FORMS AND FORMALISATION. AN ESSAY ON FORMAL LINGUISTICS.

Dafydd Gibbon
University of Bielefeld

1. Forms and formalisation

Perhaps paradoxically, this contribution on formalisation is more of an informal essay than a formal study. It is intended, in the context of concerns about formal linguistics, to provide a metatheoretical context for these concerns, and to stimulate reflection on the extent to which a topic like formal linguistics is a "standalone" field, and to what extent formal linguistics (an ambiguous term) and other approaches to linguistics are interdependent. In this essay, after a characterisation of formal and functional linguistics, some aspects of formal linguistics are discussed, in the sense of formal linguistics which regards mathematical formalisation as an indispensable intellectual tool for understanding language, for linguistic theory formation, and for computational applications of linguistics such as word processors with their spell-checkers, grammar checkers and dictionaries and character encodings, or dictation software.

A somewhat artificial distinction is made between two mainstreams in linguistics: functionalist and formalist approaches. Like many binary distinctions, this concept pair is based on an over-simplification: there are many kinds of functionalist approaches and many kinds of formalist approaches, and the simplified properties often associated with each – the 'hard-headed formalist computer freak' versus the 'soft-bellied interpretive functionalist' – tend to generate controversies which are very entertaining, but intellectually misleading.

On one reading, functionalists start with a holistic, interpretive view of the content and functions of language in communication and ignore the forms of language while formalists start with an analytic, positivistic view of the observable expression or forms of communication and ignore the functions of language. This reading is rather odd: entities such as concepts, objects, events and states exist – one assumes – independently of language, unless one is an extreme linguistic relativist. Meanings are different from these entities: either they are language-internal and intensional, defining the interpretability of signs, or they are extensional and define relations between signs and their denotations in the real world. Either way,

meanings presuppose forms – meaningless forms can be met everywhere, for example in the more obscure parts of the present paper. No form, no function; the reverse is not true.

On another reading, functionalists disapprove of formal theories and models as being static artificial constructs which have nothing to do with the natural dynamic processes of language use, while formalists claim that only formal theories and models can lead to a deeper understanding of the regularities of human language behaviour. Again, this distinction, sometimes formulated, for instance, by the more way-out followers of Natural Linguistics, confuses categories: regular natural dynamic processes in the physical, physiological, psychological or social worlds are not possible without firm foundations in the structure of these worlds. The laptop on which this essay is being written provides many examples: the simplest is the keyboard – a static "quertz" structure (it is a German keyboard) which does nothing if the laptop is not switched on. However, after switching on it is polled thousands of times per second in a querying process by the input processor, checking for key hits; and still does nothing visible to the user until the keys are depressed by the fingers in a dynamic and interactive process. No structure, no process; the reverse is not true.

On a third reading, functionalists are mainly occupied with the functions of language in society, and formalists are mainly occupied with the mental language faculty or neurophysiological underpinnings of language. Clearly these are orthogonal perspectives on language and do not exclude each other. No individual without a society, no society without an individual.

These three readings of the distinction are highly misleading: the readings overlap in complex ways. For example, interpretative conversation analysts who use participant observation methods and are concerned with conversational interaction are centrally concerned with forms or 'markers' in aspects of conversation such as turn-taking and the patterns they form. In natural linguistics, the term "natural" refers to the domain of investigation and domain-oriented criteria of evaluation, while "formal" in the sense used here is a methodological term and therefore orthogonal to "natural", and not in conflict with it. And formal semanticists and pragmaticists are not only concerned with formalisation but with understanding as precisely as possible the meanings and functions of language in the real world.

The reality is that all of these approaches select in different ways from the same semiotic domain of human communication in its various forms of speech, gesture,

acoustic surrogates, and writing, and can be located in the same complex *theory space*, suggesting that it will be worthwhile to pursue the long-term interdisciplinary goal of creating a unified semiotic perspective of language which neglects neither forms nor functions, and neither interpretation nor formalisation.

2. Linguistic Theory Space (LTS)

What might the metatheoretical theory space, in which approaches to linguistic observation, description and explanation are located, look like? If there is a consensus in the field, it will look something like this. Linguistic Theory Space (let us call it LTS) has three dimensions defining the *domains*, the *empirical methods* and the *formal methods* of theories.

The *domains* of linguistics are semiotic: investigation of the physical and physiological patterns of speech, writing, gesture and speech surrogates is not possible without at least some prior acknowledgment of their semiotic function; otherwise there would be no criterion for picking out communicative forms from any surrounding noise or visual impressions. A phonetician who investigates speech sounds, the pronunciation of words, or phrasal and discourse prosody knows beforehand that these events have a function in communication, and delimits them from other events. Likewise, a conversation analyst knows that where there are no forms there are no functions, and that the analysis of functional aspects of language therefore requires the analysis of forms.

The *empirical methods* of linguistics are very varied, and depend on different definitions of empiricism which range from positivist approaches which ground all knowledge in observable states and events in the physical world, on the one hand, to idealist approaches which regard it as naive to rely on a supposed objective reality and give highest priority to the human faculties of interpretation and understanding as sources of knowledge, on the other. Both these extremes, as well as the intervening more hybrid methods, use both qualitative and quantitative methods: the qualitative methods define which aspects of the domain are selected for observation, analysis and interpretation, and the quantitative methods define the validity of the results of observation and interpretation. So there is a place in phonetics for physical observation, measurement and statistical analysis as well as for subjective perceptual tests and statistical analysis of these, and there is a place in sociolinguistics, for example, for functional interpretations of language in context, and the achievement of validity through scientific consensus on interpretation, as

well as for observations of which forms have these functions in which contexts, and for statistical analyses of these on the basis of the accepted consensus which statistical methods share.

The starting point for *formal methods* of linguistics is *systematic textual description*, perhaps the most common form of 'formalising' the domains, methods and models of a particular approach. The other end of the formal methods scale is a *mathematical formalisation* which describes a particular model of the world, and which is based on a logic (such as predicate logic, modal logic, default logic, attribute-value logic) or an algebra (such as formal grammar theory, automaton theory) or another formalism such as graph theory. Mid-way between the systematic textual description and the formalisation, in this strict sense of the term, are various techniques of *semi-formal symbolisation*, using abbreviatory techniques and graphs for representing structures and processes. An illustration of this distinction can be found in the traditional procedure of parsing sentences. In a systematic textual description, a sentence is described as being divided into parts (hence 'parsing', from Latin 'pars' meaning part); these parts are further divided into smaller parts, until the smallest 'parts of speech' are identified. In a semi-formal symbolisation, often used in descriptive linguistics, these nested parts may be represented as tree graphs or as boxes-within-boxes and by constituent structure rules. In a mathematical formalisation, these structures may be defined by means of Predicate Logic (with such predicates as dominance and precedence), or by Formal Grammar Theory, a branch of algebra defined by Chomsky for linguistics and also used in Computer Science, which defines a hierarchy of increasingly complex grammars: for example, the formal grammar which defines branching tree-like structures is known as a Chomsky Type II Grammar, or Context-Free Grammar. Another type of formalisation, which permits the definition not only of declarative information, i.e. facts and generalisations (rules, constraints), but also of procedural information (algorithms, procedures, processes) is Automaton Theory, another branch of algebra which is rather analogous to Formal Grammar Theory, and which provides general definitions of machines which can compute grammars, structures, and sentences.

So, simplifying for the sake of summarising: first, analysis of functions without forms is not possible, but analysis of forms without functions is arbitrary; second, functionalist interpretation and formalisation based methods are not mutually exclusive.

3. Using LTS to parametrise linguistic frameworks

LST is not only a useful conceptual tool for understanding how to go about linguistics in general, it also provides a useful conceptual tool for classifying frameworks for linguistic observation, description and explanation, such as the so-called functionalist approaches of the Prague School, Halliday, Givon or Dik, and the so-called formalist theories of Bloomfield, Hjelmslev, and of Chomsky and his many followers. Without a parametric space of this type for defining commonalities as well as differences, it is all to easy to fall into the trap of intellectual laziness which assumes that each school of linguistics partitions theory space into its own exclusive domains and methods, is therefore totally different from other schools. Thereby communication and interchange between different schools becomes impossible. With a parametric theory space, a more creative approach becomes possible, in which different approaches select different areas of theory space, and can be seen to overlap more or less along each dimension.

Of course LTS is also a simplification: each of the dimensions is highly complex. Still, LTS does approximate to a consensus on what constitutes linguistic theories, and can be refined into a more highly granular metatheoretical model as the need arises, and applied in characterising, comparing and evaluating different linguistic approaches.

4. From characterisation to formalisation

Carnap, the Vienna and Chicago logician, described formalisation as a process leading from the *characterisation* of an intuitively understood concept through systematic *symbolisation* to mathematical *formalisation*. This intuitively understood concept is taken from the domain of the theory, characterised by an appropriate model. Of the three dimensions of LTS, clearly a precise characterisation of the domain – whether phonetics or phonology, morphology or grammar – is indispensable. Oddly, it is to this dimension that linguists have often paid the least attention. Consider two cases which are practically never queried in linguistics: the case of the *phoneme* and the case of *levels of description*.

First, the case of the phoneme. A common definition of the phoneme is “the smallest meaning-distinguishing unit of speech”. This seems fair enough at first glance. It is a specific kind of definition, found in dictionaries, called in traditional terms *definitio per genus proximum et differentiae specificae*, i.e. by nearest kind (i.e. *unit*, *unit of speech*, ...), and specific differences from other kinds – there are

other kinds of unit of speech. The expression to be defined, the *definiendum*, is "phoneme" and the defining expression, the *definiens*, is "the smallest meaning-distinguishing unit of speech". The basic idea is that an unknown term, in this case "phoneme", is defined by means of known terms, in this case "smallest meaning-distinguishing unit of speech". Most of the terms in the defining expression are of course known – but what is the meaning of "meaning"? Who can define it or demonstrate it? Meaning is the subject of an entirely different linguistic subdiscipline and as such can hardly be taken to be a basic concept in phonology and as part of definition of phonological terms. So what would be a better component of the definiens? Perhaps simply "word" - everyone, linguist or not, knows intuitively what a word is (well, maybe linguists don't) and can define or demonstrate words, even though the boundaries of the concept may be fuzzy. A scientifically valid definition would then be "a phoneme is the smallest word-distinguishing unit of speech". Simple. One might object that an appropriate definition might also be "a phoneme is the smallest sentence-distinguishing unit of speech", which is also true. However, changing a phoneme changes a word, and changing a word changes a sentence, so it follows automatically from the first definition, and the definition of a word, that phonemes are also the smallest sentence-distinguishing units. On the other hand, changing a sentence does not necessarily imply changing the words in it – sentences may be distinguished by different intonation patterns, too.

Second, consider the common ordering of levels of language description into a series "phonetics – phonology – morphology – syntax – semantics – pragmatics", where the ordering is represented by a dash. When closely scrutinised, this ordering does not make too much sense, because the dash has different meanings in different positions, and certain important components of the architecture of language are completely missing, such as prosody and the lexicon. The "allo-dashes" in the above representation of the common ordering of levels can be modelled as follows:

1. The phonetics-phonology relation is *realisation* by a physically describable entity (phone, allophone) of an abstract, relational entity (phoneme).
2. The phonology-morphology relation describes an *encoding* relation of semantically meaningful units (words, morphemes) by distinguishing units (phonemes).

3. The morphology-syntax relation describes a *constituency* relation: morphemes are parts of sentences (perhaps indirectly by being parts of words which are parts of sentences).
4. The syntax-semantics relation describes a *denotation* or *connotation* relation between complex signs (sentences) and their meanings (note that a level of words or lexical items and their meanings is absent in this series).
5. The semantics-pragmatics relation describes a relation of *usage* between what Grice has called "utterance meaning" and "utterer's meaning".

What do we learn from this? We learn that the traditional ordering of levels of description is unsuitable as a characterisation of the architecture of language and therefore as a starting point for formalisation, because is inconsistently defined. A better model is one which consistently uses just two relations, *constituency* and *interpretation*:

1. *Constituency*: morphemes are constituents of words (possibly via a constituency hierarchy of derived and compound words); words are constituents of sentences (again, via a constituency hierarchy, this time of phrases); sentences are constituents of texts; texts are constituents of dialogues. Each of these levels has its own "grammar". This is a rank hierarchy of a type common in functionalist linguistics. Oddly, linguists usually stop at sentence level; in modern computational linguistics and text technology, however, grammars for documents and dialogues are required (for instance to generate documents systematically from a search engine database).
2. *Interpretation*: units of all ranks (morphemes, words, sentences, texts, dialogues) receive two interpretations in terms of models of the real world, a meaning interpretation (in linguistics: semantic and pragmatic interpretation) and a modality interpretation (in linguistics: phonetic or orthographic interpretation). The modality interpretation of writing (in terms of fonts, layout etc.) is entirely analogous to the modality interpretation of speech (in terms of phones, tones, etc.), the difference being only that one is in the visual modality and the other is in the acoustic modality.

In this architecture, morphemes receive phonetic and semantic interpretations, for instance; words also receive their own phonetic and semantic interpretations, sentences, texts, dialogues each receive their own phonetic and semantic interpretations. The semantic interpretation of a sentence, for example, is a proposition, its pragmatic extension is a speech act; the phonetic interpretation of a sentence is its prosody. At each level, items may be lexicalised, i.e. inventorised, or constructed by rule on the fly: morphemes may be invented, words may be invented, sentences may be invented etc., or morphemes may be lexically listed, words may be lexically listed, sentences may be lexically listed (as idioms or proverbs, for instance), texts may be lexically listed (as quotations, poems, etc.).

So what do we learn from the two cases? We learn to be critical and consistent in defining terms and modelling the structures of language; that these terms and structures have to be suitable for formalisation is a very helpful heuristic motivation.

5. Is formalisation necessary?

Looking at the major achievements of linguists - philologists as they were generally known at the time - in the 19th century, and grammarians, logicians, rhetoricians and philosophers of language before that, one sometimes wonders what linguistics has gained in the past hundred years or so. Linguistics, as we know it, did not emerge until the 1920s. During the 19th century a great leap forward in language studies was made, and by 1900 the giant achievement of classifying the Indo-European languages had been attained, basic distinctions between levels of description of language form - phonetics, phonology, morphology, syntax - had been thought out, the International Phonetic Alphabet had been invented, tried and tested, formal logical semantics had been inaugurated by Frege, and systematic semiotics had been developed by Peirce.

In the first quarter of the 20th century, both structural linguistics (de Saussure) and functional grammar (Jespersen) were introduced, and in the next 10 years both structural linguistics (Bloomfield, Hjelmslev) and much of functional linguistics (Malinowski, the Prague School, Firthian linguistics) had been outlined, with distinctive theory – the basis of all modern feature-based grammars - following shortly after.

Formal syntax and its relation to formal semantics had also been well-defined since the first three decades or so of the 20th century by Russell, Whitehead,

Wittgenstein, Carnap and the Polish school of logic. Pragmatics was developed further by Morris, the later Wittgenstein, and the Oxford school of Ordinary Language Philosophy, leading to Speech Act Theory and stimulating the development of theories of dialogue and conversation.

Linguistic argumentation had also developed into more than cataloguing and systematisation of facts and generalisations. Comparison of alternative theories of phonemics, morphemics and prosody was widely practised, under the influence of the logical atomism of the early 20th century, and the logical empiricism (Carnap) and critical rationalism (Popper) of the 1930s and after.

These achievements we now take for granted. With hindsight, this development was a movement towards two central goals of modern theoretical linguistics:

1. Precision: the formulation of descriptions of languages in terms of logically and mathematically well-defined structures and procedures, rather than in informal textual definitions and intuitively constructed - though systematic - visualisations with trees, boxes and other delightful shapes.
2. Argumentation: the comparison of alternative theories and the development of criteria of empirical and formal quality for evaluating theories and selecting the best, or the better.

6. Chomsky's role in formalisation

It is not an accident that the goals of precision and argumentation have been even more intensively pursued since the middle of the 20th century, hand-in-hand with the development of the computing techniques which are almost completely ubiquitous today. Not that there is any direct causality involved; in fact many linguists are at pains to stress the independence of the development of the theory of grammar (and the theory of language) from computing.

Chomsky's initial theorising in the 1950s was concerned with finding criteria for formalising and evaluating different theories of grammar; for this purpose he used a certain kind of algebra and formulated a hierarchy of complexity of languages and their associated grammars, now known as the Chomsky Hierarchy of Formal Languages, or simply the Formal Language Hierarchy. It turns out that

natural languages are located around the middle of this hierarchy.

The formal criteria Chomsky discovered were soon found to be equally useful for defining the grammars of programming languages. Every computer science student now learns about the Chomsky Hierarchy in introductory courses on the theory of computation. This formalisation technique lends itself well to operationalisation with well-known algorithms for top-down and bottom-up parsing and generation of words and sentences (in fact of any structures which can be represented by strings of symbols). It is this technology, and modern developments based on this technology, which underlies software such as the stylesheets, spell checkers and grammar checkers of word-processors, the automatic generation of pages on the world-wide web from databases (by search engines such as Google and e-commerce sites such as Amazon).

7. What has computation brought into linguistics?

Perhaps the most significant direct and indirect contribution of computation to linguistics, as to other sciences, is the provision of tested formal intellectual and practical tools for dealing with the complexity of our domain, language, first in terms of intricacy of structure, second in terms of difficulty of processing, and third in terms of the sheer size of the problem. The more mechanical aspects of linguistic expertise, however complex or simple, such as testing a hypothesis about grammatical, morphological or phonological rules, can be replaced by automatic theorem checking strategies based on parsing and generation procedures which are in turn based on well-understood data structures and algorithms, and can then be tested on very large corpora of language data of millions of words, or on many hours of recorded, transcribed and annotated speech. This is now normal procedure in computational linguistics and its applications in Natural Language Processing, as well as in Speech Technology.

But computational linguistics and related disciplines have also brought along a wide range of useful language and speech oriented tools, some widely known and others less familiar, which are useful both inside linguistics and outside it:

- database management systems (DBMS), which are particularly useful for managing dictionaries and corpora both in lexicography and in language and speech technology systems;
- acoustic display, editing and analysis software for phonetic studies;

- parsing and generation software for constructing and testing phonological, morphological and syntactic descriptions;
- speech synthesis and speech recognition software, including dictation software;
- word-processing software, for document design and construction, with monitoring of spelling, grammar and vocabulary.

Word-processing software and website generation software are particularly well-known cases of applied text linguistics. Documents are defined in terms of the text-linguistic *document objects* such as characters, paragraphs, tables, pages and so on which the documents consist of. Possible combinations of document objects are defined by a *document grammar*, which is used to control the displaying of objects on the screen, printing, or exporting documents to other formats. Each of these objects is assigned *styles*, i.e. rendering properties, in terms of attributes such as size, shape and colour – just as words, sentences etc. are assigned a phonetic interpretation (except that the attributes of document objects are defined in the visual domain – unless they are fed into a speech synthesiser, of course, or a haptic output device such as a Braille display). The styles for objects in a given document are standardly defined in a stylesheet for the sake of consistence; word-processing software contains a facility for automatically supporting stylesheet definitions.

Unfortunately, even many experienced linguists have not noticed the text-linguistic foundations of word-processing, and prefer to hack *ad hoc* bold and large font character sequences rather than defining consistent stylesheets for "Heading" objects, for instance. This is all the odder because from the point of view of the architecture of grammars, style-sheets correspond exactly in the printing domain to phonetic interpretation in the acoustic domain. And, of course, in order to produce word-processing software, and to localise it (and many other kinds of software) into other languages than the language of origin (most often English) teams of linguists work with the software developers.

8. Theory, model and reality

The terms *framework* (or *approach*), *theory*, *model*, and *reality* are widely used in linguistics, with widely differing meanings. A more homogeneous understanding of these concepts is used in computational linguistics, and derives from logically oriented approaches to scientific methodology.

A *framework* includes a particular understanding of some domain of reality (i.e. language in the present case), including a theory and its variants, and models described by the theory, as well as particular sets of scientific practices for argumentation and empirical observation.

A *model* is a simplified, or idealised, but systematic representation of a selected domain of reality constructed by means of *modelling conventions* which determine what aspects of of the structure of reality are to be represented. A *formal model* is a model formulated in some well-defined way, often with a variety of set theory. The domain typically contains items representing individuals in the world, and relations between them. In principle this is not too different from the concept of an architect's model, a model train, or Claudia Schiffer, in relation on the one hand to the realities they represent, on the other to the theories of building, locomotion or fashion which they interpret. All these models stand for simplified, idealised versions of the real world.

A *theory* is a set of sentences formulating general and factual axioms, from which further sentences or theorems can be inferred. The sentences are defined by means of a *formal syntax*, for which a vocabulary and rules for constructing sentences out of the elements of the vocabulary is defined. syntax. The derived theorems are used as *hypotheses* when mapped into the formal model: if they match the elements and structures defined in the formal model, then they are formally correct. Not only that: since the formal model is also intended to represent a fragment of reality, the hypotheses are also *predictions*, and these predictions can be tested against observations of reality; if the predictions are true, the hypotheses are empirically correct.

A *notation* is not a theory. Linguists, like other scientists, love their notations. Where would Optimality Theory be without its pointing fists? Where would Structural Linguistics be without its upside-down trees? Where would Discourse Representation Theory be without its boxes? Where would logic be without its *p*s and *q*s, arrows, backwards *E*s and upside-down *A*'s? A notation is a set of typographic, graphical and layout conventions for realising the categories of a formal representation, i.e. a formal language and its formal grammar. A given formal language may be expressed in any of a wide range of notations. A notation without a well-defined formal language behind it is often called a *symbolism*. This is not to say that a given notation for a theory might not be more heuristically useful than another:

1. A tree notation is perceptually much easier to work with than a bracket notation for a parsed word, sentence or text; on the other hand, a bracket notation is easier to write on a keyboard than a tree notation. Formally speaking, they are mutually translatable.
2. The exotic shapes (glyphs) which realise the character and diacritic categories of the IPA are highly distinctive. Glyphs are like phones or allophones in the visual modality: mapping from character and diacritic categories to glyphs, i.e. a *font*, is a visual analogy to phonetic interpretation in the acoustic modality. There are many mutually incompatible IPA fonts, i.e. visual interpretations of the IPA, making typing and computing with these glyphs abominably frustrating. This is why in the 1980s European speech engineers, under the guidance of John Wells, developed the keyboard-friendly and SAMPA alphabet for representing and computing with the same categories, and why the Unicode Consortium has developed standard – albeit still incomplete - glyph interpretations of characters.
3. Optimality Theory uses a tabular theorem resolution method, well-known in Artificial Intelligence, in order to traverse a weighted search space of hypotheses, and thereby derive theorems. Search spaces are often represented as trees to be traversed by an inference mechanism: seen from this point of view, the pointing fists denote the end-points (leaves) in search trees where successful hypotheses are located. Fists and tables are apparently more appealing or helpful than labelled trees, though in terms of inference steps, the trees are more explicit.

There are three interesting varieties of formal *semantics*: An interpretation of a theory by a model is known as *denotational semantics*. The derivation of theorems from axioms and other theorems by purely logical means, i.e. by deductive, inductive or abductive rules of inference, is known as a *procedural semantics* for the theory. A definition of a machine which will perform inference automatically is known as an *operational semantics* for the theory.

Summarising, the metatheoretical terminology used here can be structured into three main methodological levels:

1. Macro-methodology (the intellectual and social professional working environment of a linguist): approach, framework, paradigm.
2. Meso-methodology (the products of linguistic activity): theory, notation; model, interpretation; procedural, denotational and operational semantics..
3. Micro-methodology (the details of everyday linguistics): fact, generalisation, rule, rule application, rule ordering, algorithm, constraint, constraint resolution, inference.

These are not standard terms, but are introduced here for expository purposes.

9. Explication: clarification, symbolisation and formalisation

A central concept in the development of a linguistic theory and its interpretations is *explication*. The following variety of explication is derived from Carnap's logical empiricist approach of the 1930s, and has the following phases:

1. Informal, intuitive domain specification and modelling conventions
   a. *clarification*: what the theory is all about,
   b. *delimitation*: what the theory is not about.
2. Theory formation as explication (increase in precision and both detail and generality):
   a. *systematisation*: traditional linguistic description,
   b. *symbolisation*: notations, visualisation,
   c. *formalisation* (creation of an axiomatic system):
      - *symbolic* (e.g. logic, algebra), with denotational, procedural and operational semantics (i.e. interpretation of the theory in terms of a model),
      - *numerical* (e.g. statistics; probability theory).
3. Evaluation (derived from the methodological approach of Karl Popper)
   a. *derivation* of theorems and interpretation of these as hypotheses, i.e. statements about the real world,
   b. empirical *validation* (falsification, confirmation) of these hypotheses,
   c. *revision* of the theory.

In the framework used here, a theory is a set of consistently formed sentences, interpreted as a description and explanation of something in the real world by means of a model. A model is an idealised, simplified representation of the structures of the real world. If the theory describes everything in the model, the theory is *complete*; if it only describes what is in the model, and nothing outside the *model*, the theory is *sound*. Clearly there will always be aspects of the real world which are outside a given theory, consequently all theories are unsound in this sense; if they were not, scientists would quickly be out of work. It is essential, therefore, to distinguish between the *modelling conventions* which define the model which is relevant for interpreting a given theory, and other aspects of reality which the theory does not claim to cover. The falsification (or confirmation) of a theory occurs when the model in terms of which sentences of a theory are interpreted does not match observations of reality.

10. An example: formal grammars as theories

A modelling convention used in many areas of linguistics is to use *characters* to represent *graphemes* or *phonemes*, and to concatenate these into finite formal *strings*, i.e. sequences of characters, to represent *words* or *sentences*. Strings are concatenated into more complex strings by appropriate rules, including recursive rules which permit the construction of infinite sets of finite strings, representing more and more complex structures. The theory which describes how characters and strings are combined is known as a *formal grammar*. Various kinds of formal grammar are used in linguistics, the main kinds being logical grammars, categorial grammars and rewriting grammars. The last of these, known informally as Phrase Structure Grammars (PSGs) or Constituent Structure Grammars (CSGs) will be outlined here.

A formal grammar G is a quadruple $<V_N, V_T, \Sigma, P>$, where

V_T is a finite set of terminal symbols (terminal vocabulary).

V_N is a set of auxiliary symbols (nonterminal vocabulary).

Σ is a distinguished symbol, $\Sigma \in V_N$ (start symbol)

P is a finite set of rules of the form $\alpha \rightarrow \beta$, for $\alpha, \beta \in (V_T \cup V_N)^*$

In terms of the modelling conventions, the terminal vocabulary is interpreted as the units of language, such as morphemes, which the model contains. The nonterminal vocabulary is interpreted as combinations of these, such as phrases. The start symbol models the largest unit in the model, e.g. the sentence; and the rule set provides a procedural semantics for deriving the terminal strings from the start symbol via its components. So, if the terminal vocabulary is {*this*, *that*, *beer*, *tastes*, *nice*, *awful*}, the nonterminal vocabulary (a rather archaic one) is {*S*, *NP*, *DET*, *N*, *VP*, *V*, *AP*}, the start symbol is *S*, and the rule set is {$S \rightarrow NP\ VP$, $VP \rightarrow V\ AP$, $NP \rightarrow DET\ N$, $DET \rightarrow this$, $DET \rightarrow that$, $N \rightarrow beer$, $V \rightarrow tastes$, $AP \rightarrow nice$, $AP \rightarrow awful$}, then the following formal language can be defined, omitting details of the procedural semantics: {*this beer tastes nice*, *that beer tastes nice*, *this beer tastes awful*, *that beer tastes awful*}. In this case the formal language is finite and very small.

The sentences in the formal language are interpreted by means of a model which contains sentences of real English. If the theory describes all the sentences in the model, it is complete, relative to the model; if it describes only the sentences in the model, it is sound, relative to the model. However, a quick look at the model shows that it is far too simple, consequently it must be extended. Extension of the model makes the theory incomplete when interpreted by the extended model, and so the theory also has to be revised. On the other hand, if the theory contains a sentence which is not interpreted by the model (not shown in the present example), such as *awful tastes beer that*, then the theory is unsound, and also has to be revised.

It is not so well known in linguistics that one of the major achievements of Chomsky was not simply to introduce a standard set of modelling conventions theories of the forms of language, but also to define a set of algebras for formal grammars, in his doctoral thesis 1955, outlined in his *Syntactic Structures* in 1957, which have become a reference standard in theoretical computer science. The "Chomsky Hierarchy" defines the complexity of four types of formal language and their grammars, from Type 0 (unrestricted) through Type 1 (context-sensitive), Type 2 (context-free), to Type 3 (regular or linear). The PSGs which linguists are familiar with, and whose derivations may be represented by trees, are Type 2; these grammars are also standardly used in compiler construction in Computer Science, where they are known as BNF (Backus Naur Form or Backus Normal Form) grammars.

For the tougher readers, here are the definitions of the formal grammar types

in the Chomsky Hierarchy:

For $\alpha \rightarrow \beta$ and $V = V_T \cup V_N$:

Type 0:	$\alpha \in V^+$, $\beta \in V^*$	unrestricted, i.e. the left-hand side of the rule consists of at least one symbol from the vocabulary, and the right hand side may consist of any or no symbols
Type 1:	$\alpha \in V^* V_N V^*$, if $\alpha = \Sigma$, $\beta \in V^*$ else $\beta \in V^+$	context-sensitive, i.e. the left hand side of the rule consists of a non-terminal symbol, and any or no symbols of either kind on its left and on its right, and the right hand side may consist of at least one symbol, but if the left hand side is Σ, then the right hand side may also be empty
Type 2:	$\alpha \in V_N$ and $\beta \in V^*$	context-free, i.e. the left hand side consists of just one non-terminal symbol, and the right hand side consists of any combination of symbols
	$\alpha \in V_N$, $\beta \in V_T \| \beta \in V_N V_N$	*Chomsky Normal Form*, a special case often used in linguistics, in which the right hand sides do not mix non-terminal and terminal symbols
	$\alpha \in V_N$, $\beta \in V_T V_N^*$	*Greibach Normal Form*, a special case often used in compiler construction for efficient parsing
	$\alpha \in V_N$ and $\beta \in V_T V_N V_T$	metalinear, a special case often used in morphology for single-stem items with arbitrary numbers of prefixes and suffixes
Type 3:	$\alpha \in V_N$ and $\beta \in V_T V_N$	right-regular, right-linear, i.e. the left hand side consists of just a non-terminal symbol and the right hand side consists of one terminal symbol followed by one non-terminal symbol
or:	$\alpha \in V_N$ and $\beta \in V_N V_T$	left-regular, left-linear, i.e. like the right-regular case, but with the non-terminal symbol followed by the terminal symbol

These languages are in a relation of implication or inclusion to each other: Type 3 languages are subsets (special cases) of Type 2, Type 2 languages are subsets of Type 1, Type 1 languages are subsets of Type 0. From a linguistic point of view, it

is useful to recall the points about formal languages and grammars which are summarised in Table 1.

Types	Description	Linguistic Application	Computation
Type 0:	Unrestricted Languages & Grammars	Formalisation of transformational grammars; too unrestricted for theories of natural language syntax but heuristically useful	Turing Machine
Type 1:	Context-Sensitive Languages & Grammars, with the subset of Index Languages & Grammars	Formalisation of the most complex structures found in natural language syntax, i.e. cross-serial dependencies such as *X, Y and Z married A, B and C, respectively*	Indexed Automata
Type 2:	Context-Free Languages & Grammars, including the subset of Metalinear Languages & Grammars	Formalisation of basic hierarchical structures of natural language syntax (i.e. Phrase/Constituent Structure Grammars); metalinear languages are useful for modelling inflection and derivation (single-root complex words)	Push-Down Automata
Type 3:	Regular / Linear Languages & Grammars	Formalisation of basic linear (pure right-branching / pure left-branching) sentence structures, morphological, phonological and prosodic structures	Finite State Automata, Finite State Transducers

Table 1: Summary of formal languages and grammars and their main applications.

For linguists, it is also important to know that terminology such as "context-sensitive" and "context-free" refers to *hierarchical contexts*: if the left-hand side of a rule such as $S \rightarrow NP\ VP$ contains only one symbol it is context-free; if it contains more, it is context-sensitive. Where *linear contexts* are involved, as in phonological rules, fondly referred to by linguists as "context-sensitive rules", these rules are in fact not at all context-sensitive when modelled by formal languages, since they do

not refer to hierarchical contexts. In fact they are even more restricted than the Type 2 context-free case, and can be shown to belong to the Type 3 or Regular Grammars. The terminology is different, and misleading. A clear example would be a phonotactic rule such as $C \rightarrow$ s / #_ *CC*, which stipulates that the first consonant in a cluster of length 3 in English is /s/. This rule is straightforward to formulate in terms of Type 3 rules and to implement with a Finite State Automaton (FSA); context-sensitive rules in the formal language sense would be unnecessary overkill. Phonological rules which map underlying to surface representations are basically no different from phonotactic rules, except that they incorporate an additional mapping or translation relation and are implemented with a Finite State Transducer (FST). In this case, sequences of phonological rules can actually be collapsed into one complex rule: it is well known that FST cascades, which implement such sequences, can be composed into a single FST. The same applies, as Lauri Karttunen has shown, to ranked constraints in Optimality Theory which are, despite claims to the contrary, not unlike derivational rules. Another point to remember is that so-called "cyclical rules" are, in general, cyclical applications of simpler rules to ever larger domains, e.g. in the prosodic interpretation of compound words or of phrasal constructions.

A more detailed explanation would go beyond the frame of reference of this essay, but can be found in any introduction to formal linguistics or to theoretical computer science.

Envoi (in lieu of references)

This essay was originally conceived as a general lecture, not as a fully fledged article. References to the literature are sparse, and inexplicit. In the present context this necessity may easily be turned into a virtue: the reader, if so inclined, may regard these references and hints as a challenge for extending the horizons of his own knowledge and develop the references and hints into a complete documentation. This detective work is made rather easy, these days, by internet search, if used with care. And where would the internet – hypertext, therefore text, therefore the concern of linguistics – be without formal linguistics? A genuine research question for document theory and text technology, for example, is the following: are the XML and HTML documents used to formalise web pages Type 1 or Type 2 languages? Well, try to find out - may the debate continue!