moment's thought shows that if we interchange the symbols *"X"* and *"Y"*, we obtain (5) as well.

$$pr(X|Y)pr(Y) = pr(X\,and\,Y) \tag{4}$$

$$pr(Y|X)pr(X) = pr(Y\,and\,X) = pr(X\,and\,Y) \tag{5}$$

And since the left-hand side of both (4) and (5) are equal to the same thing (that is, to $pr(X\,and\,Y)$), they are equal to each other:

$$pr(X|Y)pr(Y) = pr(Y|X)pr(X) \tag{6}$$

And then we have Bayes's rule, as in (7) .

[6]Much of what I say here does not depend on that particular statement; one could adopt most of what we discuss and still believe that the heart of science is prediction, but I will not delve into this question.

[7]This section should be shorter, and the bulk of it should appear in Chapter One.

$$pr(X|Y) = \frac{pr(Y|X)pr(X)}{pr(Y)} \tag{7}$$

Now, this rule is used in a very surprising way within what is known as Bayesian analysis; we will take "$X$" to be a hypothesis H, and "$Y$" to be the set of observed data $D$. To make this more perspicuous, I will rewrite this and change the names of the variables:

$$pr(H|D) = \frac{pr(D|H)pr(H)}{pr(D)} \tag{8}$$

Now this says something much more remarkable from a scientist's point of view. Translating it into English, it says that the probability of a hypothesis, given what we have observed (and what else do we have other than what we have observed?) is equal to the product of two numbers, divided by a third number. It is the product of the probability that the hypothesis assigned to the data and the probability of the hypothesis in the abstract, divided by the probability of the observations themselves.

Suppose that's all true, and suppose that we can somehow come up with those values. It would then follow that we could choose our hypothesis out of a range of different hypotheses $H$ by finding the one whose probability was greatest, given the observations. That's the heart of the notion of a Bayesian analysis.

Of the three values that I described, only one is difficult to obtain, and that is the probability of the data, the denominator of (8). But we do not worry about that, because it does not really matter. Since what we care about is choosing which hypothesis is the best, given the data, we are just going to keep $pr(D)$ fixed as we consider various different hypotheses. So the hypothesis $h$ for which the value $pr(D|h)pr(h)$ is the greatest is the same as the hypothesis for which the value of $\frac{pr(D|h)pr(h))}{pr(D)}$ is the greatest, and that is the hypothesis we want. More mathematically, we say we want to identify the $H$ as follows:

$$H = \underset{h}{\text{argmax}}\, pr(D|h)pr(h) \tag{9}$$

This identifies $H$ as being the hypothesis for which the product of the two probabilities defined there is the greatest. We still need to obtain two values: the probability of the data, given any of the hypotheses we are considering, and the probability of each of those hypotheses. We obtain the first by demanding that we only consider probabilistic grammars, which we introduced (following Solomonoff) in the previous section, and we obtain the second by establishing a prior probability over grammars. That is worth emphasizing: the $H$ that we seek here is a generative grammar that assigns probabilities to its output. We will seek a way to distribute the probability mass over all grammars based just on what they look like as grammars, independent of how they treat any actual data. If we can do that, then the task of choosing a grammar, given a set of data, will be a matter of jointly considering two equally important things about the grammar: how good a job does it do of modeling the data, and how good is it as a grammar?

To summarize so far: in section 2.1, we explained that to analyze data from a particular language, we need to establish two probability distributions, one which is essentially the grammar of that language, and the other which is a hypothesis regarding Universal Grammar. In section 2.3, we saw how choosing a grammar can be understood as an optimization process: pick the grammar that maximizes the expression on the right in equation (9), which includes two parts: the probability assigned to the data by the grammar, and the probability of the grammar. We need to explore more deeply the question of what it means to assign a probability to a grammar: this is the role of $\pi_G$, the universal grammar that we mentioned briefly in connection with the express in (1), what a Bayesian would refer to as our *prior* (i.e., prior probability distribution) over a class of grammars. We turn in the next section to the question of how reasonable this prior distribution would look to a linguist if it were very, very austere.

## 2.4 Establishing a prior probability for grammars

I am going to assume henceforth that the class of possible grammars is infinite. I don't think that there is a serious alternative to this hypothesis. Occasionally the suggestion is made that the real heart of a grammar of a human language is the correct selection of values assigned to a finite set of parameters (where each parameter can in principle only take on a finite number of values). But even if one believes in such a limitation (and as it happens, I do not), the "real heart" is only the heart: there's the rest of the grammar, which includes at the very least a lexicon, and I daresay no linguist would dream of saying that there is an upper bound on the size of a lexicon. The bigger the lexicon, the less likely it is, and its probability (to say nothing of its plausibility) shrinks very rapidly as its size increases.

Most theories of grammar are "non-parametric," in the specific sense now that grammars typically consist of formal (indeed, algebraic) objects which can be made larger and larger, by adding more to them (even if the "more" is just another lexical item, or construction, phrase-structure rule, condition on a phrase-structure rule, etc.) What

we do know about them, though, is that they are built up out of a specific set of formal objects, or symbols. There is no limit to the number of grammars, because there is no limit to the number of symbols (that is, number of occurrences of symbols) that may appear in a grammar.[8]

I would like now to be able to talk about the size or length of a grammar. We are accustomed to using all sorts of different symbols in our formalism, but as we pointed out in Chapter 2, we can make life a lot easier by agreeing to view each symbol as a shorthand for a string of 0s and 1s (which is what real life computers think, anyway). It then follows that for any given length $L$, there are exactly $2^L$ different string of symbols that could in principle be grammars. (Most of the strings will be formally meaningless in all likelihood, but that's OK, because we're trying to get an upper limit on things). For technical reasons that I will not go into,[9] we will assume that it is always possible to tell, from a purely formal point of view, when we have gotten to the end of the grammar (perhaps by setting up a symbol to specifically mark for that, or in any of a variety of ways).

We know one more thing about grammars that we want to use, and that is that a shorter grammar is always better than a longer grammar, all other things being equal. The reader may object to that, and say, "we've been there before, and done that, and don't want to do it again: sometimes the notation is doctored so that a shorter grammar is not the psychologically real one." To which I would reply two things: first, when we say "all other things being equal," we really and truly mean that we are making the claim that shorter is better only when we agree to fix and hold constant the theory of grammar; and second, we are not quite saying that *better = psychologically correct*. What we're saying is that if we are to assign a probability mass over an infinite class of grammars, then it must be the case that as we look at the class of longer and longer grammars (and they are vastly more numerous than shorter grammars, since for any length $L$ there are $S^L$ of them, and that expression grows quickly with $L$), the total probability mass assigned to them gets indefinitely small. For any amount of probability mass $\epsilon$ you choose, no matter how small, there is a length $\hat{L}$ such that the sum of the probabilities of all of the infinite number of grammars that are of length $\hat{L}$ (or greater) is less than $\epsilon$.[10]

There is one more crucial step to take, and that is one that permits us to escape from the clause that says, "given a fixed theory of grammar." Because we are not "given" a theory of grammar, after all; each of us is free to develop our own theory of grammar, and how can simplicity in my theory be compared with simplicity in your theory? What if my theory has (let's say) grammatical relations as a primitive notion, and yours doesn't? My theory allows me to write some grammars very simply that yours either can't express, or can only express with great complexity.

The answer I would like to suggest is based on algorithmic complexity (and thus is an application of ideas by Solomonoff, Chaitin, Kolmogorov, Rissanen, and, a little less directly, Turing; see Li and Vitnyi 1997 [**?**] for details). The basic idea is this: any computation can be specified as a particular Turing machine, and there is, furthermore, such a thing as a *universal Turing machine*, and the latter is so important that we will give that phrase a three-letter abbrevation: *UTM*. Such a machine (and there are many of them) can be programmed to function like *any* other Turing machine, and in particular to accept programs in a higher level language, such as C, Lisp, or natural-language-grammar-language. If there were only one such machine, we could use the length of the program in its language as the basis for our notion of complexity, but the fact is that there are many, different *UTM*s, so our problem is how to deal with the nature of the differences among *UTM*s.

The reader has undoubtedly encountered the notion of a Turing machine: it is a finite-state device which is connected to an infinite tape, a tape which in turn is broken up into boxes in which only $x$'s and blanks appear. The input to the machine is written by us in the first instance, and the machine can rewrite what it sees on the tape according to its internal program. Anyone who has actually looked at instructions to a Turing machine will be struck by how elementary the statements look: e.g., "If there is an $x$ in the box you see now, erase the $x$ and move one box to the right."

But that's just typical of what instructions look like, even in real-world computers, at the level of machine language code. With real machines and also with Turing machines, one can enter a program written in a higher order language (like *C* or natural language grammar). In the case of a Turing machine, one does this by writing down two long things on the tape before beginning: the first is a compiler for the higher language (it is, so to speak, a program written in the *UTM*'s native language which will input what follows it on the tape, view it as a program and translate it into the *UTM*'s native language), and the second is the program in the higher order language. If the Turing machine is truly a universal Turing machine, then it can be made to imitate any *other* Turing machine: that is, it's always possible to write a program which, if it is used to precede any chunk of data on the tape, will cause the universal Turing machine to treat that data like the Turing machine you wish it to imitate. (To put the same point slightly differently, there is a rough and ready equivalence between Turing machines and higher-level programming languages).

At the risk of becoming too repetitive, I will underscore the point that we insist on bringing the description

[8] The boundary between parametric and non-parametric analyses is getting a bit harder to draw these days. Goldwater's 2006 [**?**] employment of the Chinese Restaurant process blurs the line further, allowing most of lexicon generation to be viewed with a parametric model.

[9] This relates to the notion that our notation has the prefix condition, which relates in turn to satisfying the Kraft inequality.

[10] This is still not clear enough.

down to the level of a Turing machine, *not* because we plan to do any serious linguistic work writing our grammars in machine level code—because we will not—but as a way of ensuring that we all play on a level playing field, to the degree that we possibly can.

Given a particular universal Turing machine $UTM_1$, our job is to write a compiler which allows us to write natural language grammars. A compiler in the real world is the name we give to a computer program that takes a relatively compact program (usually one that has been written by a human being) and automatically converts it into the language of 0s and 1s, the machine language used by a particular computer. In some ways, a compiler is a decompresser: it takes as its input a relatively short string of symbols, and creates a longer string that contains all of the detailed instructions the computer needs to carry out the intent of the original program.

Linguists write grammars, which are compact descriptions of operative generalizations in natural languages, and these descriptions allow one, in principle, to analyze sentences of a given language. And *to analyze sentences of a given language* means to give an analysis of the a particular sentence, or a long sequence of sentences. So here's where we are: the linguist who is using $UTM_1$ (which is a machine) feeds it first a grammar-compiler, and then a grammar of a particular language, and then one or more sentences to be analyzed. Out from $UTM_1$ comes an analysis of the sentences.

Let's look more closely at this grammar-compiler, which we will refer to as $UG(UTM_1)$: it is a Universal Grammar for $UTM_1$, and for any particular UTM, there can be many such. Each grammar-compiler constitutes a set of recommendations for best practices for writing grammars of natural languages: in short, a linguistic theory. In particular, we define a given UG by an interface, in the following sense—we need to do this in order to be able to speak naturally about one and the same UG being run on different $UTM$s (a point we will need to talk about in the next section). A UG specifies how grammars should be written, and it specifies exactly what it costs to write out any particular thing a grammarian might want to put into a grammar. Naturally, for a given $UTM$, there may be a large number of ways of implementing this, but we care only about the simplest one, and we will henceforth take it for granted that we can hire someone and outsource the problem of finding the implementation of a particular UG on any particular $UTM$.

Once we have such a grammar, we can make a long tape, consisting first of $UG(UTM_1)$, followed by a Grammar for English (or whatever language we're analyzing), as we have already noted—plus a compressed form of the data, which is a sequence of 0s and 1s which allows the grammar to perfectly reconstruct the original data. It is a basic fact about information theory that if one has a probabilistic grammar, then the number of bits (0s and 1s) that it takes to perfectly reproduce the original data is exactly $log_2\, pr(data)$. We use that fact here: and we set things up so that the third section of the information passed to the UTM is a sequence of 0s and 1s that perfectly describes the original data, given the Universal Grammar and the grammar of the language in question.

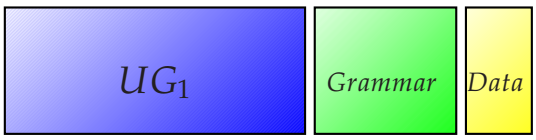



Figure 1: Input to Turing machine

As we have already mentioned, there will be many different ways of accomplishing this. Each $UTM$ is consistent with an indefinitely large number of such universal grammars, so notationally we'll have to index them; we'll refer to different Universal Grammars for a given $UTM$ (let's say it is still $UTM_1$) as $UG_1(UTM_1)$ and $UG_2(UTM_1)$, etc. This is no different from the situation we live in currently: there are different theories of grammar, and each one can be thought of as a compiler for compiling a grammar into a machine-language program that can run on a $UTM$. A UG is intended to be used to write grammars for all languages of the world. At any given time (which is to say, at any given state of our collective knowledge of languages of the world), for any given $UTM$, there will be a best UG; it is the one for which the sum of the length of UG, plus the sum of the lengths of each grammar written in UG, plus the compressed length of the data for each language in its corresponding grammar is the shortest.

We are almost finished with the hard part. We now can assign a probability to a grammar that has been proposed. Given a universal Turing machine $UTM_1$, a univeral grammar $UG_1$ written for it, and a grammar $g$ written for universal grammar $UG_1$, the probability assigned to it is

$$pr(g|UG_1) = \frac{1}{2^{length(UG_1)}2^{Length(g)}} \tag{10}$$

In effect, this is the simplest way to divide the probability mass up over the entire universe of possible universal

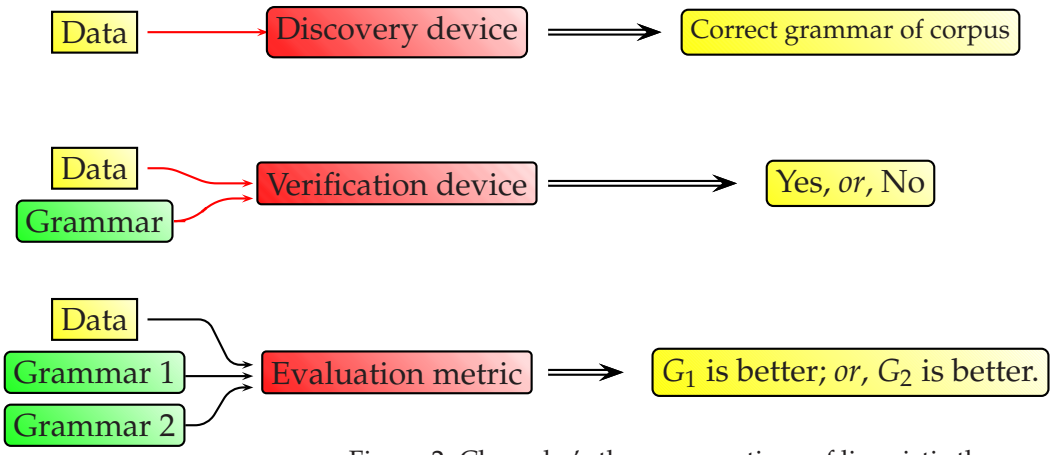



Figure 2: Chomsky's three conceptions of linguistic theory

grammars and language-particular grammars, and it extracts the probability that goes to just this one (=this universal grammar and this grammar).[11] There is a lot of serious work that has gone into this equation, and I have only skimmed over the surface here—but bear in mind that this is directly responding to the needs of equation (9) above. We now have a prior probability distribution over grammars, which is what we needed by the end of the previous section in order to develop a Bayesian approach to linguistics and grammar justification. What we have not addressed yet is the question of uniqueness. Since there are many universal Turing machines, we could expect that the distribution over $(UTM_i + grammar_j)$ will vary with the choice of universal Turing machine.

This is an interesting question, to which I will give a sketch of an answer shortly; but before I do, let's look at a bit of recent linguistic history from the Bayesian perspective that we have just sketched.

## 3 Classical generative grammar

Classical generative grammar is the model proposed by Noam Chomsky in his *Logical Structure of Linguistic Theory* [**?**], and sketched in *Syntactic Structures* [**?**] (and assumed in *Aspects of the Theory of Syntax*, 1965). In a famous passage, Chomsky suggested comparing three models of what linguistic theory might be which are successively weaker, in the sense that each successive model does less than the preceding one. In the first model, linguists would develop a formal device that would produce a grammar, given a natural language corpus. In the second model, the formal device would not generate the grammar, but it would check to insure that in some fashion or other the grammar was (or could be) properly and appropriately deduced or induced from the data. In the third model, linguists would develop a formal model that neither produced nor verified grammars, given data, but rather, the device would take a set of observations, and a set of two (or more) grammars, and determine which one was the more (or most) appropriate for the corpus. Chomsky suggests that the third, the weakest, is good enough, and he expresses doubt that either of the first two are feasible in practice.

Chomsky believed that we could and should account for grammar selection on the basis of the formal simplicity of the grammar, and that the specifics of how that simplicity should be defined was a matter to be decided by studying actual languages in detail. In the last stage of classical generative grammar, Chomsky went so far as to propose that the specifics of how grammar complexity should be defined is part of our genetic endowment.

His argument against Views #1 and #2 was weak, so weak as to perhaps not merit being called an argument: what he wrote was that he thought that neither could be successfully accomplished, based in part on the fact that he had tried for several years, and in addition he felt hemmed in by the kind of grammatical theory that appeared to be necessary to give such perspectives a try. But regardless of whether it was a strong argument, it *was* convincing.[12]

[11] Again, for simplicity's sake, I am assuming that our $UTM$'s can be extended to allow input strings which contain all the symbols of an alphabet such as $\mathcal{A}$.

[12] Chomsky's view of scientific knowledge was deeply influenced by Nelson Goodman's view, a view that was rooted in a long braid of thought about the nature of science that has deep roots; without going back too far, we can trace the roots back to Ernst Mach, who emphasized the role of simplicity of data description in the role played by science, and to the Vienna Circle, which began as a group of scholars interested in developing Mach's perspectives on knowledge and science. And all of these scholars viewed themselves, quite correctly, as trying to cope with the problem of induction as it was identified by David Hume in the late 18th century: how can anyone be sure of a generalization (especially one with infinite consequences), given only a finite number of observations?

While it's not our business here today to go through the work of this tradition in any detail, it is nonetheless useful to understand that the problem can be sliced into **two parts**, the **pre-symbolic** and the **symbolic**. The pre-symbolic problem is: how do we know the right way to go from observations that we make to statements which represent or encode the observations? Who determines what that process is, and how can

Chomsky proposed the following methodology, in three steps.

First, linguists should develop formal grammars for individual languages, and treat them as scientific theories, whose predictions could be tested against native speaker intuitions among other things. Eventually, in a fashion parallel to the way in which a theory of physics or chemistry is tested and improved, a consensus will develop as to the form and shape of the *right* grammar, for a certain number of human languages.[13]

But at the same time, linguists will be formulating their grammars with an eye to what aspects of their fully specified grammars are universal and what aspects are language-particular. And here is where the special insight of generative grammar came in: Chomsky proposed that it should be possible to specify a higher-level language in which grammars are written which would have the special property that the *right* grammar was also the *shortest* grammar that was compatible with any reasonable sized sample of data from any natural language. If we could do that, Chomsky would say that we had achieved our final goal, *explanatory adequacy*.

There are two crucial aspects of this picture that we must underscore: the first involves the limited role that data plays in the enterprise, and the second involves the open-endedness of the search for the correct method for computing grammar length. In this view, the search for universal grammar consists of a first chapter, in which data is used in the usual scientific way (whatever that means!) to select between proposed grammars for individual languages, and when we have enough such grammars, we use our knowledge of correct grammars to develop a formalism which allows for a simple formulation of those grammars. Now, with this formalism in hand, we can (we hope) make sense of the notion that linguistic theory picks the simplest grammar that is consistent with the data, because "simple" will now have meaning in terms of the formalism that was discovered in a scientific, empirical fashion. In this second chapter of linguistic theory, we compute the complexity of a grammar by computing its length, and we compute the complexity of a set of grammars for several different languages by summing up the lengths of each of the languages.

There are two assumptions in this picture that are wrong, I will suggest. The first is that it is possible to have a second phase of linguistic theory in which we care only about the complexity of the grammar, and not the grammar's tight mesh with the data, and the second is that we can ignore the complexity of the system that that does the actual computation of the complexity of each grammar (i.e., the grammar's length). The first assumption concerns THE SMALL ROLE OF DATA FALLACY, and the second is the UNIVERSAL GRAMMAR IS FREE GRAMMAR FALLACY.

In Figure 3, we see a schematic of the problem that the classical generativist thought he was providing a solution to. You and I have different views about grammar and how individual grammars should be expressed. For purposes of exposition and discussion, we agree to formalize our theoretical intuitions in a fashion that permits us to evaluate grammar complexity in purely quantitative terms, but as long as we don't *agree* on a common way to do that upfront, we will typically get stuck in the sticky situation illustrated in that figure: I think my set of grammars is better (because shorter) than yours, and you think your set of grammars is better (because shorter) than mine. We do not have a common and joint language by which to settle our disagreement. We calculate the width of the green rectangles for you and for me; whoever has the smaller total length wins. The data lurking on the right hand side do not enter into the calculation except in some implicit way: in some fashion, our grammars must each generate the data for English and Swahili, and not generate too much that isn't grammatical as well.

The classical model of generative grammar made clear that this general perspective only makes sense if we explicitly agree on using the same Universal Grammar. If you and I use different universal grammars, then it is perfectly possible that we will arrive at a situation as in Figure 3, where you and I have different opinions as to who won: I think I won, because the total length of my grammars for English and Swahili is shorter than the lengths of yours, but unfortunately you think you won, because as you calculate the lengths of the grammar, the sum total of the lengths of your grammars is shorter than that of mine. There is no guarantee that we will agree on computing the complexity or length of a grammar: that is the problem. And the generative solution was to say that we must—somehow—come to an understanding of the *right* UG, and then this problem will never turn up, since everyone will use the same measuring tools.

---

we even try to make explicit what the right connections are? Fortunately for us, I have nothing to say about this extremely difficult problem, and will leave it utterly in peace.

The **second**, symbolic problem matters to us, though. The second problem is based on the notion that even if it's possible to make fully explicit the ways in which we must translate statements from one system of symbolic representation to another, the two systems may disagree with respect to what is an appropriate generalization to draw from the same set of observations.

This was the problem that Chomsky proposed to solve, and he proposed to solve it by removing it from philosophy or epistemology, and moving it into science (also referred to as *naturalizing* it): the choice of the language in which generalizations are expressed cannot be decided on apriori principles, he suggested, and the methods of normal science should be enough to settle any question that might arise.

[13] I am not going to criticize this suggestion in this paper, but I think that while this methodological principle sounds very good, it has proven itself to be unfeasible in the real world. That is, this principle amounts to the belief that there are methods of proving that grammar Γ does, or does not, represent the psychologically real grammar of English, or some other language, based no more on considerations of simplicity than any other laboratory science would use, and based on the same kinds of laboratory-based methods that any other science would use. This method has to be very reliable, because it serves as the epistemological bedrock on which the over-arching Universal Grammar will be based. This method, however, does not exist at the current time. I am not saying it *could* not exist; anyone is free to believe that, or not. But in 50 years of syntactic research, we have not developed a general set of tools for establishing whether a grammar has that property or not. Without a functioning version of this test, however, classical generative grammar cannot get off the ground.

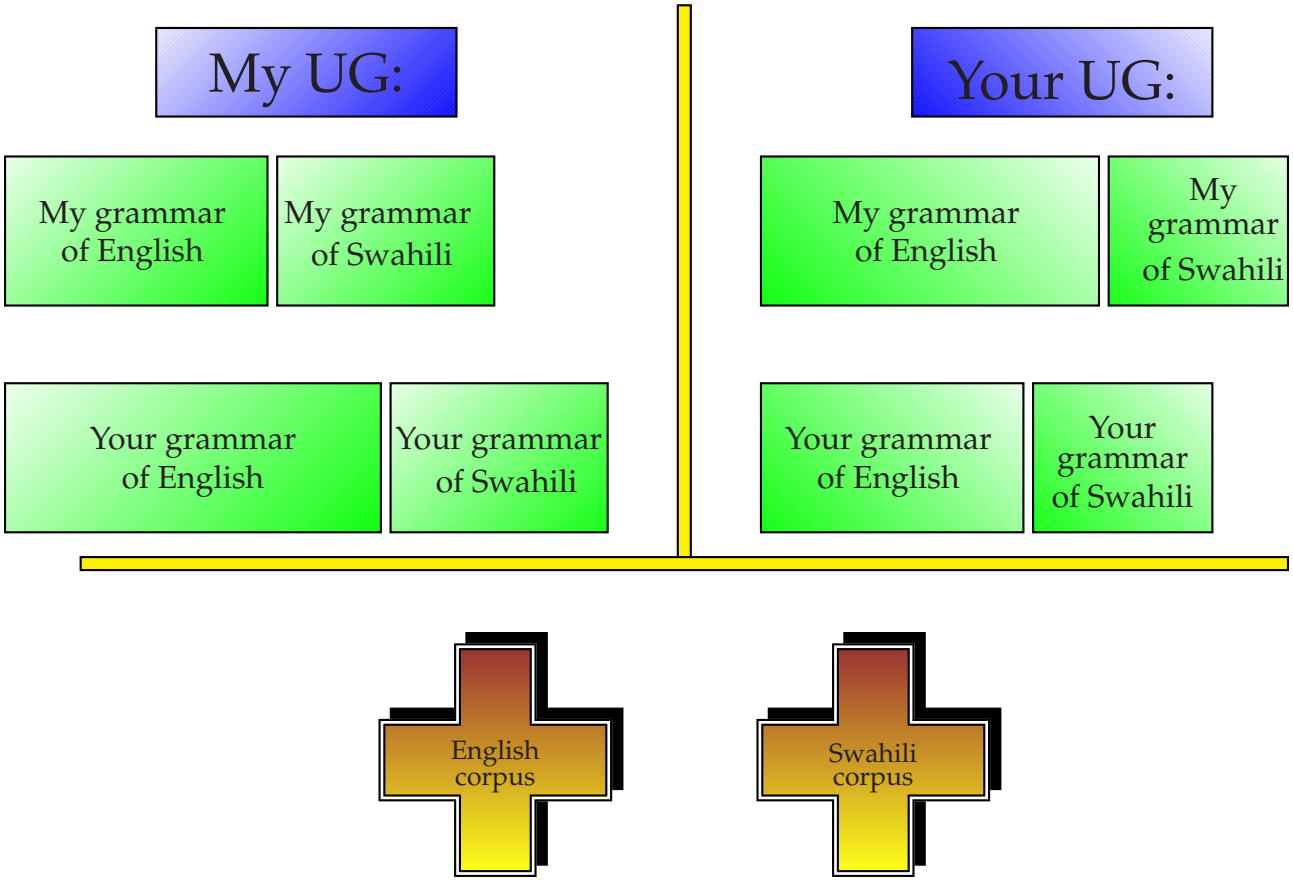



Figure 3: The pre-classical generative problem

### 3.1 The flaws

There are two fatal flaws to this program, though, I regret to say: I regret it because it sounds marvelous, and I would have been happy to devote many years to its development and execution. But there are two fatal flaws nonetheless, as we have noted. The first is the SMALL ROLE OF DATA fallacy, and there were people, like Ray Solomonoff, who were working to demonstrate this at exactly the same moment in the 1950s.

The second fatal flaw is a bit more complex, but it can be summarized as this: Chomsky's actual method of theory development put a strong emphasis on developing the grammar-writing linguistic theory (which he eventually called Universal Grammar) right from the start, and this led to a situation in which it seemed like the linguist needed to pay a certain "cost" for adding complexity to the grammars of individual languages, but there was no cost for adding complexity to the Univeral Grammar. This is a subtle point, but it is absolutely vital. I will call this second flaw the UNIVERSAL GRAMMAR IS FREE GRAMMAR fallacy. We will discuss these two points in the next section.

But the good side of this is that both flaws can be described by saying that we have left terms out of the expression that we need to minimize: there needs to be one term involving the Universal Grammar, and one involving the data—all of the data from all of the languages; and once we have done that, we will have made a good deal of progress.

## 4 More Bayesian discussion

In this section, we will solve the two problems of classical generative grammar with the two expressions that we discussed in section 2, which were *the probability of the data* and *the length of the grammar*. The probability of the data is the key to the solution of the problem of confirmation. A probability is always an arithmetic value between 0 and 1, but rather than work with that number (which tends to be a very, very small positive number), we use -1 times the logarithm of the probability, which will be a reasonably large positive number, and we will refer to that as the *positive log* (or *plog*) of the probability.

### 4.1 The logic of confirmation

Ray Solomonoff was in Cambridge, Massachusetts, working on the problem of induction of generalizations on the basis of finite data, at the same time that Chomsky was working on the problem of justifying grammars—that is, during the mid 1950s. Solomonoff had been an undergraduate at the University of Chicago, where he had studied with Rudolf Carnap, the most influential of the members of the Vienna Circle who had come to the United States. During this period (1930s through 1950s), there was a lot of discussion of the nature of the logic of confirmation. There were many puzzles in this area, of which perhaps the most famous was the why it was or wasn't the case that the observation of a white iPod confirmed the statement that all ravens are black. Since "All ravens are black" certainly appears to be equivalent to "anything which is not black is not a raven," and my white iPod seems to confirm that, why doesn't observation of my white iPod confirm an enormous number of irrelevant generalizations?

Without pretending to have shed any light on the question, it would not be unfair to say that this example brings home the idea that there is a difference between "real" confirmation of a generalization by a particular observation, on the one hand, and "simple" consistency of an observation with a generalization. I recently noticed a comment made by Chomsky in *Language and Mind*, **[?]** that made it clear that he was well aware of this issue, or so it seems to me. On pp. 76-77, he observes:

> A third task is that of determining just what it means for a hypothesis about the generative grammar of a language to be "consistent" with the data of sense. Notice that it is a great oversimplification to suppose thata child must discover a generative grammar that accounts for all the linguistic data that has been presented to him and that "projects" such data to an infinite range of potential sound-meaning relations....The third subtask, then, is to study what we might think of as the problem of "confirmation"—in this context, the problem of what relation must hold between a potential grammar and a set of data for this grammar to be confirmed as the actual theory of the language in question.

If a grammar generates too many sentences—if it is too permissive about what it allows, so to speak—and we do not add enough negative grammaticality judgments as part of our data that we must account for, then we will be wrongly biased by our method to select an overly simple grammar—that is the issue that Chomsky was referring to.

The bottom line regarding probabilistic models is that they provide an answer to the question of how different grammars that generate the same language may be confirmed to different extents by the same set of observations. Each grammar assigns a finite amount of probability mass (in fact, it assigns exactly 1.0 total units of probability mass) over the infinite set of predicted sentences, and each grammar will do that in a different way. We *test* the

grammars by seeing what probability they assign to actual observed data, data that has been assembled in some fashion which is not biased with respect to intentionally producing data that pushes in favor of one person's theory or another. We use the differing probabilities assigned by the different grammars to rank the grammars: all other things being equal, we prefer the grammar that assigns the most probability mass to data that had already independently been observed.

This is a radical shift from the Chomsky-Putnam assumption that the only reasonable way to link a grammar to the empirical ground is by seeing how well the boundary between grammatical and ungrammatical maps to the boundary between acceptable and unaccpetable. It says: If you have some good data from a language, then compute the probability assigned to that data by each of the candidate grammars, and as a first approximation, you should choose the grammar that assigns the highest probability to the data. Choosing the one that assigns the largest probability is mathematically the same as selecting the one with the smallest plog probability (i.e., the one for which $log\frac{1}{pr(data)}$ is a minimum), and that is how we shall state this criterion: all other things being equal, choose the grammar which assigns the smallest plog probability to the data.

This gives us an initial hold on the solution to the first problem, that of the small role of data, or equivalently, the problem of establishing an explicit measure of degree of confirmation of grammar by data. Given a corpus of data $d$, For each grammar $g$, we must calculate both the length of the grammar, and the plog of the data that $g$ assigns to the data, and we select the grammar for which the sum of those two quantities is a minimum.

The reader may quite reasonably be wondering by what rights we simply add together two quantities, one of which is the length of a grammar and the other of which is the logarithm of the reciprocal of a probability. What do these two have in common that would make it a meaningful act to add them? We will see that they are both measured in the same units, the Shannon bit, but we are not there yet.

On this account, Figure 4 illustrates how we select our grammar for English and Arabic: for each, we have a corpus of data, and we consider various grammars for the data; we select the grammar which jointly minimizes the sum of the grammar length and the "data length," where the "data length" is the plog of the data assigned by that particular grammar. In that figure, analysis 2 will be the right one for both languages.

To make a long story short, and shamelessly blending together Solomonoff's work [**?**] with that other people's (notably Kolmogorov, Chaitin, and Rissanen [**?**]; see [**?**], we can conclude the following: we can naturally assign a probability distribution over grammars, based on their code length in some appropriate universal algorithmic language, such as that of a Universal Turing machine; if such grammars are expressed in a binary encoding, and if they are "self-terminating" (i.e., have the prefix property: there is no grammar $g$ in the set of grammars which is a prefix to some longer grammar $g'$), then we assign each grammar $g$ the probability $2^{-|g|}$.

We can draw this conclusion, then:

If we accept the following assumptions:

- Our grammars are probablistic (they assign a distribution to the representations they generate);
- The goal, or one major goal, of linguistics is to produce gramamrs;
- There is a natural prior distribution over algorithms (though we will have to speak to some concerns about choice of universal Turing machine or its equivalent;

then we can conclude:

- there is a natural formulation of the question, what is the best grammar, given the data D? The answer is:

$$\arg\max_{g} pr(g)pr(D|g)$$

where

$$pr(g) := 2^{-|g|}$$

We can now describe the work of the linguist in an abstract and idealized fashion as follows. She has a sample of data from a set of languages, $\mathcal{L}$. She has a computer and a computer language in which she develops the best grammar of each language individually. If she has only one language (English, say) in her corpus, then she will look for the grammar which maximizes the probability of the data by minimizing the description length of the data, which is to say, minimizing the sum of the length of the grammar plus the inverse log probability of the data, given the grammar. She will have no motivation for conceptually dividing her grammar of English into a part that is universal and a part that is English-particular.

But suppose she (the linguist) is studying two languages, English and Arabic; English and Arabic have different structures, and probability must be assigned according to different models. Some parts of the model will be specifically set aside for treating sentences from English, some for treating sentences from Arabic, and other parts will be relevant for both.

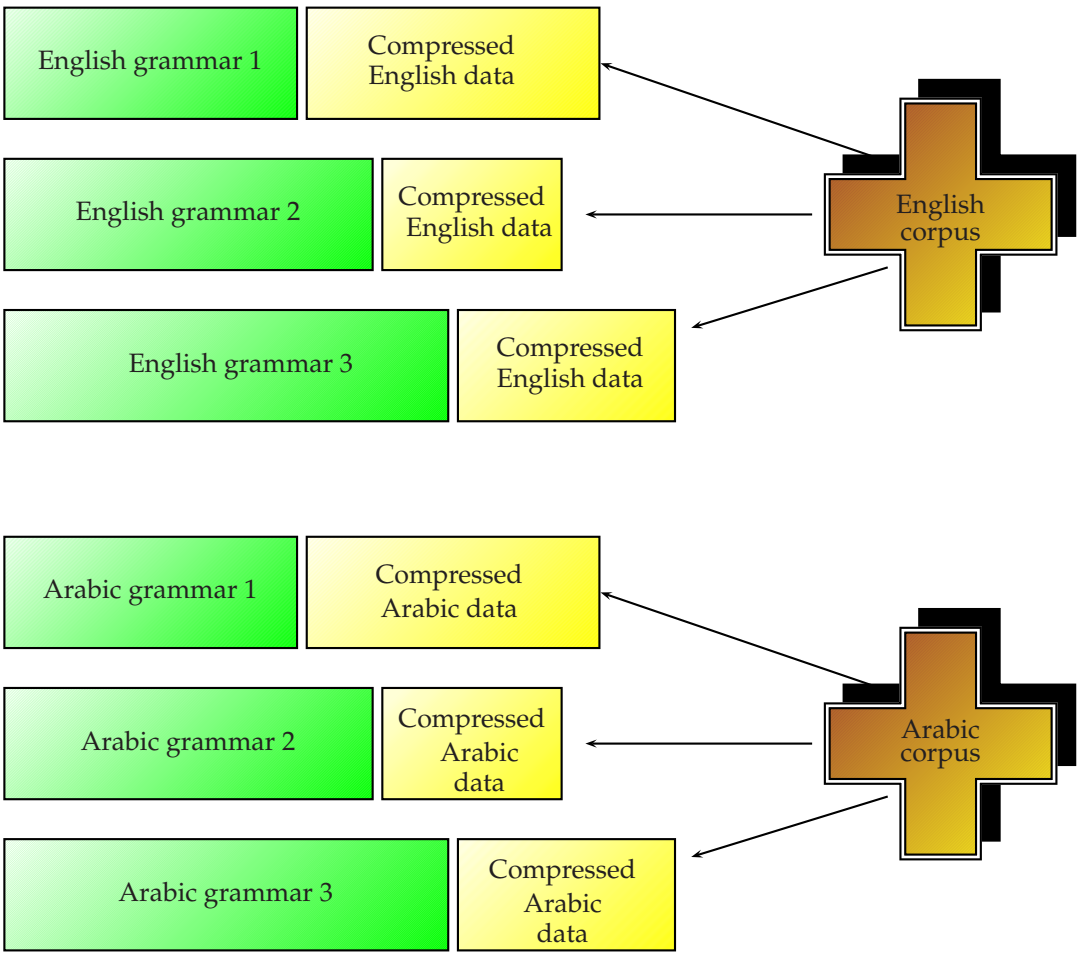



Figure 4: Generative model with a data term

In order to write a compact grammar, even for a single grammar, it is always a winning strategy to build a compact notational system for the various complexities of the language: those that involve phonology, morphology, syntax, and so on. To put it another way, the programming language that is handed to the linguist for writing grammars may not be at all to her liking: it may be highly redundant and difficult to use, so she has a strong interest in developing a simple-to-use grammar-writing language, which can be compiled into the language of the machine in question. This process is essentially that of developing a formal linguistic theory. To put it another way: it is a reasonable goal for linguists to develop a Linguistic Theory which is a specific explicit way of writing grammars of any and all natural, human languages. This Linguistic Theory is in effect a higher-level computer language which, when given a complete grammar, can perform tasks that require knowledge of language, and only knowledge of language (like parsing, perhaps).

## 4.2 TANSTAAFUG: There ain't no such thing as a free UG

We noted above that there are two fatal flaws to the classical generative picture: the first (which we discussed in the preceding section) was its failure to deal with the relationship between grammar and data (the "*small role of data" flaw)*, while the second is the assumption that the complexity of Universal Grammar is cost-free for the scientist, the *Universal Grammar is Free Grammar* fallacy. Classical generative grammar operated as if there were a valid principle in effect that the less information the linguist included in his language-particular grammar, the better things were. This could be accomplished with or without increasing the complexity of UG, both in theory and in reality. When a grammatical proposal does not increase the complexity of UG but does simplify the grammatical description of a language, then everyone agrees that the change constitutes an improvement. But quite often, a proposal is made to simplify the description of a particular language (or two or five) by removing something that all these grammars shared, and placing them not in the particular grammars, but in the Universal Grammar common to all the grammars.

The *Universal Grammar is Free Grammar* fallacy is the following assumption: while the complexity of a particular grammar counts against it as a scientific hypothesis, and is an indirect claim about the information that must be abstracted from the "training data" by the language learner, the complexity of Universal Grammar has no cost associated with it from a scientific point of view. Its complexity may be the result of millions of years of evolutionary pressure—or not; the linguist neither knows nor cares.

I call this a fallacy, because it inevitably leads to a bad and wrong result. There is one very good reason for insisting that the researcher must take into consideration the informational cost of whatever he postulates in his Universal Grammar. If he does *not* do so, then there is a strong motivation for moving a *lot* of the specific detail of individual grammars into Universal Grammar, going so far as to even include perfectly ridiculous kinds of information, like the dictionary of English. UG could contain a principle like, "if the definite article in a language $\mathcal{L}$ is *the*, then $\mathcal{L}$ is SVO and adjectives precede their head nouns." That would simplify the grammar of English, and if there is no cost to putting it in UG—if that does not decrease the prior probability of the entire analysis—then the rational linguist will indeed put that principle in UG.[14]

## 4.3 The bigger picture

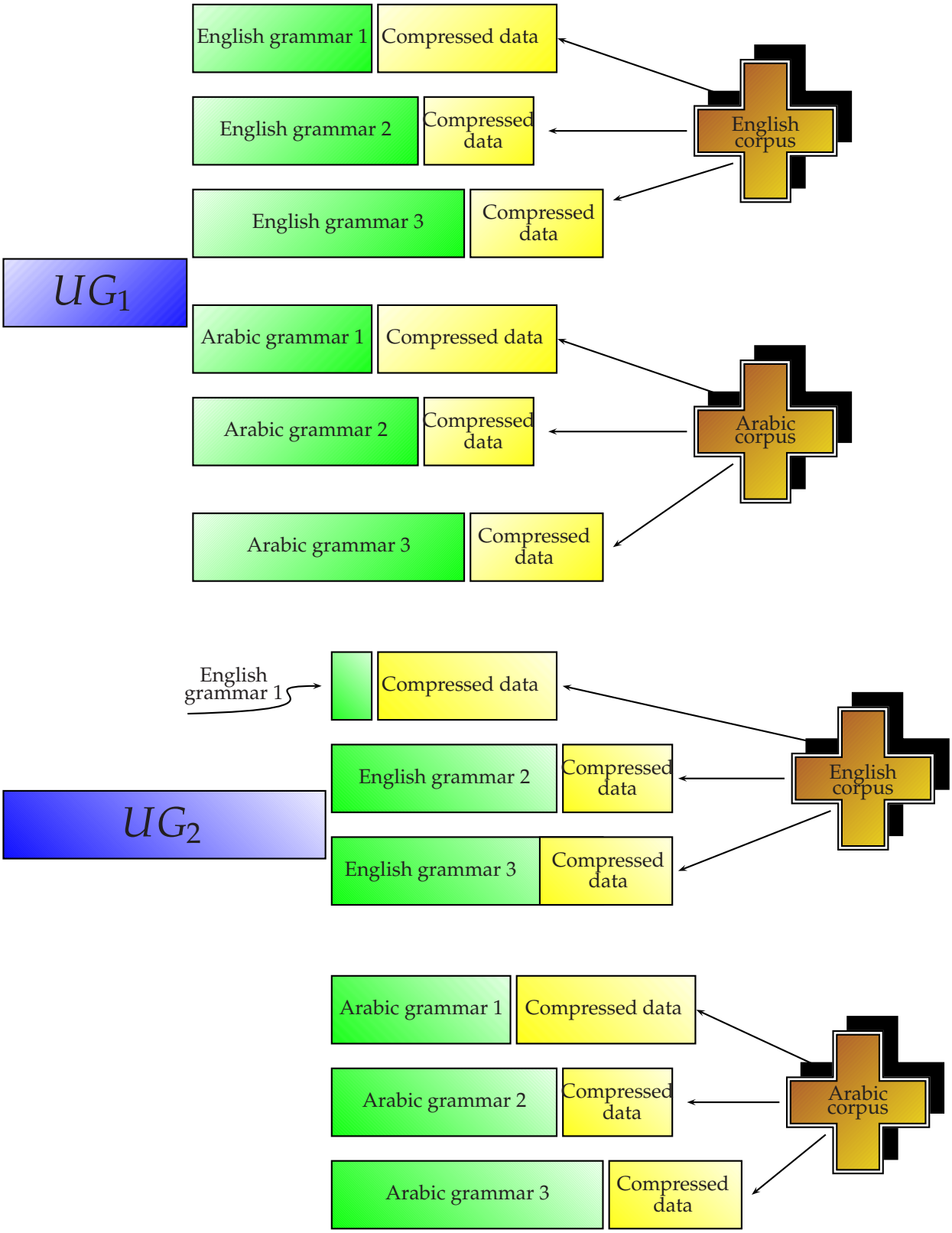
The picture of linguistics that we have arrived at is this: we are all seeking a Universal Grammar, $UG_1$, which will run on an all-purpose computer, and which will accept grammars and corpora, and the output of this large

[14]Now, there is a natural way to understand this which is valid and convincing, in my opinion at least. We can consider the goal and responsibility of the linguist to account for all observed languages, and if we really can find some elements in all our grammars of languages and then place them in a common subroutine, so to speak, changing nothing else. then we will save on the overall length of our grammatical model of the universe: if we shift $N$ bits from individual grammars to universal grammar, and the cost of a pointer to this code is $q$ bits, and if there are $L$ languages in the current state of linguistic research that have been studied, then over-all we will save $(L-1)N - Lq$ bits. We save $LN$ bits by removing $N$ bits from each grammar, but this costs us $Lq$ bits where we leave a pointer, a placeholder for the function. Then we pay $N$ bits for the increase in size of Univeral Grammar.— Hence the prior probability of our account of the world's data will have just been multiplied by $2^{(L-1)N-Lq}$, which is no mean feat.

Let's push this a bit. What if out of our $L$ different languages, *half* of them need to refer to this function we have just shifted to Universal Grammar? Then each language must be specified for whether it "contains" that material, and that will cost 1 bit (more or less) for each language. Then moving this *out of* individual grammars and into UG will save us only $(L/2-1)N - L - \frac{Lq}{2}$ bits.

Generalizing this a bit, and assuming that $L_{yes}$ languages contained this material in their grammar, and $L_{no}$ did not (so that $L = L_{yes} + L_{no}$), then shifting material from individual grammars to UG will save us $(L_{yes}-1)N - L_{yes}log\frac{L}{L_{yes}} - L_{no}log\frac{L}{L_{no}} - qL_{yes}$ bits.

It may not appear it at first blush, but there is some real news here for someone defending this rationalist view of Universal Grammar. Such a person must find the least costly way to formulate any addition to Universal Grammar (that value is $N$), and in addition he must pay a price for every language in the world that does not need to refer to this aspect of Universal Grammar. Is that true? Or can the particular languages' grammar just pass over it in silence?


English grammar 1
Compressed data
English grammar 2
Compressed data
English grammar 3
Compressed data
English corpus
$UG_1$
Arabic grammar 1
Compressed data
Arabic grammar 2
Compressed data
Arabic grammar 3
Compressed data
Arabic corpus
English grammar 1
Compressed data
English grammar 2
Compressed data
English grammar 3
Compressed data
English corpus
$UG_2$
Arabic grammar 1
Compressed data
Arabic grammar 2
Compressed data
Arabic grammar 3
Compressed data
Arabic corpus

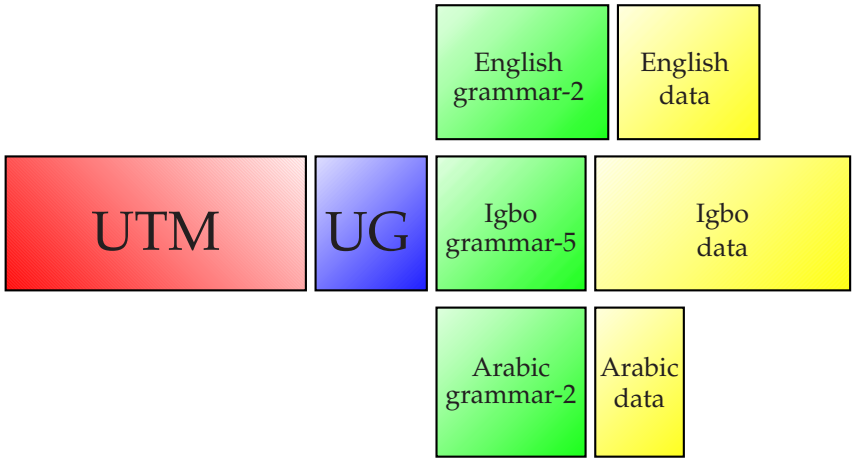



Figure 6: Total model

contraption is grammatical analyses of the corpora. With this $UG_1$, we can do a good job (though not a perfect job) of finding the best grammar for each language for which we have data; we do this by selecting grammars for each language, such that the overall sum, over all the languages, of the length of the grammar plus the compressed length of the data is maximally compact (i.e., minimized).

What could possibly go wrong in this beautiful picture? It would seem that this is the garden of paradise: if someone finds a way to improve the grammar of one language, while not making the grammar of the other languages worse, then we will be sure just by running the numbers, so to speak, on the computer. Different linguists may have different hunches, but we will be able to easily distinguish a hunch from an advance or breakthrough.

There is one thing wrong, however. We must worry about the possibility that there are two groups of linguists, using different underlying computers (different Universal Turing Machines, that is) who arrive at different conclusions. It is not enough to require that the linguist pay for the algorithmic cost of his Universal Grammar. That cost can always be shifted to the choice of universal Turing machine that is employed. And every linguist will be motivated to find or design the Universal Turing Machine that incorporates as much as possible of the linguist's Universal Grammar, making a mockery out of the requirement we have already discussed. What do we do now?

Readers who are familiar with foundational work in Minimum Description Length are likely to say now that there is no way out. We must (they will say) make an arbitrary choice regarding the UTM of choice, and then work from there. Yes, different choices of UTM will give somewhat different results, but in the end, there isn't a whole lot of difference between them.

This is not good enough an answer for us. We need a better answer, and an escape from this *conventionalist dilemma*.[15] We need to find a way—a good way, though it need not be a *perfect* way, since what in life is perfect?—by which to settle the question as to which UTM to use for our linguistic purposes.

# 5 The limits of conventionalism for UTMs

## 5.1 Join the club

We propose that the solution to the problem is to divide our effort up into *four* pieces: the selection of the best UTM, the selection of a universal grammar $UG^*$ among the candidate universal grammars proposed by linguists, the selection of the best grammar $g$ for each corpus, and the compressed length (the plog) of that corpus, given that grammar $g$: see Figure 6.

We assume that the linguists who are engaged in the task of discovering the best UG will make progress on that challenge by competing to find the best UG and by cooperating to find the best common UTM. In this section, we will describe a method by which they can cooperate to find a best common UTM, which will allow one of them (at any given moment) to unequivocally have the best UG, and hence the best grammar for each of the data sets from the different languages.

[15] The term refers to a difficult situation where we want to, but cannot, escape from a situation where each person involved ends up saying *You do things your way, and I'll do my things my way*, and there is no way to move past that.

The concern now, however, is this: we cannot use even an approximation of Kolmogorov complexity in order to help us choose the best UTM. We have to have already chosen a UTM in order to talk about Kolmogorov complexity. We need to find a different rational solution to the problem of selecting a UTM that we can all agree on.

We will now imagine an *almost* perfect scientific linguistic world in which there is a competition among a certain number of groups of researchers, each particular group defined by sharing a general formal linguistic theory. The purpose of the community is to play a game by which the best general formal linguistic theory can be encouraged and identified. Who the winner is will probably change over time as theories change and develop.

The annual winner of the competition will be the one whose total model length (given this year's UTM choice) is the smallest: the total model length is the size of the team's UG when coded for the year's UTM, plus the length of all of the grammars, plus the compressed length of all of the data, given those grammars. Of these terms, only the size of the UG will vary as we consider different UTMs. The winning overall team will have an influence, but only a minor influence, on the selection the year's winning UTM. We will return in just a moment to a method for selecting the year's winning UTM; first, we will spell out a bit more of the details of the competition.

Let us say that there are N members (that is, N member groups). To be a member of this club, you must subscribe to the following (and let's suppose you're in the group $i$):

1. You adopt an approved Universal Turing machine ($UTM^{\alpha}$). I will explain later how a person can propose a *new* Turing machine and get it approved. But at the beginning, let's just assume that there is a set of approved UTMs, and each group must adopt one. I will index different UTM's with superscript lower-case Greek letters, like $UTM^{\alpha}$: that is a particular approved universal Turing machine; the set of such machines that have already been approved is $\mathcal{U}$. You will probably not be allowed to keep your UTM for the final competition, but you might. You have a weak preference for your own UTM, but you recognize that your preference is likely not going to be adopted by the group. The group will jointly try to find the UTM which shows the least bias with respect to the submissions of all of the groups in a given year.

2. All the teams adopt a set of corpora which constitutes the data for various languages; everyone in the group must adopt all approved corpora. Any member can propose new corpora for new or old languages, and any members can challenge already proposed corpora. At any given moment, there is an approved and accepted set of data. The set of languages we consider is $\mathcal{L}$, and $l$ is a variable indexing over that set $\mathcal{L}$; the corpora form a set $\mathcal{C}$, and the corpus for language $l$ is $\mathcal{C}_l$.

3. The activities involved in this competition are the following. You will have access to the data $\mathcal{C}$; you will select a Universal Turing Machine of your choice; you will come up with a Universal Grammar UG, and a set of grammars $\{\Gamma_l\}$ for each language. You will calculate two quantities: (a) the length of the Universal Grammar $UG_1$, on your chosen $UTM^{\alpha}$ and (b) the length of the linguistic analyses, which we may call the *empirical term*, which is the lengths of all of the individual language grammars plus the compressed lengths of the corpora for all the languages. Symbolically, we can express the length of the linguistic analyses with an empirical term *Emp* for a given triple, consisting of: $UG_1$; a set of grammars $\{\Gamma_l\}$; and the length of the unexplained information in each corpus, as in (11):

$$Emp(UG_i, \{\Gamma_l\}, \mathcal{C}) = \sum_{l \in \mathcal{L}} |\Gamma_l|_{UG_i} - log\ pr_{\Gamma_l}(\mathcal{C}_l) \tag{11}$$

This group is trying to minimize the quantity:

$$|UG|_{UTM^{\alpha}} + Emp(UG_i, \{\Gamma_l\}, \mathcal{C}) \tag{12}$$

which is essentially the minimal description length of the data, given $UTM^{\alpha}$. Sometimes we will want to speak of different sets of grammars for our set of languages, because typically competing frameworks will compete, among other things, for the right grammatical description; when we do so, we will speak of two such sets of grammars for the same set of languages as $\{\Gamma_l\}^1$ and $\{\Gamma_l\}^2$.

Another way to put this is that we are doing standard MDL analysis, but restricting our consideration to the class of models where we know explicitly how to divide the models for each language into a universal part and a language-particular part. This is the essential ingredient for playing the intellectual game that we call theoretical linguistics.

We *might* then imagine you win the competition if you can demonstrate that the final sum that you calculate in (3) is the smallest of all the groups. But this won't work correctly, because it is perfectly possible (indeed, I think it is unavoidable[16]) that two competitors will each find that their own systems are better than their competitors' systems, because of the UTM that they use to find the minimum. That is, suppose we are talking about two groups, Group 1 and Group 2, which utilize $UTM^{\alpha}$ and $UTM^{\beta}$.

[16]If a group wants to win the competition as I have defined it so far, they can modify their UTM to make the Universal Grammar arbitrarily small.

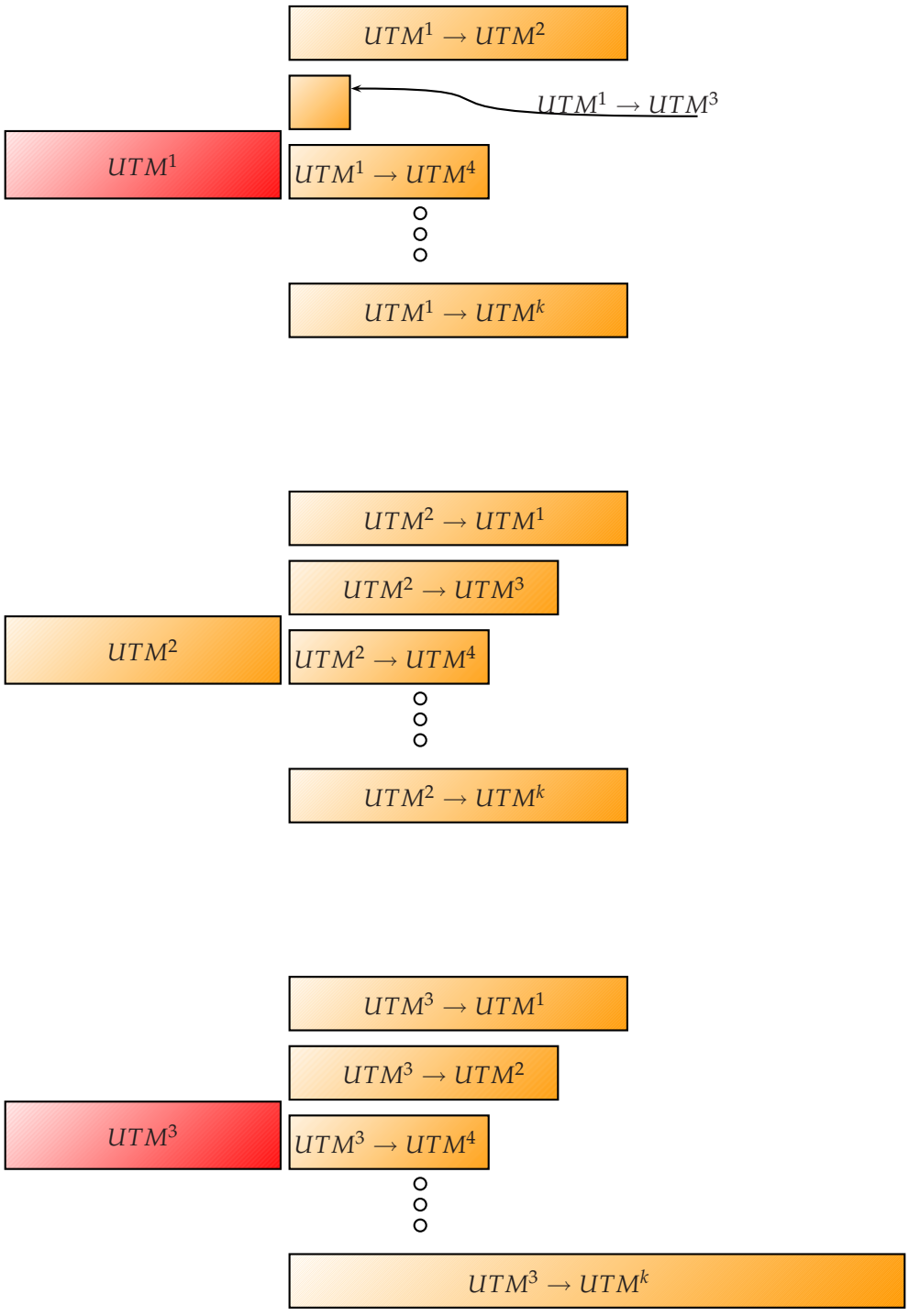



Figure 7: 3 competing UTMs out of $k$

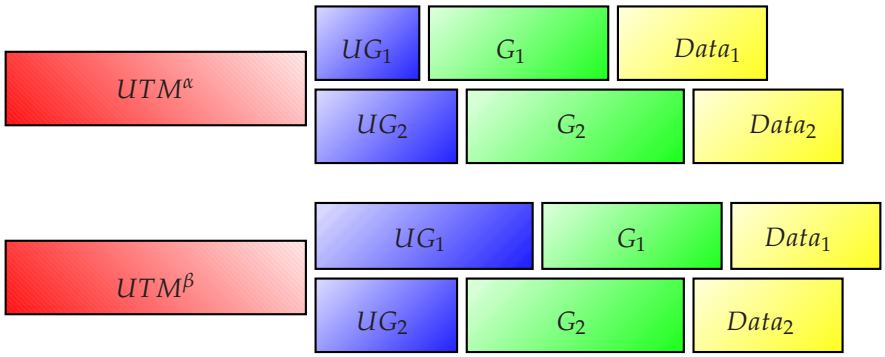



Figure 8: The effect of using different UTMs

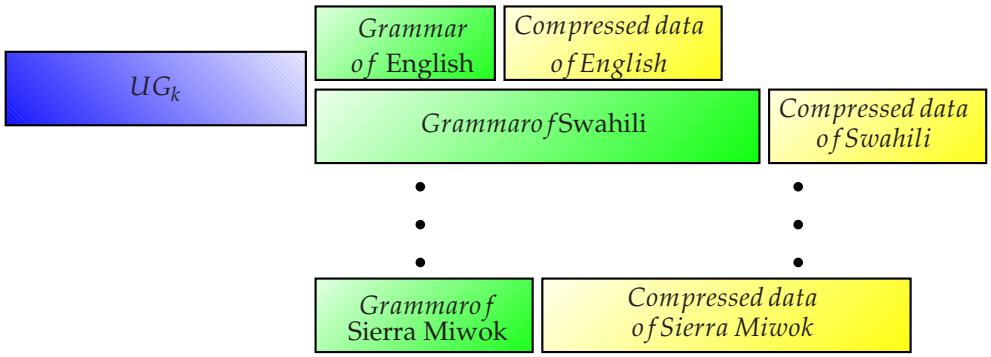



Figure 9: What Linguistic Group k wants to minimize

It is perfectly possible (indeed, it is natural) to find that (see Figure 8)

$$|UG_i|_{UTM^\alpha} + Emp(UG_i, \{\Gamma_l\}^1, \mathcal{C}) < |UG_j|_{UTM^\alpha} + Emp(UG_j, \{\Gamma_l\}^2, \mathcal{C}) \tag{13}$$

and yet, for a value of $\beta$ different from $\alpha$:

$$|UG_i|_{UTM^\beta} + Emp(UG_j, \{\Gamma_l\}^1, \mathcal{C}) > |UG_j|_{UTM^\beta} + Emp(UG_j, \{\Gamma_l\}^2, \mathcal{C}) \tag{14}$$

This is because each group has a vested interest in developing a UTM which makes their Universal Grammar extremely small. This is just a twist, just a variant, on the problem described in the UNIVERSAL GRAMMAR IS FREE fallacy that I discussed above. Comparison of UGs, grammars, and compressed data is made, relatively easily, across different groups of researchers, because for these three things, there is a common unit of measurement, the bit. This is not the case, however, for UTMs: we have no common currency with which to measure the length, in any meaningful sense, of a UTM. We need, therefore, a qualitatively different way to reach consensus on UTM across a group of competitors, our research groups.

### 5.2 Which Turing machine? The least biased one.

With all of this bad news about the difficulty of choosing a univerally accepted Universal Turing machine, how can we play this game fairly? Here is my suggestion. It is tentative rather than definitive, because while it should work among a group of researchers, it is possible to game the system, that is, to intentionally collude in such a way as to act unfairly.

The general problem is this: we have a set of $N$ approved UTMs. A UTM is, by definition, a machine which can be programmed to emulate any Turing machine. Let's denote an emulator that makes $machine_i$ into a simulator of $machine_j$ as $\{i \Rightarrow j\}$, and the length of that emulator as $[i \Rightarrow j]$. We set a requirement that for each $UTM_i$ ($ileN$) in the group, there is a set of smallest emulators $\{i \Rightarrow j\}$ (in a weak sense of the word *smallest*: these are merely the shortest ones found so far).

When a group wants to introduce a new UTM, $UTM_{N+1}$, to the group, they must produce emulators $\{i \Rightarrow N+1\}$ (for all $i \leq N$) and $\{N+1 \Rightarrow i\}$ (for all $i \leq N$). But anyone who can find a shorter emulator can present it to the group's archives at any point, to replace the older, longer emulator (that is, you do not have to be the original proposer of either $i$ or $j$ in order to submit a better (i.e., shorter) emulater $i \Rightarrow j$ for the community).

All of the infrastructure is set up now. What do we need to do to pick the best UTM? Bear in mind that what we want to avoid is the case of a tricky linguist who develops a UTM (let's call it $UTM^*$) which makes his or her UG unreasonably short by putting information that is particular to certain languages into their UTM. It should be possible to discover that, though, by the way in which all the emulators for $UTM^*$ are unreasonably long—longer than the others. That is, the rational choice is to pick the $UTM$ which satisfies the condition:

$$\arg\min_{\alpha} \sum_i [i \Rightarrow \alpha] \tag{15}$$

Such a $UTM$ will be the one which is selected by all of the members of the community as the one which is easiest, overall, for all the other $UTM$s to emulate. A $UTM$ which is unfair is one which cannot be easily reproduced by other UTMs, which is the point of the choice function proposed here. We want to choose the UTM which overall is the easiest for all the other UTMs to emulate. The less this UTM has inside it, so to speak, the easier it is for all of the other UTMs to emulate it; conversely, a UTM into which a lot of specific computations have been loaded will require those specific computations to be added to the emulator of any other UTM which did not build in that specific computation.

## 6 Your Universal Turing Machine or mine?

The case that would concern us is the case where the choice of two different universal Turing machines would lead us to select two different Universal Grammars (UGs). What should we do if we find that at some particular point in our knowledge of languages, there is a UG, $UG_1$, which runs on $UTM_1$, and it outperforms every other UG on all the languages of the world. But on $UTM_2$, $UG_2$ outperforms every other UG, including $UG_1$, on all the languages of the world. We can assume that there is at least one language for which the two UGs select different grammars for the same data; we will restrict our attention to the data from that language, and the grammars for that language.[17] What should we do?

[17] There is a slight irregularity in my doing this, which the careful reader will note. The differences between the ways that the two UGs work on all the other languages are being ignored, and that is not quite right. Again, we're trying to get to the heart of the matter.

If you are following carefully, you will notice that it's not always obvious that we can talk about the *one and the same* grammar being run on two different UTMs each with their different UGs (that is, $UG(UTM_1)$ and $UG(UTM_2)$). What if one of the UGs allows us to refer to "subjecthood",

Back to the problem of whose UTM we are going to use as our reference. Our problem case will arise as follows. Suppose we have data from one language, and two grammars, $G_1$ and $G_2$. If we choose $UTM_1$, then $G_1$ is preferred over $G_2$, while if we choose $UTM_2$, then $G_2$ is preferred over $G_1$. This would happen if

$$|UG_1|_{UTM_1} + Emp(UG_1, \{\Gamma_l\}^1, C\}) < |UG_2|_{UTM_1} + Emp(UG_2, \{\Gamma_l\}^1, C) \tag{16}$$

but

$$|UG_1|_{UTM_2} + Emp(UG_1, \{\Gamma_l\}^1, C\}) > |UG_2|_{UTM_2} + Emp(UG_2, \{\Gamma_l\}^1, C) \tag{17}$$

Here, changing from $UTM_1$ to $UTM_2$ has reversed the order of the inequality. Imagine, if you'd like, that $UTM_1$ permits some important and complex operation to be expressed simply and this operation is used by $G_1$, but $UTM_2$ does not. However, except for that difference $G_2$ is a better grammar, i.e., shorter. Now, because these UTMs are in fact universal, this means that there is a translation program from one to the other, and in fact for each pair of UTMs, there is a shortest translation device used to allow us to simulate $UTM_i$ by using $UTM_j$; that is, we could say that it turns a $UTM_j$ into a $UTM_i$. We indicate the length of the shortest such emulator as $[j \Rightarrow i]$, which is necessarily greater than 0; think of it as "the size of a program that turns a $UTM_j$ into a $UTM_i$". Then it follows that on $UTM_1$, $UG_1$'s analysis of grammar $G_2$, using the best universal grammar it has access to, can never be longer than $UTM_2$'s analysis of the data using grammar $G_2$ plus the cost of emulating $UTM_2$ on $UTM_1$, which is $[1 \Rightarrow 2]$. Informally speaking, a UTM will emulate another machine if the emulation does better than its own native performance, taking the cost of the emulator into account.

If we translate this into inequalities, then we have the following. First, starting from (**??**) and adding the constant $[2 \Rightarrow 1]$ to both sides, we get the following, which says that Group 2 agrees with Group 1 that if you accept Group 1's assumptions, then $UG_1$ is indeed better (I leave $C$ out of the arguments of Emp for simplicity's sake):

$$[2 \Rightarrow 1] + |UG_2|_{UTM_1} + Emp(UG_2, G_2) \geq [2 \Rightarrow 1] + |UG_1|_{UTM_1} + Emp(UG_1, G_1) \tag{18}$$

But from Group 2's point of view, things are the reversed:

$$|UG_1|_{UTM_2} + Emp(UG_1, G_1) > |UG_2|_{UTM_2} + Emp(UG_2, G_2) \tag{19}$$

$UTM_2$'s evaluation of $UG_1$'s complexity (i.e., length) will not be greater than $UTM_1$'s evaluation of $UG_1$ plus the length of the emulator that makes $UTM_2$ behave like a $UTM_1$:

$$[2 \Rightarrow 1] + |UG_1|_{UTM_1} + Emp(UG_1, G_1) \geq |UG_1|_{UTM_2} + Emp(UG_1, G_1) \tag{20}$$

Putting these together, we get

$$[2 \Rightarrow 1] + |UG_2|_{UTM1} + Emp(UG_2, G_2) > |UG_2|_{UTM_2} + Emp(UG_2, G_2) \tag{21}$$

or

$$|UG_2|_{UTM_2} - |UG_2|_{UTM_1} < [2 \Rightarrow 1]. \tag{22}$$

By symmetric reasoning, we obtain:

$$|UG_2|_{UTM_1} - |UG_2|_{UTM_2} < [2 \Rightarrow 1], \tag{23}$$

and

$$|UG_2|_{UTM_1} - |UG_2|_{UTM_2} < [1 \Rightarrow 2]. \tag{24}$$

What this says effectively is this: if you and I use different Universal Turing Machines to analyze a set of data, and my UTM is able to implement my grammar more easily than it can your grammar; and if your UTM is able to implement your grammar more easily than it can my grammar; then the discrepancy in the complexity of the theories of grammars used by my UTM and your UTM is bounded from above by the size of the emulators required by each of our machines to emulate the other.

More specifically, the difference between the complexity of the theory of grammar on my machine $UTM_1$ for my grammar (that's $UG_1$) and the complexity of the theory of grammar that your $UTM_2$ assigns to my grammar must be less than the cost of emulating my machine on yours. If you put it that way, it's obvious.

for example, and the other UG has no way to talk about subjecthood at all? These concerns can make the whole problem very messy. Let's try to keep things simple, and for present purposes assume that any UG can in some obvious sense encode any grammar that another UG can, but the *length* (i.e., complexity) may vary greatly from one to the other. We will also assume that we can make sense out of the idea that one and the same Universal Grammar can appear in two different implementations for two different UTMs. That does not seem problematic, but I am assuming our ability to resolve a number of technical problems.

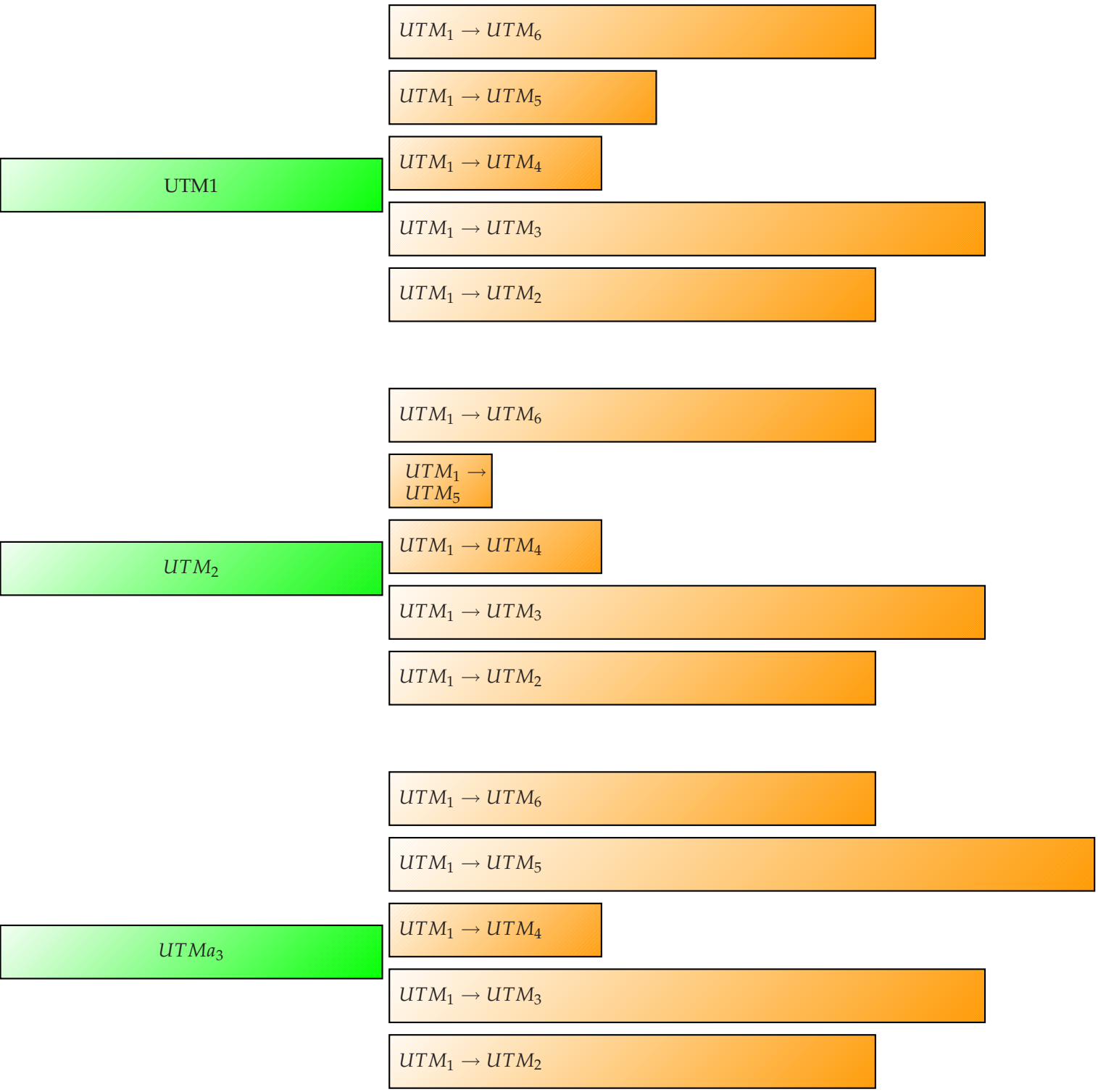



Figure 10: Competing to be the UTM of the year

But the consequence of this are concrete. What we have just seen is that each group can be sure that the extra burden placed on its entry in the competition will be no greater than the size of the emulater of the winning UTM on its own machine. And the winning UTM has been chosen so that the sum of the lengths of the emulators of all of the candidate UTMs to it are a minimum, as in eq (15). [check numbering]

Here is where this discussion has taken us. There is a very reasonable way to define success in linguistic analysis based on Minimum Description Length analysis, and a common corpora that needs an explanation based on a linguistic theory and set of grammars. Where MDL may be thought to break down, which is its inability to choose one UTM and leave the others behind, we can establish a rational program for selecting a UTM out of a set of UTMs competing to be the least biased UTM of all the candidates. While no single linguist is obliged to play this game, so to speak, any linguists who believe that their theories and grammars are the best has a good, and hard, method of proving the point—an opportunity that will appeal to many, it seems to me.[18]

We are very near to a comfortable resting point. We could in fact set up a community of universal grammar writers, and invite each to submit their UTMs along with the best emulators they could construct, and we could determine which was the least biased UTM along the lines we have just sketched. This practice has the advantage that it uses competition in the marketplace as a substitute for any formal proof that we have found absolute minima in our search for optimal data compression.

# 7 So this is empiricism?

We have discussed a particular account of scientific knowledge concerning language in quite some detail.Why should we call this account *empiricist*? On one account, empiricism is the view that all knowledge comes from the senses, and yet we have put a lot of effort into building up knowledge of a priori simplicity based on Kolmogorov simplicity. This is empiricism? This is *just knowledge that comes from the senses*?

No. The empiricist we describe and address in this book is not one who thinks that knowledge begins and ends with what the senses provide. So is it fair to call someone like that an empiricist—someone who cares about theory above and beyond the description of the data?

One reason for calling this approach *empiricist* derives not so much from what empiricism has traditionally *excluded* (which is any knowledge that does not come through the senses), but on what it *includes*. The empiricist believes that a thorough and quantitative consideration of the fit of theory to data is as important as theoretical conciseness, and that theoretical insight must sit shoulder-to-shoulder with empirical covereage in science.

One of the hallmarks of 20th century empiricism was the goal of separating statements into those that represented reality and those that were statements about how words should be used. If that distinction could be made and defended, then the truth value of statements that are about reality can be determined only by looking at the world, and the truth value of statements about how words should be used cannot be determined by simply looking at the world. But the hope that this project could be carried out has long since been dashed, and a naive interpretation of this idea is no longer an option. We do not start from the *assumption* that we know where the separation is between theory and data, and we may even draw the line differently for different purposes on different days. In that regard, the contemporary empiricism that we are exploring in this book differs from the extreme view of logical empiricism (and it is not clear that the historical logical empiricists took a uniformly extreme view along these lines, critics' remarks to the contrary (Quine 2 Dogmas)).

The discussion to this point has offered a justification of an empiricist definition of linguistic analysis. It is empiricist primarily in the sense that it takes as its central task to offer a quantitative account of a body of observations. Of course, one could take the resulting grammar and hypothesize that it was in some sense embodied in the brain, but that would have to be tested in brain-particular ways. It is a call to take both formalism and data very seriously—more seriously, I think it is fair to say, than it is being treated currently. I believe that someone should sit down, right now, and select a particular and convenient universal Turing machine, and write a compiler for a language in which grammars can be written. We can start small, with morphology and phrase structure rules; but let's get started, at least.

## 7.1 Chomsky's critique of empiricism

Noam Chomsky is surely the most articulate spokesperson for the voice critical of empiricism in linguistics, and as one of the many people who have learned a good deal of what they know about linguistics from him, I think it is useful to hear what he has to say on the subject. I do not agree with his view of the relationship of theory and

[18]There are two closely related concerns, however, that we have not satisfactorily addressed. The first is why we chose in (15) [check] to minimize the sum of the lengths of the emulators, rather than use some more complex function (sum of the logs of the lengths, etc.); about this we will say nothing directly. But it remains a possibility that a group could collude to give an unhealthy result in the competition which we have sketched; we discuss this in the appendix to this chapter.

evidence, which bears centrally on the present issue.[19] At times Chomsky has gone so far as to suggest that his method of science is one which allows a serious researcher to ignore data when it is incompatible with his theory. In a recent interview, Chomsky discussed both the methodological notions (which is what concerns us here) and some of the substantive notions involved in minimalism. Chomsky looked at Galileo, and said that

> [w]hat was striking about Galileo, and was considered very offensive at the time, was that he dismissed a lot of data; he was willing to say, "Look, if the data refute the theory, the data are probably wrong." And the data he threw out were not minor. For example he was defending the Copernican thesis, but he was unable to explain why bodies didn't fly off the earth; if the earth is rotating why isn't everything flying off into space?...He was subjected to considerable criticism at that time, in a sort of data-oriented period, which happens to be our period for just about every field except the core natural sciences. We're familiar with the same criticism in linguistics....that's what science had to face in its early stages and still has to face. But the Galilean style...is the recognition that it is the abstract systems that you are constructing that are really the truth; the array of phenomena are some distortion of the truth because of too many factors, all sorts of things. And so, it often makes good sense to disregard phenomena and search for principles that really seem to give some deep insight into why some of them are that way, recognizing that there are others that you can't pay attention to. Physicists, for example, even today can't explain in detail how water flows out of the faucet, or the structure of helium, or other things that seem too complicated....the Galilean style referred to that major change in the way of looking at the world: you're trying to understand how it works, not just describe a lot of phenomena, and that's quite a shift. [**?**]

Chomsky summarizes the Galilean style as "the dedication to finding understanding, not just coverage." Of course that sounds great—the problem, though, is that there is no one who is *against* understanding. Even thoughtful pre-Galilean people were in favor of understanding. No one wants to join a team that declares itself not interested in understanding.

It's certainly a wildly inaccurate description of what Galileo was doing to suggest that his methodological advance was to ignore data, and I find it hard to conceive of why Chomsky would offer that interpretation, other than as a justification for urging others to ignore data when the data contradict their favorite theory. There is a very troubling and disturbing problem we encounter as soon as we undertake to ignore data—does it need to be spelled out? The problem with this methodology is this: it works just fine for me, but it is not fine for you, as far as I'm concerned. I am confident about *my own* ability to identify true conjectures which do not appear to be supported by the data, but I am not so confident about yours. And who knows? You might feel exactly the same way about it, only in reverse. *That's* the problem.[20]

If Galileo's insight was not to ignore data, then what was it? First of all, he came to his work with a deep and thorough skepticism regarding the received truth of the day, which was the Scholastic interpretation of Aristotle. In Chomsky's student days, the equivalent would have been a deep and thorough skepticism regarding American structuralism; in today's world, it would be a deep and thorough skepticism regarding minimalism.

Beyond skepticism, though, Galileo's theorizing was based on two principles before all others, and he could not have said any more clearly what they were: first, that we must look not to books but to Nature, the real phenomena, if we are to understand the world, and second, the language in which Nature is written is mathematical, which is to say, quantitative in character. It was not for nothing that Galileo is remembered for measuring the distance traveled by ball rolling down inclined planes: it was the study of what things really do that allowed him to show that these patterns did not fit the received wisdom of the time, no matter how well those theories satisfied the intellectual palettes of established scholars.

The fact is, there is no philosophy of science that allows one to ignore data. There is something else, though, which we can do when we see our theories running into empirical difficulties: we can acknowledge that our theories are still imperfect, and are inadequate for accounting for many things linguistic. There is no shame in that. There is nothing wrong with a science, such as linguistics, allowing for some research programs to be conducted despite poor empirical results, if there is enough agreement that the hypotheses may pan out someday; this is the scientific equivalent of the "let a thousand flowers bloom" philosophy.

There is a much deeper flaw, though, in Chomsky's appeal. Remember: Galileo *wasn't* a success until his theories had been established empirically, both by matching prediction to observation, and by showing that what had appeared to be false predictions were only apparent, and not real. There's no merit in ignoring data at the time; the only merit is in retrospect, after the stunning predictions actually do match the observations, when the scientist can

[19] If it wasn't obvious already, I might as well acknowledge that this paper is a dialog with a position that fits my best understanding of Chomsky. He has posed many of the right questions in a way that many others might have but didn't.

[20] We do not have to express this in an amusing way. An ethical theory generally adopts some version of Kant's categorical imperative, the notion that we must act only according to a principle that we wish to apply to everyone. But if all researchers wish to give all researchers the limitless right to ignore data so long as they are interested in understanding, not just coverage, we will have chaos rather than a discipline.

pat himself or herself on the back for having never given up on a theory that eventually paid off. Holding on to a theory whoses predictions don't match the facts is like holding on to some stock in the stock market when everyone else says you should sell. You probably *should* sell, but if you don't, and you eventually make a million dollars from it, then you can tell everyone how smart you are. But you can't start telling them how smart you are until the stock actually goes up in value. There are far, far more people who have held onto to theories that never came back to life than there are people whose hunches overcame initial disappointment. It is romantic to think that holding on to a theory that seems to have been falsified is what made Einstein Einstein, but that kind of thinking won't work for cold fusion (or if you are still holding out for cold fusion, choose your favorite once-exciting-but-now-shown-false theory to make my point).

When all is said and done, it would verge on the irrational to deny that the long term goal of our research is to produce theories that simultaneously account for all of the relevant data, and to do so with a minimum of assumptions. The new empiricism offers a way to measure success along these lines. It may not be successful—we may find that probabilistic models cannot be established for some important areas, or that surprisingly arbitrary constraints need to be imposed upon the class of possible grammars. But it seems to me that we stand to learn a great deal from trying it out: we will learn where it succeeds, and I am sure we will also learn in the places where it may fail.

Thus Chomsky's first argument against this sort of empiricism may be summarized (not unfairly, I think) as this: *we should follow the footsteps of the original scientific revolutionaries*. The response to this is that Chomsky has both misread the historical record, and failed to propose a methodological canon that we can all share (that is, it cannot be the case that we all get to choose which hypothesis is maintained regardless of the data; there will have to be shop stewards—or mandarins, or power brokers—who get to decide; I've tried to suggest that this is a hopeless and unattractive position to maintain).

Chomsky has offered a different argument, and one that carries more conviction, I think, but it too is based on just exactly what it is that we mean by *science*. His argument is that linguistics is either about something in the real world, or it is not. If it is about something in the real world, the only reasonable candidate about which linguistics can make claims is the human brain. If linguistics is not about the human brain, then it is not about anything in the real world, and there is therefore no truth of the matter, and therefore any linguist is free to believe anything s/he wishes to believe, and there are no scientific guidelines or standards—and in particular, linguistics is then not a science. Hence, if linguistics *can* be a science, then it *must* be a science of the brain.

While that is my summary of Chomsky's idea, it has been laid out explicitly in a number of places. Here is one place, where Chomsky is responding to critics whose criticism he finds impossible to fathom, linguists who do not believe that they are making claims about the human brain:

> Since there are no other objects in the natural world that the linguist's theory is about, the demand apparently is that the linguist construct a theory of some non-natural object. Again the tacit—and sometimes explicit—assumption seems to be that there are entities independent of what people are and what they do, and these objects are what theories of language are about, and further, must be about, on pain of irresponsibility. Again, we are left in the dark about these curious entities and how we are to identify their properties. Considerations of communication, the theory of meaning, the theory of knowledge, and folk psychology have also been adduced to argue that there are independent entities, external to the mind/brain, of which each of us has only a partial and partially erroneous grasp, always leaving as a mystery the manner in which they are identified, except by stipulation, and what empirical purpose is served by assuming their existence. I think there are ample grounds for skepticism about all of these moves...
>
> Adopting this approach, we abandon the hopeless search for order in the world of direct experience, and regard what can be observed as a means to gain access to the inner mechanisms of mind.[21]

There are two central points here: the first is whether linguists must declare their theories to be about minds (or brains) or else suffer the conclusion that theirs are not theories at all; the second is whether the search for order in the world of experience (or "direct experience," as Chomsky calls it) is an important part of the scientific tradition from which we hope to learn, and to which we hope to contribute.

Do linguists know what objects their theories are about, some or all of the time? Let's put that question on hold for a moment. Do scientists in general know what their theories are *about*? The answer to this latter question is not as unequivocal as we might think. It is not infrequently the case that creative new scientific theories come along with a nagging uncertainty as to what they are about. Newton's theory of gravitation was the first classic case of this sort: Isaac Newton, like all the other great thinkers of the scientific revolution, was committed to a mechanistic world-view, one in which the only sensible way in which objects could interact was through immediate contact and collision. And yet his theory of gravitation flew in the face of this, and many of his contemporaries were aghast

[21] Chomsky 1997[**?**], p. 18-19; all citations in this section are from that source.

at the theory's inability to give an account of what this thing could be that leads to the action at a distance that we observe and call "gravity." Newton was as unhappy as anyone, but he could not deny the compelling force of the mathematical model that he developed which allowed an elegant and powerful account of motion both on the Earth and in the solar system. Mathematics and accurate prediction, according to Newton, trumps being able to say what in the real world the theory is *about*. (In the end, it was not until the beginning of the 20th century that a new idea came along—that of space-time with an inherent curvature—that allowed us to say what the theory of gravitation is about, and it did it by throwing out, in a sense, all of the substance of Newton's ideas.)

Mendel's theory of the gene is another classic example of a scientific theory which does not know what it is *about*. Mendel's account, followed up on by many others, was based on finding repeated statistical effects in the distribution of traits in successive generations of plants, fruit flies and everything else. But it was not until the discovery of the form and function of DNA nearly a century after Mendel that we began to know what kind of thing a *gene* is in the physical world. Before then, biologists developed a theory of genetics and inheritance without knowing what theirs was a theory *of* in the physical world.

A third example along these lines is that of the wave theory of light. By the end of the 19th century, there were strong proponents on both sides of the divide between those who viewed light as a wave of something, and those who viewed it as movements of particles.[22], but overall, it is probably fair to say that the wave-theorists had the strongest arguments when all is said and done: they could give an account of wave interference patterns that was very hard for the particle theorists to counter. And they gave a name to the substance that light was a vibration *of*: they called it *ether*. And they were utterly certain that ether existed, because after all, they had a great scientific theory, and it had to be about something in the physical universe. But they were simply wrong: there is no ether, and while their theory based on ideas involving vibrations led them to good models for a large number of physical phenomena, it turned out that all along, they had been wrong in thinking that they even had a clue as to what their theory was about, at a scientific level: we had to wait for the quantum theory of the photon to reach a point where we could say that we have a reasonably good idea of what the theory of light is *about*.

What all these cases—Newtonian gravity, Mendelian genes, undulatory theory of light— have in common (and we could have chosen many others to make the same point) is that careful and detailed observation and measurement were compared with quantitative models, usually but not always under controlled conditions, and convincing cases were made for the ability of formal and mathematical methods to model the data.

I have gone into this matter in this detail because I want to emphasize that the empiricist model of linguistics as a science which I have sketched is not intended to be a partial approximation to some larger scientific model which will include a brain, and without which linguistics will have no sense or meaning. Linguistics is already a science of the data that linguists can argue need to be accounted for, and the bayesian methods that we have explored here provide an unambiguous account of what it means to compare two or more theories.

There is a flip side to this as well. I believe that many linguists think that they believe their theories are about minds and brains, but do not *act* as if they believe it. The vast majority of linguists do not pursue their linguistic theories with the goal, direct or indirect, of establishing a location in the brain and a function in the neural circuitry for their syntactic or phonological components. At best, they offer an I.O.U.—that is, a promise that at some unspecified future date, a physical location in the brain will be found. Our view is that *there is no difference between promising, some day, to do something in a yet undiscovered way, and not promising to do it at all.* It is not the linguist's job to determine how the brain works: that is a good thing, since there are few linguists with any serious training in neuroanatomy. It is the linguist's job to figure out how language works, and as that challenge continues to be handled, linguists and neuroscientists will be able in the future to come up with a synthetic view.

Consider a well-establishesd theoretical concept in phonology: take the mechanisms of autosegmental phonology, as applied to problems of analyzing Bantu tone languages, for example. To the best of my knowledge, despite more than thirty years' of work on the linguistic side of things, there is no evidence that the brain employs autosegmental representations—nor even a clear specification of what kind of data could be interpreted as confirming, or disconfirming, that hypothesis. And yet the phonologist's confidence in the model is not shaken; it is not the case that the linguistic arguments for autosegmental analysis were not very good, that they were the best we could come up with at the time while we waited for neurosciences to make good the promise to test some hypothesis about neural circuitry that the linguist came up with.

The naive realist[23] against whom I am arguing thinks that the empiricist is deluding himself; the naive realist thinks that the empiricist really *does* believe that the objects described by the theory do exist, and that is why even the empiricist wants to scour the universe to discover whether neutrinos and gravitons and Higgs bosons and autosegments and all sorts of exotic theoretical objects exist. The naive realist tells the empiricist: you see! you

[22] cite Mertz vol. 2

[23] Not everyone who uses the term "naive" takes the trouble to define it, which leaves a reader's mind willing to read too much or too little into it. I will say what I mean by this characterization. I think that a naive defender of a position is one who thinks that his opponents disagree with him because they have not even considered the evident, obvious reasons for his position. In short, the naive realist does not realize that there are intellectually valid and defensible positions different than his own, and thinks that empiricists say disagreeable things because they haven't thought much about the problem.

really *do* want to test whether the things that you postulate exist. If you can't find them, you (or your colleagues) will no longer be satisfied with your theory. Just accounting for the observations isn't enough, if you know that the entities you postulate to account for those observations cannot be found when the tests are run.

The empiricist thinks that the naive realist gets carried away. The empiricist can love an elegant theory as much as the next person (and maybe more), but he knows that history has shown that dragging the entities postulated by a theory into the world of what is observed is difficult and treacherous.

It's not difficult the way it is difficult to drag up a ship that has sunk to the bottom of the sea; it is difficult the way it is to get a stain out of a rug, it is difficult the way it is to find an equitable way to share the wealth of our society. These are difficult things to do, and even if we try hard we may not succeed in accomplishing these tasks, and it does not mean that we were misguided by trying to accomplish them. But our best hopes may not be met. That is how the empiricist feels about theoretical entities: it is very reasonable to undertake to find them in space and time, if they are entities that are postulated to exist in space and time. If we find them, we will have enhanced our observed universe. But science continues, and it continues in its full-bore fashion with no problems at all, thank you very much, whether those theoretical entities are observed, or observable, or neither.

The realist replies: if you do not believe that those entities really exist, then you cannot believe, really and truly believe, the scientific theory, and you do not have a real explanation for the regularities you have found in the data and the observables until you acknowledge that you also believe that the unobserved things the theory postulates do in fact exist, so that they can actually *cause* the things observed to come into existence.

The empiricist realizes that the stakes have grown larger now. The realist has just accused him of bad faith, deep down inside. But the fact is that even the linguist who rejects empiricist philosophizing has a very limited kind of belief in the reality of the theoretical entities he says he believes in.

The empiricist's reply to the naive realist is this: you did not really believe that your unobserved entities existed. If you want to insist that you really do believe they exist, you are going to have to acknowledge that it is in a rather different sort of way than the way in which you believe that the Empire State Building exists, or the Pentagon, or your left foot. Because we all know that scientific theories change, and what was once strong motivation for believing that something (like caloric or phlogiston, or the ether of which electro-magnetism is but a vibration, or the passive transformation if you are a generative grammarian) may tomorrow burn off like the morning dew. You, naive realist, you are willing to change your view as to what exists on the basis of learning a better theory! The very foundation of your belief in theoretical entities is the conciseness of the theory that links the theoretical entities to the observations, and if there is a better way to account for the observations, you have no trouble at all dropping yesterday's belief in phlogiston or grammatical transformations. You would never do that in the case of a physical object: if the World Trade Center is gone today, it is because something happened to it, not because we have a better theory today that does not need it anymore (whatever that might mean!). In short, the law of rejected entities: both realists and empiricists say good-bye to rejected theoretical entities with no trouble at all.

# 8 Doing linguistics this way

It may well appear to the reader that the discussion to this point has been abstract, and distant from the working life of the linguist. But the fact is that the ideas presented in this paper have all emerged out of very concrete research projects, and in this section I will describe how the study of morphology can be pursued in a way that can be derived from the empiricist principles we have discussed. *Linguistica* is an open-source unsupervised morphology learning program which has been described in a series of papers.[24] The goal of this project is to determine whether an empiricist program of the sort that I have outlined in this paper can succeed in inducing a natural language morphology. In reality, work of this sort means investigating and exploring morphology-induction hypotheses, and trying to learn from both the successes and failures what changes need to be made to make the system more successful—where "success" means that the system is able to take a large corpus of words from an unknown language (unknown to it, in any event), and parse the words into morphs, develop a morphological grammar in the form of a finite-state automaton, and propose hypotheses regarding morphophonology, that is, the changes in the shape of morphemes that occur under the influence of near-by morphemes (e.g., the plural *-s* suffix in English is preceded by a schwa after strident coronal consonants).[25]

*Linguistica* in effect takes in a set of words as its data, and produces a probabilistic grammar that generates those words (and perhaps unseen other words as well). It knows essentially only this: that it would like to maximize the probability that it assigns to the data, and it would like to keep its grammar as small and simple as possible. It could maximize the probability of the data by having a trivial morphology that generates each word in the corpus as an unanalyzed string, and assigning to each word precisely the frequency with which it occurs in the data; it is a mathematical fact that such a system would assign the highest probability to the data, among all the ways that

[24] See http://linguistica.uchicago.edu for the program, as well as technical discussions in [**?**], [**?**], and a more general overview in [**?**]

[25] The *morphs* of a *morpheme* are the various phonological realizations of that morpheme.

could be considered. But such a morphology is unreasonably large, and fails to capture any generalizations at all, and hence is bloated, to the point of morbidity, by over-representing material in the morphology (which is nothing more than a word-list). If the data includes *linguist*, *linguists*, and *linguistic*, it will fail to notice that the string *linguist* has occurred three times, and that string can be augmented (so to speak) by the addition of suffixes ($-\emptyset$, *-s*, *-ic*) that can be similarly employed throughout the morphology.

At the opposite extreme, the morphological grammar can be made very simple if we allow it to generate all sequences of letters (or phonemes): this grammar is very small and simple, but it assigns extremely low probabilities to each of the observed words—it is helpful to bear in mind that the more separate *pieces* an analysis posits in the data, the smaller (all other things being equal, or even roughly equal) will be the probability assigned to the data.

*Linguistica* proposes and evaluates a sequence of states, beginning with one in which each word is unanalyzed, achieving initially a great deal of savings in the spelling out of the morphology by virtue of extracting redundancies, which is to say, the existence of morphs in the analysis of particular words. Positing a suffix *-s* increases the complexity of the morphology as a graph *per se*, but it greatly reduces the complexity of the labels on the edges of the graph, and thus on balance decreases the complexity of the lexicon.

But the model of learning that emerges is not at all one in which a series of ordered procedures must be followed to the data in order to arrive at the ultimate grammar, along the lines that have been attributed to American structuralist linguists in the 1940s and 1950s. The learning system spends considerable amount of time considering alternative analyses that it ultimately rejects, rejecting them on the grounds that all things considered, they are not superior to the analysis that has been considered so far. As long as we have a good grammar evaluation model, we can be less concerned with a language discovery device that considers possibilities that are incorrect for the language in question.

So when *Linguistica* explores the data of a European language like English or French, it does a very good job of discovering stems and affixes; it runs into trouble when a morpheme has two possible realizations: the final vowel of *beauty* is distinct from the second syllable of *beautiful*, from both a phonological and an orthographic point of view, and the analysis of *beauti+ful* as stem plus suffix can be strongly suported only once the alternation between beaty- and beati- (so to speak) has been considered as a candidate generalization in the language. Thus to get *Linguistica* to better learn morphology, it is important to give it the smarts to induce morphophonological generalizations, which in the final analysis are generalizations which allow the morphology to be simplified.

Knowledge of syntax can be important in inducing the correct morphology as well, to be sure: again considering English, while most pairs of words of the sort *dim, dimly* are adjective and adverb pairs, there are many that are noun/adjective pairs, such as *friend/friendly*.

Development of *Linguistica* allows a range of important questions to move from highly theoretical to plainly concrete. Consider the set of stems that it discovers in English that appear with the suffixes $-\emptyset, -ed, -ing, s$, and those that it finds that appear with the suffixes $-\emptyset, s$. It would be wrong to conclude that the those in the second set (which are nouns, of course) are of the same morphological category as stems of the first set (verbs like *jump* or *walk*), but on what basis should that conclusion be drawn? There is a great deal of evidence that supports the conclusion, notably the fact that there are noun stems that occur with very high frequency in a text without either *-ed* or *-ing* appear after them: the effect of this kind of negative evidence is easy to build into the learning model. But it would not be unreasonable to attribute to the language learner the *apriori* concepts of noun and verb, associated with the semantic notions of entity and predicate, and use this knowledge to induce the morphology learner to associate different morphological patterns with the two distinct concepts. (We do not in fact do this in Linguistica, but this is a case where a not unreasonably linguistic bias could play a role in correct morphology induction.)

On the whole, Linguistica does quite well, though the project is certainly closer to its beginning than it is to its completion. Most importantly, though, it is a concrete embodiment of the kind of empiricist language-learning described here.

# 9 Conclusion

The observations I've made in this paper undoubtedly sound quite abstract, but they do have a very practical side to them, as I've briefly sketched in the preceding section. The discussion here has been based on Minimum Description Length (MDL) analysis (Rissanen 1989 [**?**]), and MDL analysis lies at the heart of *Linguistica*; MDL and related approaches are being pursued by a number of researchers at this point.[26] I believe that the approach discussed here can be applied quite directly to linguistic problems, and the reader is welcome to see an example of that at the website indicated. In fact, the order of things has been quite the opposite of what might appear to be the case, given the present paper: in actual fact, the concrete applications came first, and the theory came later (which is the usual order, in the lives of theories). The goal is to develop an understanding of what it means to develop a

[26] Any list that I could provide of such work would inevitably leave scholars out who deserve to be cited, and so I will abstain from providing such a list; an internet seach on the terms *"unsupervised learning grammar"* will generate a long list of candidate resources in this area. Add reference to [**?**].

grammar of a set of data which is explicit enough that it embodies the considerations that a human linguist applies in determining what is the best grammar for the data.

The hypothesis that I have made in this paper is very simple: that a universal measure of algorithmic complexity is enough to provide an explanation for properties of grammars. This may not be true, from the point of view of neuroscientists: it may be that it is necessary to define, for example, a highly restricted subclass of grammars that are possible human grammars, because we discover that the algorithmically simpler ways of accounting for the data in these languages is not the way used by the human brain. I don't think that there is any reason for such pessimism at this point, but it is certainly possible in principle.

But the main take-home point is that algorithmic complexity, working together with probabilistic grammars, allows for a very appealing conception of what linguistics is, and developing an empiricist conception of the task that is remarkably true to the spirit of Chomsky's *Logical Structure of Linguistic Theory*.

## 10 Acknowledgments

I'm grateful to several people for conversations on these topics over a long period of time, including Carl de Marcken, Mark Johnson, Bernard Laks, Partha Niyogi, Aris Xanthos, Antonio Galves, Jason Riggle, Jorma Rissanen, Jens Erik Fenstad, Pierre Laszlo, and somewhat earlier, at a time prior to the principles and parameters approach, Noam Chomsky; and special thanks to Jason for finding several errors. I first encountered the use of probability theory for linguistic encoding in de Marcken's work, notably [**?**], which contains an elegant presentation of the idea. This paper was written while I was a guest of the Centre national de la recherche scientifique at Université de Paris X, and I am grateful for that support.

## References

[1] Adriana Belletti and Luigi Rizzi. An interview on minimalism with Noam Chomsky. University of Siena, Nov 8-9, 1999 (rev: March 16, 2000), 1999.

[2] Noam Chomsky. *The logical structure of linguistic theory*. s.n., s.l., preliminary draft. edition, 1955. Noam Chomsky. 29 cm.

[3] Noam Chomsky. *Syntactic structures*. Janua linguarum, series minor, nr. 4. Mouton, 's-Gravenhage, 1957. by Noam Chomsky. Bibliography: p. [115]-118.

[4] Noam Chomsky. *Language and Mind*. New York: Harcourt Brace, 1968.

[5] Noam Chomsky. *The Logical Structure of Linguistic Theory*. Plenum, New York, 1975[1955].

[6] Noam Chomsky. Language and cognition. In David Johnson and Christina Erneling, editors, *The Future of the Cognitive Revolution*. Oxford, 1997.

[7] Carl de Marcken. *Unsupervised Language Acquisition*. PhD thesis, MIT, 1996.

[8] John A. Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198, 2001.

[9] John A. Goldsmith. An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, 12(4):353–371, 2006.

[10] John A. Goldsmith. Probability for linguistics. *Mathmatiques et Sciences Humaines*, 2007.

[11] Sharon Goldwater. *Nonparametric Bayesian Models of Lexical Acquisition*. PhD thesis, Brown University, 2006.

[12] Shalom Lappin and Stuart M. Shieber. Machine learning theory and practice as a source of insight into universal grammar. *Journal of Linguistics*, 43:1–34, 2007.

[13] Ming Li and Paul Vitnyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Verlag, Berlin, 1997.

[14] Jorma Rissanen. *Stochastic complexity in statistical inquiry*. Series in computer science ; vol. 15. World Scientific, Singapore ; Teaneck, N.J., 1989. 89024817 Jorma Rissanen. Includes bibliographical references and index.

[15] Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, 1989.

[16] Ray Solomonoff. The discovery of algorithmic probability. *JCSS*, 55(1):73–88, 1997.