*Foreword:* The objective of my research is to prove that all languages are *constructed*. In this respect, languages can be either socially constructed (especially natural languages) or non-socially constructed ("artificial"). To give support to such a view, the goal of this paper is to demonstrate that, when "possible languages" are limited to meaningful and fully complex ones, the space of possible grammars is so strictly restricted that all languages can be traced down to a common structure.

This means on the one hand, that when the grammar of any language is gradually deconstructed by reducing complexity but maintaining the semantic expressiveness, all deconstructions end with the same terminal point. On the other hand, this point necessarily forms the foundation of all languages, and is found as an element in all grammars. As a construction process, it is possible to construct an immense number of grammars, but human constraints and the size of a finite dictionary do set a loose upper limit to possible language complexity.

While it is theoretically impossible to prove that there are no languages which are not based on the aforementioned common structure, this claim is taken as true until falsified; the burden of proof in this case is on the opposing side. The falsifying process is however simple: if anyone presents a meaningful and fully complex language which is not based on the common structure (most practically: does not have nouns and dependency relations), this theory is considered false.

A PhD in syntactic theory is scheduled to be finished in 2020.

# The inevitable reason why all languages are similar

It is an old observation that all languages, despite apparent differences on surface level, seem to be significantly similar in their core structure. It was hypothesised by the 17th century Port-Royal philosophers Lancelot and Arnauld that all languages are governed by the same universal rules, being based on the one and only logic available for mankind. The rules of this *universal grammar*, applicable for all languages, would need to be reconstructed in order to discover the essence of language. (Jermołowicz 2003.)

Although the philosophy of Port-Royal enjoyed great interest until the beginning of the nineteenth century, attempts to reconstruct such a grammar ultimately failed. The quest for a universal grammar was eventually revived by Noam Chomsky, although with an important departure from the Port-Royal Grammar: according to Chomsky, the universal syntactic principles are genetically encoded in humans. (Jermołowicz 2003, Chomsky 1972, Christiansen & Chater 2008.)

What is more, in *Syntactic Structures,* Chomsky (1957) introduced Emil Post's mathematical production system to linguistics as rewrite rules (Pullum 2010, cf. Post 1943). With this novel method it may have been anticipated that Universal Grammar (UG) would be reconstructed as a formal grammar. Quite conversely, Chomsky's UG seems to have remained on the level of an abstraction (cf. Chomsky 1986), and in the more recent years there have been a growing number of voices suggesting that the UG project has failed (e.g. Dąbrowska 2015, Evans & Levinson 2009). The closest thing to establishing the rules of UG was perhaps the Principles and Parameters framework (Chomsky & Lasnik 1993). This approach suffered from counter-evidential assumptions concerning the number and quality of linguistic universals which led Chomsky to move on to a simplified theory known as the Minimalist Program (MP; Chomsky 1995, Dąbrowska 2015).

In this paper we revisit the idea that syntactic similarity between languages depends on logical necessity, rather than an innate mechanism, and provide evidence for our hypothesis in a concrete, falsifiable format. In order to do so we use standard methods of formal language theory with the extension of mathematical graphs. Evidence from natural languages will also be discussed.

## 1. The shape of the space of possible grammars

To avoid confusion of terminology, we turn the focus to a "fundamental grammar": a formalisation which functions as the basis of all possible grammars. Looking at a most basic production, the simplest possible grammar consists of the rule $S \rightarrow a$ only where $a$ denotes a *symbol*; a word in more common terminology. We can state that this is the most fundamental grammar in the sense that all grammars include the rule $S \rightarrow a$ in one form or another. In practice it means that all possible grammars generate symbolic languages, or are based on using at least one word. This most fundamental grammar is extremely limited in expression as it has only one symbol, and each utterance can only consist of this one symbol.

Adding recursion, we witness the appearance of the regular grammar $S \rightarrow aS \mid \lambda$, which allows us to produce utterances of any desired length; and to make use of a dictionary, we construct the grammar $S \rightarrow AS \mid \lambda$ where $A$ denotes a semantic category − some word class − and normally adds its own production rule $A \rightarrow \ldots$ , followed by a lexicon in the form of a list of terminals.

We may choose to call each of the three grammars fundamental because all natural languages, as well as all programming languages, build on the third grammar, which builds on the second grammar, which builds on the first grammar. Together the fundamental grammars introduce the two or three building blocks which function as the make up of all grammars: terminals (denoted with a minuscule letter), nonterminals (denoted with a capital letter) and the mechanism of recursion (i.e. a terminal or a nonterminal on the left-hand side repeated on the right-hand side of the production). Adding such building blocks we can eventually reach the complexity of programming languages, followed by the complexity of natural languages.

Looking at the simplest grammars we can observe some important matters of complexity. The simplest grammar, $S \rightarrow a$ has only one production rule, while the simplest recursive grammar, $S \rightarrow aS \mid \lambda$, has two, the vertical bar representing disjunction. The recursive element $S$ on the right-hand side must be assessed in order for the parsing process to terminate wherefore the addition of an empty symbol (a lambda rule) is required[1]. The simplest grammar which may include a word list, $S \rightarrow AS \mid \lambda$, additionally requires a command line for its lexicon. We will count this as having two syntactic production rules proper plus a word list of indefinite length.

We can now asses the three fundamental grammars as having the complexity of one production rule, two production rules, and two production rules plus one word list, respectively. There is no logical limit to the complexity of a grammar. As combinations of grammatical rules of indefinite length are logically possible, the number of possible grammars is also indefinite.

[1] Or alternatively one or more free terminals, as is done with phrase structure grammars; for this grammar, the production rule $S \rightarrow a$ replaces the lambda.

In generative grammar framework such an observation has had a significant impact on language acquisition theory. As formal language theory shows that the space of possibilities is infinite, it would seem that such space needs to be constrained in order to make language acquisition possible. In the framework of the Poverty of the Stimulus theory, UG is a device which limits the space of possible grammars so that a child only needs to confront humanly learnable languages. (see de Landa 2011.)

To picture this idea we can imagine an unbounded space of languages. In this space, natural languages occupy a tiny confined area at a random location, almost like our own planet in an ever-expanding universe (figure 1).

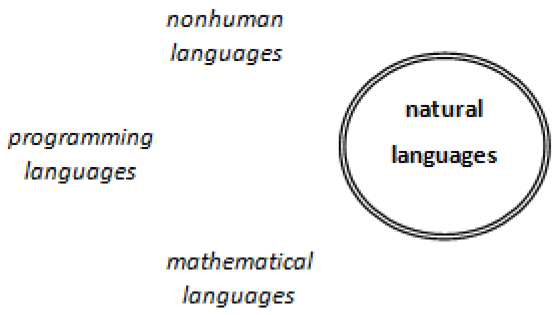


**Figure 1**: an unbounded space of possible languages where natural languages occupy an area, confined by UG, at a random location.

This is however a misconception. When we study the shape of the space of possibilities, it becomes apparent that, although the space is indeed virtually infinite, it is far from being unbounded. The simplest and most fundamental grammar, $S \rightarrow a$, has the complexity of one rule which includes just one element besides the obligatory start symbol. The space of possibilities cannot include any grammar below this complexity, which means that $S \rightarrow a$ is positioned at an absolute low end point of the space. What is more, there is only one possible grammar with the the complexity of just one element besides the obligatory start symbol. This means we are looking at the very tip of the space of possibilities.

As complexity increases, the space widens gradually. With the complexity of two elements besides the start symbol we find the possible grammars $S \rightarrow aa$ and $S \rightarrow ab$, that is, exactly two grammars[2]. With three such elements we find the grammars $S \rightarrow aaa$, $S \rightarrow aab$, $S \rightarrow abb$, $S \rightarrow abc$ as well as $S \rightarrow a \mid b$[3]; five grammars in all. Allowing for four elements we witness the appearance of the first recursive grammar, $S \rightarrow aS \mid \lambda$, alongside an increased number of nonrecursive ones.

This observation gives us a slightly more informed view of the space of possible grammars. It is not unbounded, but an expanding pyramid shape, strictly limited at three sides while open at the upper end. Towards the tip there are extremely simple grammars which function as the building blocks of more complex languages, such as mathematical and programming languages. Moving

[2] The difference between $S \rightarrow ab$ and $S \rightarrow ba$ is trivial because here $a$ can be understood as denoting "the first symbol" and $b$ as "the second symbol".
[3] where disjunction counts as one element

upwards on the complexity scale, natural languages follow. Adding to their complexity, we can imagine an infinite number of languages which are excessively complex for human needs.

This still a rather superficial examination of the space of possible grammars does not provide enough information to allow one to place human languages in a horizontally specific location. We can however make the assumption that vertically they typically occupy a space between programming languages and imaginary languages of excessive complexity. With this perspective in mind, the postulation of a Universal Grammar to limit the space of possible grammars may not seem motivated because the space is already limited by logical constraints (see figure 2).

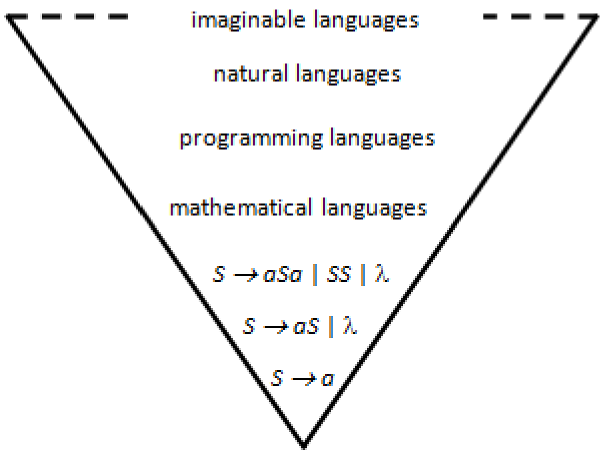


**Figure 2**: a more informed view of the space of possible languages as a pyramid shape where natural languages are placed, according to grammatical complexity, between programming languages and excessively complex yet theoretically possible languages. Minimal grammars which function as a foundation for the more complex grammars are found towards the tip of the space. Such grammars named here are (bottom-up:) i) the simplest finite-choice grammar; ii) the simplest recursive grammar; and iii) the simplest nonlinear grammar.

To be as exact as possible, natural languages are strictly confined within a space which spands vertically between the complexity of (i) the simplest natural language, and (ii) the most complex natural language. This enclosure is likewise horizontally restricted by the number of possible combinations allowed by the number of elements in the two extreme grammars. This limits the space of human languages into a quadrilateral area with no open edges left (figure 3).

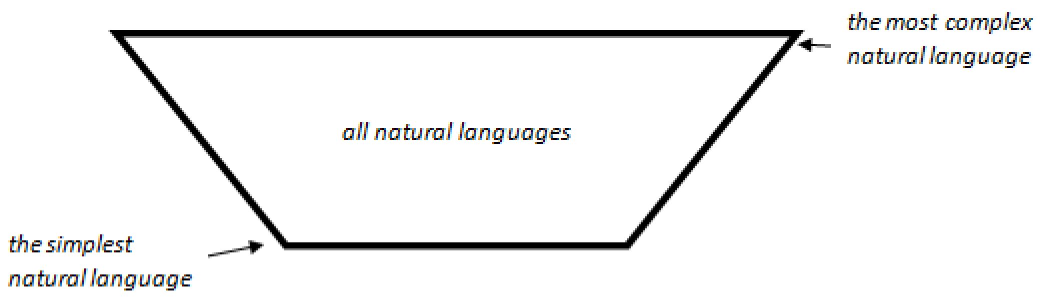

the most complex natural language
all natural languages
the simplest natural language

**Figure 3**: natural languages in a closed area within the space of possibilities.

Given the complexity of natural languages the number of possible combinations is likely to be very large; on the other hand, given that each different grammar is a mere matter of organising the building blocks (terminals and nonterminals) in different arrangements, whichever grammars are included in this area can only be variations of one another.

It would seem natural that only grammars that introduce a new paradigm to the production system have a special place in the space of possibilities. Therefore, from the perspective of formal language theory, each natural language, such as English or Chinese, each programming language, such as Pascal or APL, and each mathematical language, such as the syntax of arithmetics or first order logic, is just another formally − yet by no means culturally − axiomatic combination of the two basic elements.

## 2. Simplicity in human language

In the previous chapter we presented a series of fundamental grammars which function as the basis of all languages. They are however very elementary, far from the complexity of human languages. What may seem more interesting is to find the closest common denominator of all natural languages. We will attempt to construct such a grammar based on what is known about structural simplicity in the languages of the world. The logic is straightforward: a construction which only includes features shared by all natural languages is considered fundamental. With this criterion in mind we are going to look for the nearest diverging point where a shared line of fundamental grammars branches off into two or more different types of languages.

The question is: *how simple can a language be while conveying all the meaning of human verbal communication?* To illustrate, one of the simplest systems one might imagine is something we could call a *laconic police report*; a system where each required piece of information is covered with one item, as in the following example:

crime: theft
perpetrator: man
object: bicycle
victim: Mary
time: yesterday
place: market

These expressions tell us that a man stole a bicycle from Mary at the market yesterday. While the objective of such a report is to be completely unambiguous, dependencies tend to become vague in linear form. For instance, we might want to add a further piece of information:

crime: theft
perpetrator: man
**quality: grand**
object: bicycle
…

With the added information the report has three possible interpretations: (a) *a grand man* stole a bicyle, (b) a man stole *a grand bicycle*, or: (c) the crime is a matter of *grand theft*. Such ambiguity may be undesireable.

With an automated report template one can add item-specific information into a desired place. This method builds a hierarchical tree (figure 4).

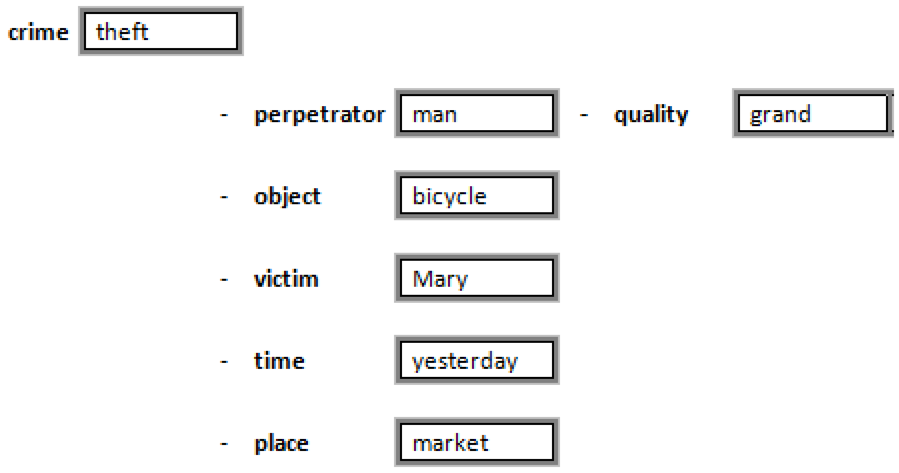


**Figure 4**: a "laconic police report" in dependency form.

Being completed correctly, the tree format expresses unambiguously that the the crime is *theft* while the perpetrator is a *grand man*.

The laconic police report is an example of a syntactically minimal language. In addition to its straightforward dependency structure, our example sentence is noun-heavy with ten English nouns, one adjective (*grand*) and one adverb (*yesterday*). Before finding a formalisation for this grammar, we will in the next chapter discuss the minimal number of semantic categories required in order for human language to be functional.

## 2.1 Categorial flexibility in natural languages

Especially since the 1970s research into the languages of the world has suggested that certain languages contain categorial flexibility to the degree of having just one flexible word class to correspond to the familiar European classes of verbs, nouns and adjectives. Claims for such languages include a number of Malayo-Polynesian, Salishan, Wakashan, Munda, Turkic and Australian languages. Despite evidence from field research, the subject is controversial in syntax theory. (Lier & Rijkhoff 2013.)

We do not consider the controversy as an obstacle for this paper because, rather than making claims concerning a particular language, our focus is in looking for the basic elements of all languages. For this purpose it seems safe enough to conclude that all known languages have at least content words and function words, although a monocategorial system may also be considered as a foundation. However, we will additionally discuss the consequences of rejecting flexibility between nouns and verbs (2.4).

A second factor that speaks in favour of accepting categorial flexibility is in formal logic where expressing members of the above three parts of speech as a class of *predicates* has a long tradition (cf. Jaszczolt 2002). Therefore it can be considered decided that in any case a functional grammar may have a flexible word class for what are usually called verbs, nouns and adjectives.

To be more precise, a flexible word class could also include a number of other traditional classes. For instance, a separate category for numerals was previously thought to be an absolute universal (Greenberg 1978). Counterexamples have since been found (cf. Everett 2005; Gordon 2004). Several languages of New Guinea do without numerals, using the names of body parts for counting (Comrie 2013). In addition, noticeable interaction between nouns and numerals has been reported in a number of European languages (Hurford 2003).

A second example of a category which may be incorporated into nouns is the class of pronouns. Some Southeast Asian languages lack clear personal pronouns, using titles instead[4], while many languages lack third-person pronouns (Cysouw 2001). Sign languages like ASL (American Sign Language) also lack pronouns, using pointing instead (Evans & Levinson 2009). As hinted by their Latin names, nouns and pronouns are two semantically related classes. The most striking difference is that while the former is an open class, the latter is in most languages a closed one. Japanese pronouns have however been described as an open class (Sugamoto 1989).

A third case of a potentially superfluous class are lexical adverbs. These can often be expressed as adverbials, such as "*in a quick manner*" for 'quickly'. Furthermore, colloquial English often makes no difference between adjectives and adverbials of manner, e.g. "*You better come **quick***" for "*You had better come quick**ly***", while this is standard for some words, e.g. "*You must work **hard***." This is not exceptional among the languages of the world, e.g. Dutch (Stern 1984).

In our search for a precise fundamental basis for all natural language grammars, it is consequentially important that such a grammar does not make use of a separate class of adverbs, pronouns or numerals, or any other lexical category that is not included in every language. By this we do not imply that there is a logical problem with any of the classes; what we mean to say is that, from a minimalistic[5] point of view, the more specialised categories can be regarded as subsets of the more general category of content words rather than a basic semantic element.

## 2.3 Defining a functional category

In 2.2 we established the basis of a flexible content word category for a bicategorial candidate for a human fundamental grammar. What remains is to define the type of words which should make up the required functional category.

There are different types of function words to choose from. For instance, articles are a very frequent subclass of non-content words in English and many other languages, while nearly a third of languages featured in WALS (World Atlas of Language Structures; Dryer 2013a) have neither a definite nor an indefinite article. A number of languages on the other hand use demonstratives as definite articles, or the numeral 'one' as an indefinite article. An example of flexibility between possibilities is Finnish where demonstratives have been proposed as an article due to relatively frequent use especially in colloquial speech despite the language being traditionally considered as

[4] of the kind "honorable sir".

[5] We use the word "minimalistic" as a reference to its dictionary definition, not to Chomsky's Minimalist Program.

having no grammatical articles. It has however been suggested more recently that such practice is the matter of using demonstratives as an optional determiner (Larjavaara 2001).

As we have already defined a means of expressing determiners as content words, a class of grammatical articles seems superfluous for our purposes. There is a more potential candidate for a general functional class, one which consists of relation marking particles that together with content words can make up adverbials of different types, as in our example "***in** a quick manner*".

An important function of such particles is covered by conjunctions. In many languages the division between adpositions and conjunctions is not clear-cut, with many adpositions regularly surfacing as conjunctions (Schmidtke-Bode 2009). English *and, plus, but, like, than, to, except, after, before, until* and *since* serve both functions with the same semantics (conjoining, comparing, excluding etc.), depending on the construction they appear in. Placed before a noun phrase, they are normally analysed as prepositions, but placed before a clause they are normally analysed as conjunctions. As Huddleston (1984) has suggested, prepositions and conjunctions could in theory be viewed as one word class.

Although a separate class of conjunctions may seem elementary from a western point of view, especially with logical operators in mind, typological research shows that it is not quite the case. For example, only 61 languages (ca. 27 %) in a sample of 225 use adverbial conjunctions as primary gestalt features in purpose clauses, while adverbial verb suffixes are nearly as common. Other strategies include case affixes, adpositions and other types of particles among others. (Schmidtke-Bode 2009.)

While the need to mark sub clauses as expressing purpose is a relatively strong tendency, it seems that language structures allow for several different ways to do so, and these are most typically the combination of a content word and a particle which marks the role of the content word in the sentence. This is the structure we will build our candidate grammar on.

Based on what is known of categorial flexibility in natural languages, we choose two word classes for our candidate grammar. These are (i) a flexible class of content words, and (ii) a class of function words which label content words according to their role in the sentence.

| nouns | nouns | nouns and N-pronouns | content words | content words + role markers |
|---|---|---|---|---|
| numerals | | | | |
| pronouns | D-pronouns | | | |
| articles | | | | |
| verbs | verbs | | | |
| adjectives | | | | |
| adpositions | role markers | | | |
| conjunctions | | | | |
| adverbs | | | | |

**Table 1**: starting with nine possible lexical categories (leftmost column), a step-by-step fusion into a bicategorial system (rightmost column) is carried out.

Our strategy is to start from a very basic grammar and to build up to what is the closest common denominator of natural languages. In order to produce testable sentences with the grammar, we will use a standard English dictionary[6]; it should however be noted, that the resulting language is not

[6] For instance http://www.oxforddictionaries.com

English despite the surface-level resemblance. For convenience we will call the grammar FG2, marking initials from "Fundamental Grammar", and the digit for the number of semantic categories.

Quite conveniently, the simplest possible nonlinear grammar − one which generates a conventional parse tree − is a binary one. We will use the grammar $S \rightarrow aSb \mid SS \mid \lambda$ which produces strings of a's and b's, starting with an *a* and ending with a *b*, and having an equal number of each (cf. Sudkamp 1997). This grammar has three producton rules, but we will need to expand it to having the two semantic categories discussed above, each given its due word list. The full grammar consists of three syntactic rules and two word lists:

$S \rightarrow ASB \mid SS \mid \lambda$
$A \rightarrow$ <list of function words>
$B \rightarrow$ <list of content words>

Here we choose to use function words prepositionally. To demonstrate how relations are expressed in this minimalistic language, we will translate the following English example sentence: "*A girl sees a boy on a hill.*" For this we expand the above grammar into the following toy grammar:

$S \rightarrow ASB \mid SS \mid \lambda$
$A \rightarrow$ so | agent | patient | on
$B \rightarrow$ see | girl | boy | hill

A translation of the example sentence into FG2 is "*so agent girl patient on hill boy see*." Using English words here is a matter of convenience, but as there are no nominative and accusative particles available, we use grammatical terms to express these relations. Using a Japanese dictionary we could borrow *ga* and *o* to replace *subject* and *object,* respectively. Whether the resulting FG2 sentence seems grammatical in English or Japanese is however not relevant here. Instead, the example sentence is proven grammatical with the above toy grammar using the CYK algorithm which generates the tree in figure 5.

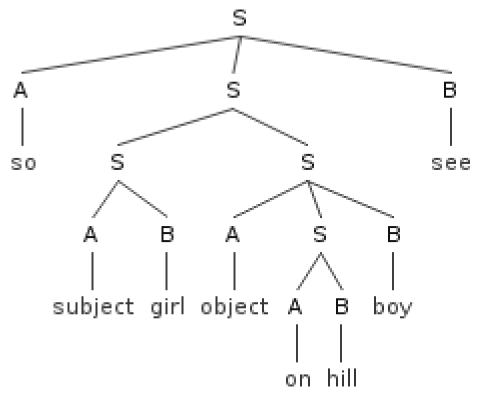


**Figure 5**: a parse tree for an example sentence in the deterministic binary grammar FG2. Examined bottom-up, the locative phrase "*on hill*" is an embedded dependent of the object phrase, while the subject phrase and the complete object phrase are dependents of the labelled verb phrase "*so see*".

All phrases in the binary FG2 grammar are composed of pairs of a function word and a content word. A marker is required for each content word, including the verb. We chose the coordinator 'so' as a neutral topic marker for the main clause. Grammatical articles are left out to simplify the expression although they could be expressed with pronominal determiners (*this, that, some, one* etc.) either in an appositional structure or in direct dependency.

The grammar has extreme syntactic precision. Structural ambiguity is often understood as being a question of whether an adjunct such as *on a hill* refers to the action: a girl [sees [on the hill]]; or to the patient [a boy [on the hill]]. (Bird et al. 2009). Our toy grammar produces a third valid arrangement, "*so agent on hill girl patient boy see*" where "*on hill"* refers to the agent: [agent [on hill] girl]. FG2 is typologically a deterministic context free grammar (DCFG), a type which is much used for programming languages due to its structural unambiguity (cf. Sudkamp 1997).

In our toy grammar we have used English particles for function words when available, and otherwise replaced them with nouns to assign a semantic role. Using such a method, all particles can in fact be replaced (see table 2).

| FUNCTION WORD + | CONTENT WORD ==> | PHRASE MEANING |
|---|---|---|
| destination | school | "to school" |
| manner | clever | "cleverly" |
| quality/adjective | run | "running" |
| polarity | negative | "not" |
| quantity | plural | "more than one"; "*−s*" |
| time | past | "before this moment"; "*−ed*" |
| aspect | completeness | e.g. "drink **up**" for *drink* |
| mood | imperative | e.g. "drink!" for *drink* |

**Table 2**: combining function words and content words into adpositional phrases to form various types of meanings.

As a rule of thumb, we suggest that such marking is considered semantically correct if the resulting phrase translates to a genitive-locative relation: "*destination school*" = "in the destination of school"; "*manner cleverness*" = "in the manner of cleverness", etc.

In our next scheme, both *A*'s and *B*'s of the grammar $S \rightarrow ASB \mid SS \mid \lambda$ are expressed using English nouns, some of which have a functional task while others express semantic content. Using this method our original example sentence can be translated as "*statement subject girl object boy **location** hill see.*" Here the word 'see' is meant to be understood as a noun, as in "*long time no **see***", an English idiom which is thought to borrow a Chinese or Native American structure (Partridge & Beale 2002).

Our conjecture is that all verbs can logically function as nouns. Whether this conjecture is accepted or not may greatly depend on the reader's native langage. English displays a degree of flexibility between nouns and verbs, for example 'throw' which according to the dictionary can be either a verb or a noun as in the idiom "*a stone's **throw** away*". As such, it can describe an event, or it can be understood as a unit of measure. Disambiguation is possible by compounding: a "*throw-event*" vs. a "*throw-unit*". At any rate such compounds, including the possible '*see-event*', are unambiguously nouns.

Another disambiguation method provided by the FG2 grammar is adpositional tagging: "*event throw*" vs. "*unit throw*." What is more, many languages disambiguate with a nominal suffix, cf. English *throw–**ing***. There are many possibilities as of how to use nouns to express actions or events; a specialised category of verbs is not elementary for functional or even unambiguous language.

We hereby come to the formalisation of our "laconic police report" (2). For an unambiguous syntax, the three-rule deterministic grammar $S \rightarrow ASB \mid SS \mid \lambda$ is fit for the task. To translate our example expression (figure 4) into FG2 , words in text boxes are content words (*B*'s), and the prepositions in bold are function words (*A*'s), or role markers. This gives the parse tree in figure 6.

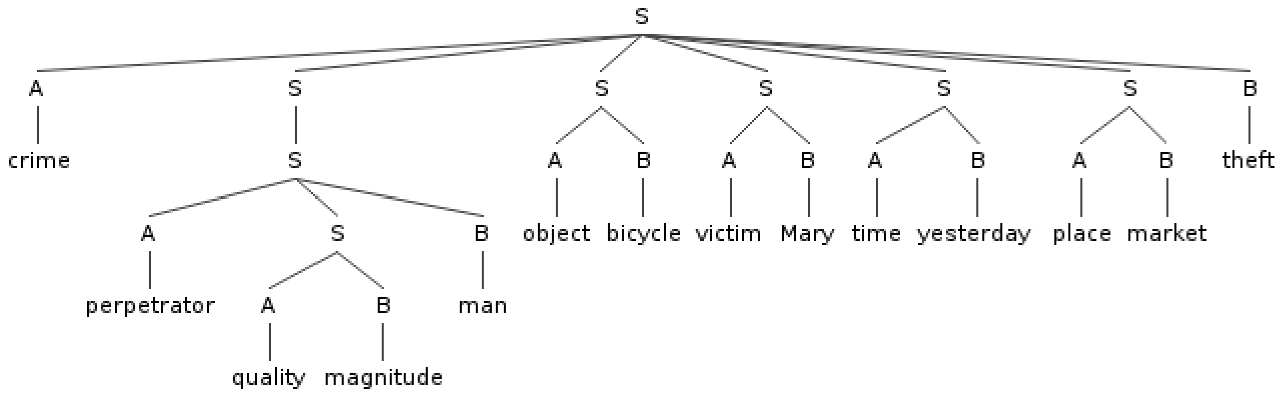


**Figure 6**: an unambiguous parse of the example sentence of *laconic police report* (cf. figure 4) in FG2. The adjective *grand* is replaced with the noun *magnitude*. *Yesterday* is considered a noun, as in the phrase "*I've always had the view that you remember yesterday, work for today, but also work towards tomorrow*"[7].

As the language can also be used with nouns only, the monocategorial grammar $S \rightarrow ASA \mid SS \mid \lambda$, where $A_1$ assigns the semantic role of $A_2$, is likewise plausible. This makes sentences syntactically ambiguous because there is little formal way of telling adpositions and nouns apart in the equivalent sentence "*crime perpetrator quality magnitude man object bicycle victim Mary time yesterday place market theft*."

As such the ambiguous grammar resembles natural languages where the recognition of a semantic structure is a probabilistic process. Unlike machines, people may, despite a lack of formal method, be able to identify preposition-noun pairs such as "time: yesterday", and "place: market", through probabilistic mental processes, and reject unlikely possibilities, such as "yesterday: place".

In other words $S \rightarrow ASA \mid SS \mid \lambda$ appears to share similarities with a prototypical human language, one that may seem to have the consituent order SOV, although it is more precisely a free one. This syntax is sufficient to express all human verbal communication. If such grammar was proven to be the closest common denominator − a type of "universal grammar" − for all natural languages, it would seem to impose significant pressure on nativist theories which would have to defend their view that the $S \rightarrow ASA$ structure is too complex to be learned and therefore must be pre-wired in the child's brain. We will return to this question in the following chapter.

## 2.4. So which one is the actual "universal grammar"?

[7] Example from http://www.oxforddictionaries.com/definition/english/yesterday.

In the previous chapter we presented the unambiguous grammar $S \rightarrow ASB \mid SS \mid \lambda$ and the ambiguous grammar $S \rightarrow ASA \mid SS \mid \lambda$. Despite different formalisations, both can be understood as being based on two semantic categories: content words and role markers. We will use these grammars as a starting point in our search for a formalisation of a grammar which is the actual closest common denominator for all human languages.

Looking at the basic mechanism of the production system, it is possible to establish grammars of "fundamental" or "universal" character with mere reasoning. In contrast, formalising the closest common denominator for all human languages depends on our empirical knowledge. Even though formalising one or more candidate grammars is a simple process, it may be that reaching unanimous consensus will not be possible at this time.

To see what we can put together preliminarily, it may be closest to current knowledge of human languages that having two semantic categories is the minimal requirement of categorial complexity. As discussed earlier, this has been contested: according to many linguists (see Lier & Rijkhoff 2013), all languages must distinguish between verbs and nouns. If the case turns out to be such, we may easily add a further category *C* for verbs, while the current category *B* is reserved for nouns only. In such case we may reformalise our nonlinear grammar as $S \rightarrow ASB \mid ASC \mid SS \mid \lambda$.

It is more compelling that our grammar already contains features which seem too specific to be universal. Two obvious complications are (i) the requirement of an obligatory role marker for each sentence as well as each content word, and (ii) the centre-embedding structure. Centre embedding of prepositional phrases is by no means alien to natural languages[8], but surely all languages do not have it, and in terms of case languages, such structures may even seem difficult to produce.

A solution to both complications is to reject the centre embedding structure, allowing for ambiguity. The grammar $S \rightarrow AS \mid SA \mid BAS \mid SBA \mid SS \mid \lambda$, where *A*s are content words and *B*s are prepositions, allows for the omittance of the conjunction as well as that of the preposition, and replaces centre embedding with either left or right recursion. This may be as close as we will get to the closest common structure for all natural languages as a context free grammar; the grammar is prepositional which likewise is not a universal. Therefore it may not be quite correct to claim it as the basis of all natural languages.

On the other hand, the grammar generates the same language as the equivalent postpositional grammar $S \rightarrow AS \mid SA \mid ABS \mid SAB \mid SS \mid \lambda$. These two grammars may consequently be accepted as representing the same logical foundation, which is that all natural languages have content words and related particles placed either before or after them, unless omitted. Likewise, postpositions may be considered equivalent to suffixes, and prepositions equivalent to prefixes. This way all four types may be understood as being essentially the same.

To falsify the claim that this grammar is the closest common denominator, one would have to find a language which only places particles in infix or circumflex position, or completely lacks role markers. In fact there are a small number of languages which use inpositions as a main type (Dryer 2013b); but if strictly speaking no such natural language exists, our grammar would seem as a human-chosen departure from the mere logic of the production system.

[8] E.g. written Swedish: *Problemet var bara att normal föreningsverksamhet, [enligt [i de nordiska länderna] utvecklad praxis*] (…)" ("The problem was only that common organisational customs [according to [in the Nordic countries] developed policy]; <http://tinyurl.com/z3r78mg).

Taken more literally, one might however choose to reject both the prepositional and the postpositional alternative. Because linguistic structures must occur chronologically as people speak or write their language, decisions concerning word order will necessarily have to be made. All languages do not have both prepositions and postpositions (ibid.), but the only way to avoid having to choose between a prepositional and a postpositional mode is to build structures nonlinearly, as in a tree or a mind map where dependencies are expressed two-dimensionally.

A chronologically linear form forces one to place particles either before or after the main word. This means that a phrase structure grammar (PSG) may never be completely truthful in conveying a universally neutral structure. This dead end may however depend on methodological problems rather than the ultimate impossibility of reconstructing a universal grammar. One thing to consider is the parsing method: the CYK algorithm is well established, but it is not the only option.

One interesting observation about the FG2 grammar $S \rightarrow ASB \mid SS \mid \lambda$ is that, being a binary structure, each $A$ functions similarly to an open bracket while each $B$ functions similarly to a closed bracket in a dependency tree. Using an all-noun word list, it is possible to differentiate between function words and content words by adding an open bracket prefix to each function word, e.g. *[direction*, and a closed bracket suffix to each content word, e.g. *school]*, forming the pair *[direction school]* ("to school") which can be inserted as it is into a general syntax tree generator[9].

With such POS tagging a sentence "*statement subject girl object boy location hill see*" (figure 5) becomes *[statement [subject girl] [object [location hill] boy] see]*. Inserting this sentence directly into a syntax tree generator generates figure 7.

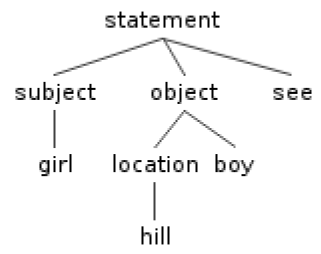


**Figure 7**: an automatically generated dependency tree for a POS tagged FG2 sentence. This structure is also unambiguous, but much simpler than the PSG which consists of eleven added nonterminals (figure 5). Here *hill* is a dependent of *location* which is a dependent of *object*.

This simple parsing method gives the grammar remarkable expressive power, as it can now generate a great variety of different tree shapes, in contrast to the phrase structure bound by the obligatory nonterminals (figure 5). A second interesting feature in the all-noun dependency structure (figure 7) is that each of its branches forms an endocentric compound, from leaf to head:

girl-subject-statement
hill-location-object-statement
boy-object-statement
see-statement

[9] E.g. http://ironcreek.net/phpsyntaxtree/

Each nonlinear dependency tree may be expressed as the sum of its linear branches. The grammar $S \rightarrow AS \mid \lambda$, which generates a linear string of symbols, can be used to express each separate branch. Additionally, when each recurring symbol is considered identical to the one before, the strings are automatically joined into nonlinear form, such as a tree or a graph. In other words we can see here that the linear grammar $S \rightarrow AS \mid \lambda$ has at least all the expressive power of the nonlinear context free grammar $S \rightarrow ASB \mid SS \mid \lambda$. We name this linear monocategorial grammar FG1, as opposed to the nonlinear bicategorial grammar FG2, and look further into alternative parsing methods in the next chapter.

## 3. Graph grammars as a parsing method

It is our view that, when looking for answers with a (non-Chomskyan) minimalist approach, the starting point should be the simplest way of describing human language; and whichever grammar does this is necessarily the one which best congrues to the description method itself. Therefore, in order to find the simplest grammar, one must first find the simplest method.

Phrase structure grammar is rejected here because it is less minimalistic than a dependency grammar (cf. figure 5 and figure 7). In contrast, meaning-text theory (MTT) offers interesting conversions between representations as it studies the correspondance between a semantic graph (semantic representation), a dependency tree (syntactic representation), and language as a chronological linear string (morphological representation; Polguère 1998).

An important concept is rewrite rules as *replacement rules*. A linear CYK tree generated by the FG1 grammar $S \rightarrow AS \mid A$ (figure 8 abcd) suffers from an unnecessary complication as the structure is bound by the recursion of nonterminal symbols. Avoiding this issue is the matter of excluding nonterminals from the final form by replacing each $S$ with a corresponding $A$, and each $A$ with a corresponding terminal. This will result in a group of path grammars which consist of terminals only.

a. b. c. d.

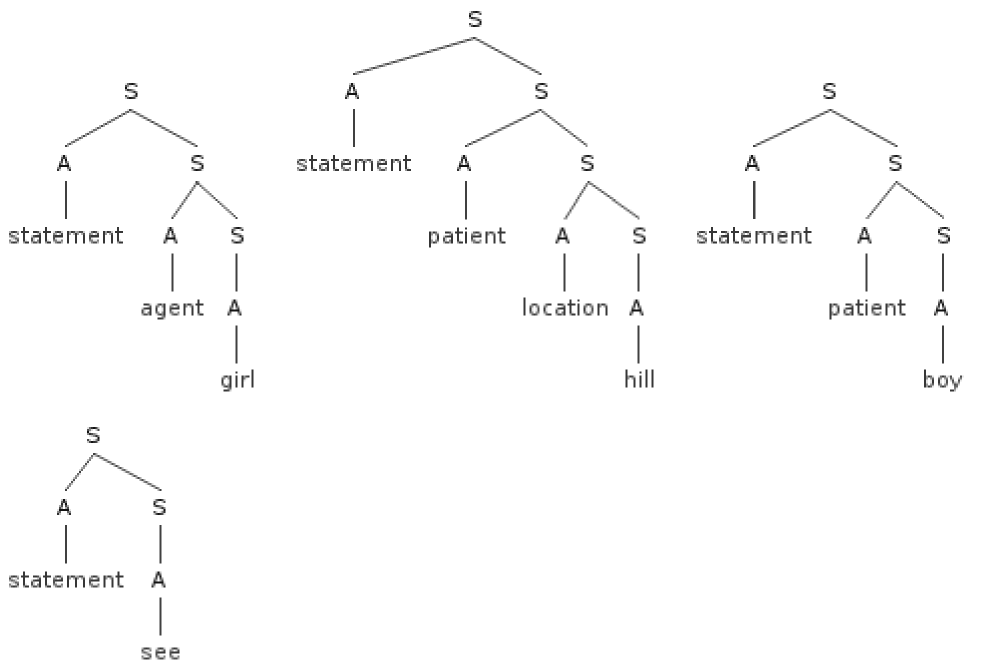


**Figure 8**: the four branches of an FG2 sentence (cf. figure 7) converted into the regular grammar *S* → *AS* | *A* (with right to left input), as analysed as a PSG by a CYK parser, generates four trees with the total of 36 symbols.

Graph grammars are an interesting alternative which provides a simple and semantically precise method of describing language. A method which has become available in the recent years is Abstract Meaning Representation (AMR), a semantic formalism which encodes the meaning of a sentence as a rooted, directed acyclic graph. The formalism is based on propositional logic and neo-Davidsonian event representations. (Flanigan et al. 2014.)

This method organises sentences as semantic dependencies, where each node represents a concept, and each labelled directed edge represents a relationship between concepts (figure 9). The description allows for an extremely simple syntax: each relationship can be described as a transitive predicate *P*(*xy*) where *P* is the labelled edge, *x* is an agent argument, and *y* is a Davidsonian event argument.

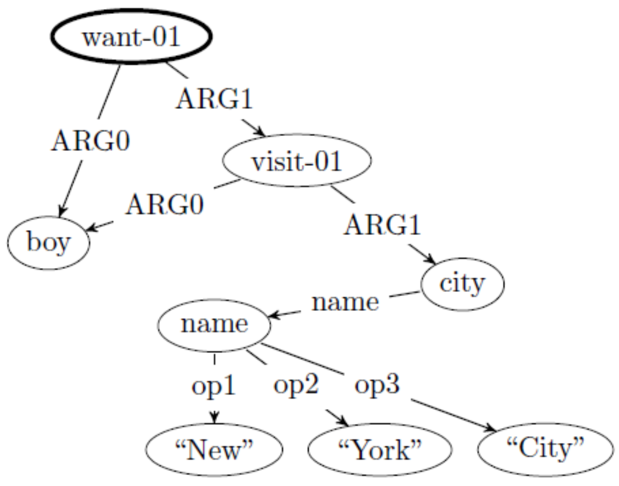
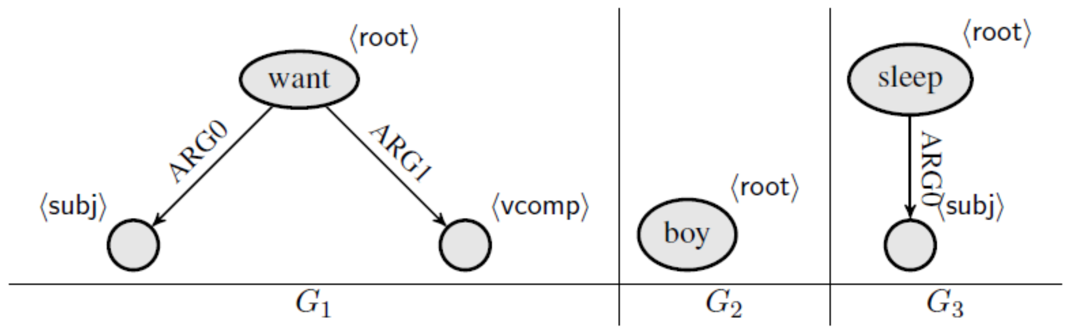


**Figure 9**: a graph of an AMR sentence (Flanigan et al.). The head *want* has two arguments: *boy* and *visit*. Here both arguments have different duplicate roles: the object *visit* also functions as a head node for its own arguments while *boy* functions as an agent for both *want* and *visit*, as denoted with arrows pointing from these verbs towards the agent (ARG0).

A related formalisation which allows for different interpretations is the s-graph, based on HR algebra, where source names represent the different possible semantic argument positions of a grammar. An s-graph is a directed graph with node and edge labels where each node may be marked with a set of source names. One node at most may be labelled with each source name. Like regular tree grammars (RTGs), s-graph includes a replacement method of nonterminals (figure 10). (Koller 2015; cf. Groschwitz et al. 2014; Chiang et al. 2013).

a.



b.

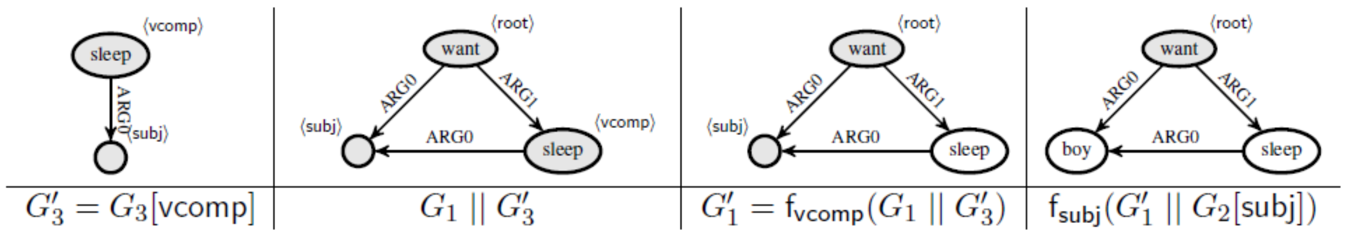


**Figure 10**: (a.) three s-graphs representing the sentence "*the boy wants to sleep*"; (b.) these are combined into a common graph using a mathematical method with the the last graph (lower right) being the final result. (from Koller 2015).

To make use of a simple dependency tree, more or less similar to one generated by a general syntax tree generator (figure 7), we turn our look to basic mathematical graphs. For example, to generate trees for transitive predicates *P*(*xy*), we can use the formal grammar $S \rightarrow ABAS \mid \lambda$ where the *A*'s represent the constants *x* and *y*, and *B* represents the predicate *P*.

Using an example sentence "*I do not want anyone to read my book carelessly*" (borrowing from Groschwitz et al. 2015), each argument for the predicate *want* is given its own *ABAS* phrase where *B* represents the thematic role of an argument $A_2$ for event $A_1$. The first three phrases are "*want ARG0 i*", "*want ARG1 read*" and "*want polarity negative*".

To parse these phrases using the toy grammar

$S \rightarrow ABAS \mid \lambda$
$A \rightarrow$ want | i | read | negative | book | anyone | careless
$B \rightarrow$ ARG0 | ARG1 | polarity | manner | poss

we generate a directed graph where *B* is the labelled edge of each phrase from node $A_1$ to node $A_2$. We apply a replacement rule of nonterminals by terminals. When each distinct value of *A* is considered the same, a parse graph is generated (figure 11).

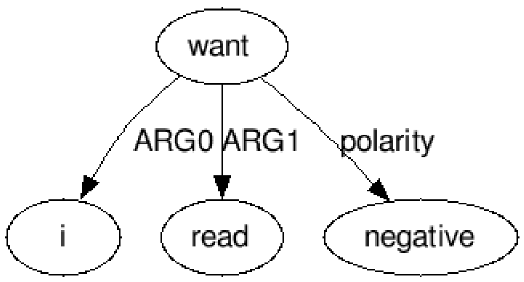


**Figure 11**: a directed tree for the *ABAS* phrases "*want ARG0 i*", "*want ARG1 read*" and "*want polarity negative".*

The sentence continues with arguments for the event argument *read*: "*read ARG0 anyone*", "*read manner careless*", "*read ARG1 book*"; and additionally "*book possessor i*". As "*i*" refers to the same instance both as the subject of *want* and the possessor of *book*, it has a duplicate role.

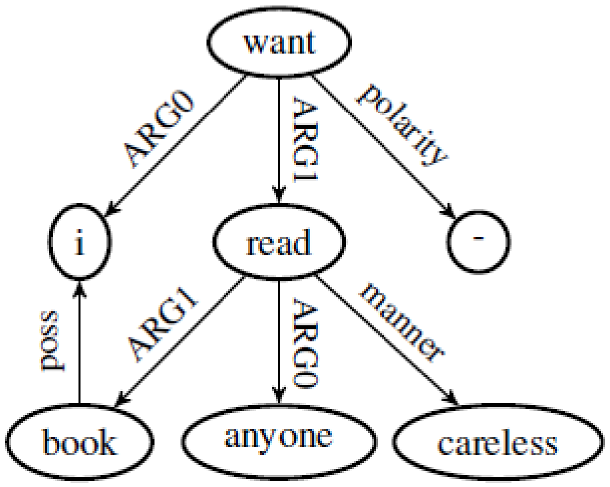


**Figure 12**: the full parse for the logical form (LF) of the sentence "*I do not want anyone to read my book carelessly*" (from Koller et al. 2015). Note that this graph is acyclic; in a different layout, the node *i* may be placed below *book* (cf. figure 15).

As PSGs, regular grammars generate linear trees, but with a method which allows for the identification of non-unique symbols, the expressions generate nonlinear trees; this is a question of the parsing method. Done as described above, the transitive grammar, *ABAS* | $\lambda$, generates an unambiguous nonlinear tree; or an acyclic graph if joining of leaves is allowed, as in figure 12. This again is a question of the method chosen; figure 13 is the same sentence in a conventional parse tree while figure 14 is a phrase structure parse.

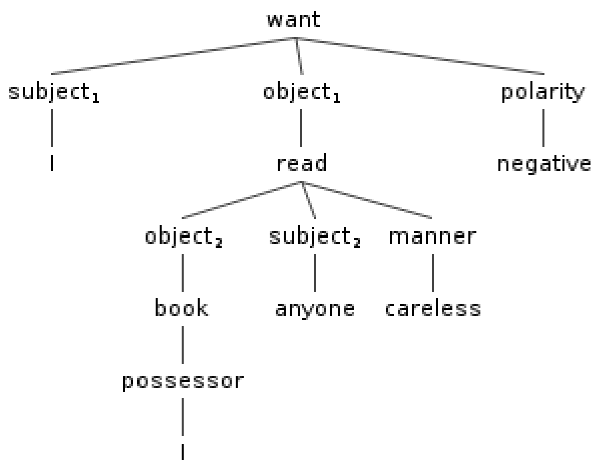


**Figure 13**: the example sentence in a dependency tree. Instead of having labelled edges, all values are assigned a node in the tree. Dependencies are expressed as nondirected lines, assuming a top-down direction. Here joining of the two same-identity instances of *I* does not occur.

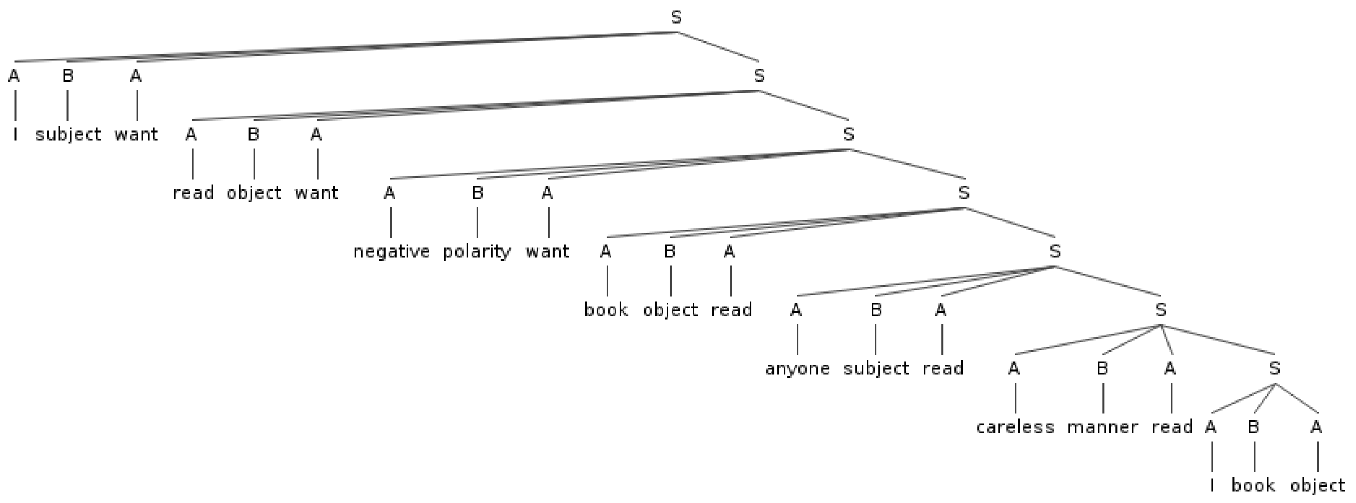


**Figure 14**: the example sentence as analysed with the CYK algorithm. Semantic connections between the phrases are cut apart by the phrase structure layout. This makes it difficult for to understand the meaning of the utterance. Transitive predicates are however commonly used in CPU architecture[10]; it could be argued that a computer can interpret the chain structure as semantically identical to a graph (figure 12).

At any rate, an unambiguous transitive grammar $S \rightarrow ABAS \mid \lambda$ is unlikely to be suitable for human communication due to its requirement of repeating connecting words to join a semantic map. The equivalence between a graph, a tree and a chain however gives us a CFG ↔ REG translator which allows one to write unambiguous FG2 sentences which can be directly translated into unambiguous linear trees or path graphs, and back.

When we examine the graph structure generated by the transitive grammar, we see that it is an example of a directed acyclic graph (figure 12). The graph drawing structure is restricted in that edge values may be connected to node values, but no edge values can be connected to other edge

[10] See for example http://www.inf.uni-konstanz.de/dbis/teaching/ws0304/computing-systems/download/rs-02.pdf

values. In this the graph generated by $S \rightarrow ABAS \mid \lambda$ is less expressive than an FG2 tree which treats all values similarly − there are no specific "edges" − and allows for branching at any point (figure 7).

This would hardly come as a surprise because, in the Chomsky hierarchy, context free grammars are strictly more expressive than regular grammars which are based on a simpler production principle. What may therefore be surprising is that we can solve the expressiveness issue by *simplifying* the regular grammar into the intransitive $S \rightarrow AAS \mid \lambda$ which makes no distinction between node and edge values. Each pair of *AA*s represents a unary predicate $P(x)$. Here the sentence "I do not want anyone to read my book carelessly" is built up from the following phrases (predicates in capital):

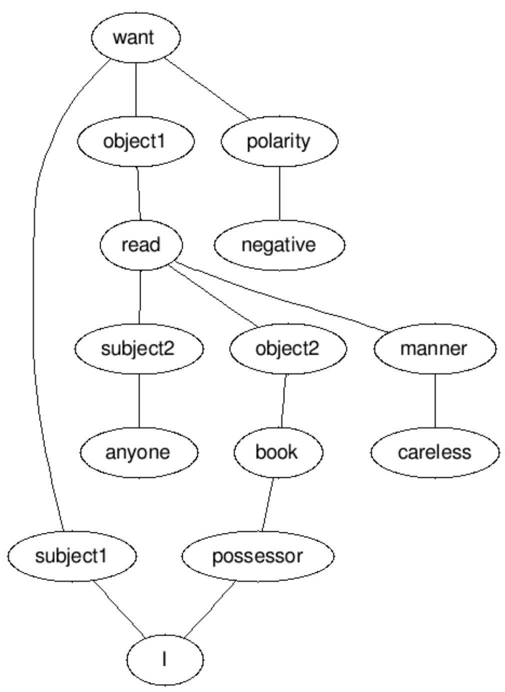
WANT subject$_1$
WANT object$_1$
WANT polarity
SUBJECT$_1$ i
OBJECT$_1$ read
POLARITY negative
READ subject$_2$
READ object$_2$
READ manner
SUBJECT$_2$ anyone
OBJECT$_2$ book
MANNER careless
BOOK possessor
POSSESSOR i

The exact notation is of course a matter of convention. As a point of reference, the exact DOT (graph description language) command

```
graph {
want--subject1;
want--object1;
want--polarity;
subject1--I;
object1--read;
polarity--negative;
read--subject2;
read--object2;
read--manner;
subject2--anyone;
object2--book;
manner--careless;
book--possessor;
possessor--I;
}
```

generates the exact graph in figure 15a in a graph application designed for students of discrete mathematics[11]. It seems that using this mathematical method we are approaching the simplest description of language.

a. b.



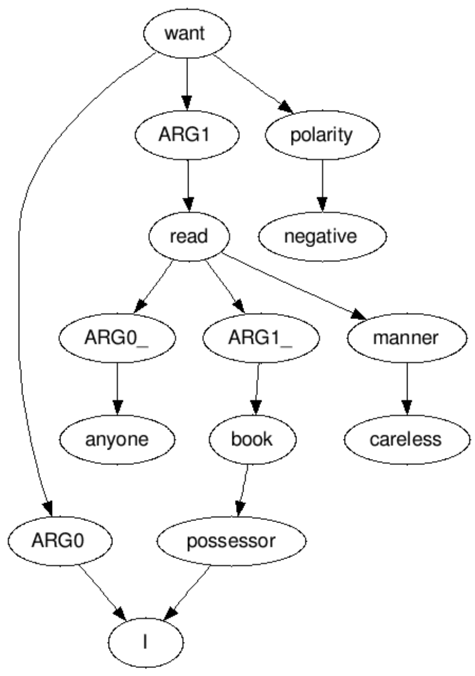


**Figure 15**: (a.) an undirected graph generated by the unary grammar $S \rightarrow AAS \mid \lambda$. In expressive power this shape is equivalent to a recursively enumerable grammar (type-0) in the Chomsky hierarchy as it allows the contraction of two productions (*subject1* and *possessor*) into a single one (*I*). 15b has the same layout in a directed (arrows) graph and core arguments ARG0 and ARG1 for

[11] available online at http://graphs.grevian.org/

agent (subject) and object patient (object), respectively. Essentially, figure 15a., 15b. and 12 express an identical structure despite using different cosmetic conventions.

The expressiveness of grammars in the Chomsky hierarchy is based on a mechanism which allows for minimal expressive power for the simplest regular grammar, and maximal expressive power for the most complex recursively enumerable (unrestricted) grammar. This leads to a problem in computer science where simpler grammars are preferred over grammars with great expressive power because the latter allow less efficient reasoning in polynomial time (Sudkamp 1997). What we however see in our study is that simple graph grammars actually have more expressive power than complex ones.

What is more, the overall expressive power of graph grammars is yet superior to the whole Chomsky hierarchy because graph grammars allow for the occurrence of joining of leaves, as well as looping. To illustrate, we give the example FG1 phrase "*kick agent girl*". Adding the phrase "*girl kick*" or, alternatively, expanding the full phrase to "*kick agent girl kick*", according to the given rules, a cycle is created from the lowest dependency, 'girl' back to top, 'kick' (figure 16a). Semantically, this defines the girl as being the girl who kicks (in the event where the girl kicks).

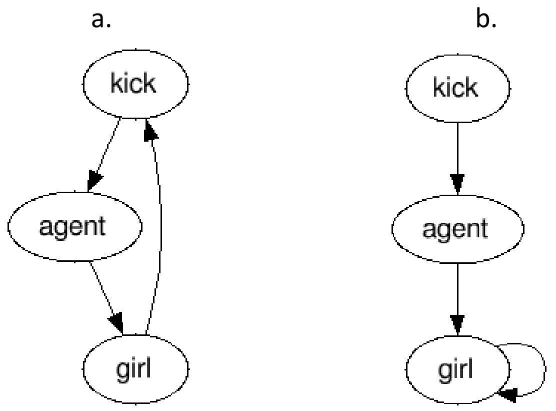


**Figure 16**: (a.) an FG1 sentence as the DOT command *kick->agent->girl->kick,* and (b.) another FG1 sentence as the command *kick->agent->girl->girl* generate cyclic graphs. Both regular expressions are in excess of the expressive power of all PSGs.

Even furthermore, the established rules allow for cycles from a node back to itself. With the above starting phrase "*kick agent girl*", we may choose to create a cycle from 'girl' back to itself, by either adding the phrase "*girl girl*", or by expanding the full phrase to "*kick agent girl girl*" (figure 16b.) Semantically this implies that the girl who kicks is the girl who part of the event of being *girl*.

To sum up, although PSGs are well suited for analysing linear language, due to their methodological complexity, they are less useful for a minimalistic approach; basic dependency trees and graph grammars on the other hand perform very well. What is also compelling is the question whether the Chomsky hierarchy is upside down as it seems that the simplest recursive grammar, FG1, has absolute expressiveness, and each added rules is a restriction to the expressive power. With PSGs, the expressive power increases with added complexity while absolute expressive power is never reached.

While this is not sufficient to suggest that simpler grammars always have more expressive power than complex ones, we see that the question is too complicated to accept the paradigmatic assumption that regular grammars have strictly less expressiveness than CFGs as this depends on the parsing method.

## 3.1 What do graphs describe essentially?

At a first look on a dependency tree generated by FG2, we noticed that each branch is an endocentric compound word (2.4). In further observation this seems a somewhat general charasteristic as PSG trees actually have a similar structure. For instance, figure 17 contains the five following endocentric compounds each giving a piece of information regarding the question "what kind of sentence is this?"; the answer reads: "It is a…":

colorless-adjective-nounphrase-sentence
green-adjective-nounphrase-nounphrase-sentence
ideas-noun-nounphrase-nounphrase-sentence
sleep-verb-verbphrase-sentence
furiously-adverb-verbphrase-sentence

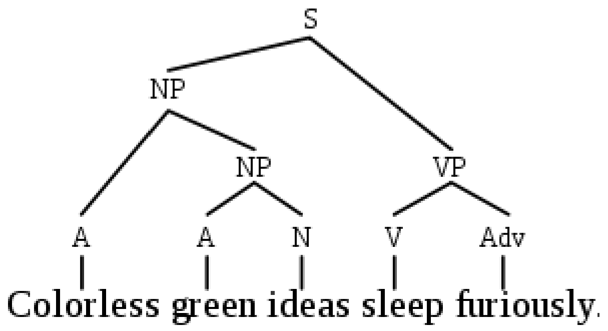


**Figure 17**: a constituency tree, which is a type of dependency tree. Each branch forms an endocentric compund from leaf to head.

In other words the constituency grammar actually generates a *dependency tree* where each child node is a member of its parent. Such description is by no means limited to linguistics methodology. For instance, taking a closer look at the structure of the operating system of a personal computer, we find a drop-down tree with branches such as *computer* > *disk* > *user* > *folder* > *file*, which means that there is a file folder (a folder for storing files), a folder user (a user of folders), a user disk (a disk for one or more users), and a disk computer (a computer operating on one or more disks). Together these pairs form the endocentric compound "*file folder user disk computer*"; that is a computer which operates on disks shared by users making folders for their files.

Similarly, a company structure can be expressed as a directory consisting of branches such as *executive management* > *retail* > *high-street* > *Scotland* > *Lothian* > *Bruntsfield* where the executive management can be described − among many other such things in a complete tree − as being a *Bruntsfield, Lothian, Scotland high-street retail executive manager*. In a top-down reading this means that the executive management is responsible of retail (among other things), which includes high-street retail, which includes Scotland, which includes Lothian, which includes the Bruntsfield branch; the executive management is indirectly responsible for the Bruntsfield branch.

To sum up, our linguistic model is a matter of a hierarchical expression where each node is a *subset* of its parent. We may just as well use a circle diagram to express syntax. For example, the FG1 phrases "*see subject girl*", "*see object boy*" and "*see object location hill*" form a diagram (figure 18) where the whole expression is part of the event, 'see', and the girl is part of the subject. The object consists of two subsets, which are the boy and the *location*.

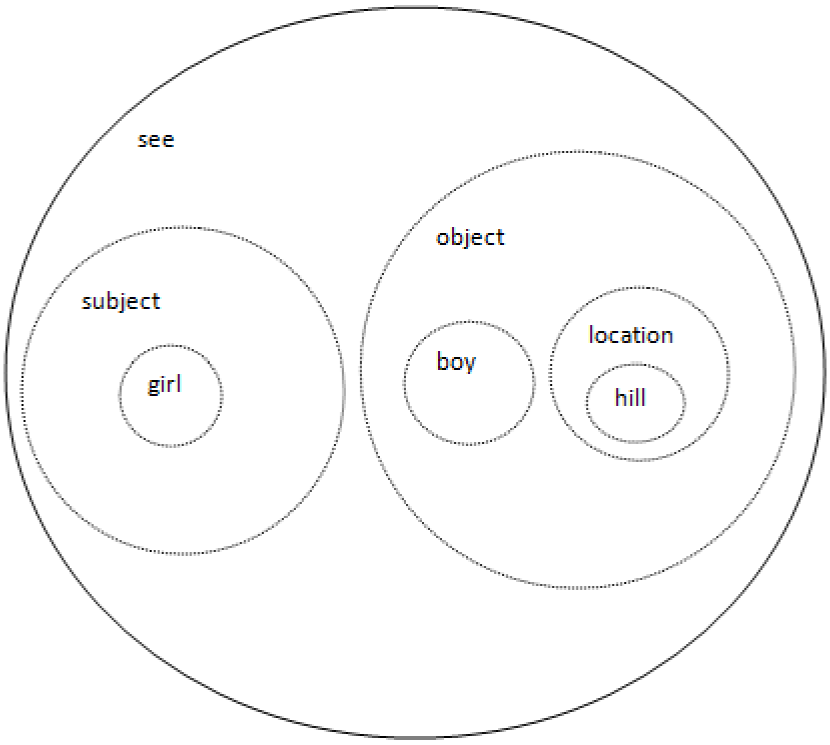


**Figure 18**: a non-overlapping syntactic structure as a Venn diagram. Overlapping structures may appear when an element belongs to more than one sets (cf. figure 20).

For a more illustrative example we have the English sentence "*a girl kicks a cat in a house*". Translated into FG2, we have the sentence *[event [subject girl] [object [inside house] cat] kick]*. The noun 'inside' expresses inessive location. Pasteing this sentence into a syntax tree generator we receive figure 19 where all constituents are part of the *event*. These are subject, object, and *kick*. Here the object consists directly of *cat* and *inside*; but only indirectly of *house*. This makes much sense in a semantic interpretation where, by kicking the cat, the girl directly kicks the *inside* of the house, because the cat is inside the house. The girl does not however directly kick the house.

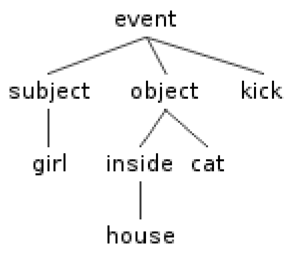


**Figure 19**: the FG2 sentence *[event [subject girl] [object [inside house] cat] kick].*

This leads us to the notion that language is essentially a logical expression of structural hierarchy; but this expression is part of a universal logic which is applicable to different ways of describing dependencies; for instance, circle diagrams (Venn diagrams) are typically used in set theory. In terms of a replacement graph grammar, FG1 is the absolute universal grammar which generates all logically possible hierarchical trees and diagrams, as well as all possible strings. Its underlying structure $S \rightarrow aS \mid \lambda$ is the fundamental grammar for all recursive grammars.

## 4. Is there any room for a nativist interpretation?

In the previous chapters we have demonstrated that the simplest − and most powerful − way to describe language corresponds to the formal grammar $S \rightarrow AS \mid \lambda$, or FG1. While it may be unambiguous as a graph grammar, allowing for syntactic ambiguity, it can describe all verbal human communication, such as this writing, where words are placed one after another.

The closest common denominator for all natural languages may be more complex, e.g. the grammar $S \rightarrow AS \mid SA \mid BAS \mid SBA \mid SS \mid \lambda$ (2.4). This structure is still so simple it would seem difficult to argue that children could not acquire it by using mere reasoning. As such, this would seem to formally demonstrate that universal grammar theory does not postulate a biological basis.

This obviously depends on what is meant by a biological basis. Surely there is something in the biological consitution of homo sapiens which allows us, unlike insects for example, to understand such syntax.

Chomsky has criticised empiricism for insisting that the brain is a tabula rasa, unstructured at least as far as cognitive structure is concerned (Chomsky 1977). We do not object the idea that the human cognition is structured, and as such, we may conclude that there is indeed a universal grammar, and that it would seem that the human cognition is structured in accordance with it. This idea would seem to agree with the hypothesis of the faculty of language in the broad sense (FLB) which states that it is an animal aspect (cf. Hauser et al. 2002).

Chomsky's argument is however that while the mechanisms of FLB are present in both human and non-human animals, the computational mechamism of recursion is evolved solely in humans (ibid.) Research with bonobos and orangutans seem to support Chomsky: the most complex grammar reported to have been understood by animals is a monocategorial syntax which consists of intransitive predicates (cf. Gil 2006).

This can be formalised as the ambiguous "bonobo grammar" $S \rightarrow A \mid AA \mid \lambda$ while, as presented in this paper, all candidates for a human universal grammar include a recursive element (*S*) on the right-hand side of the production. As Chomsky argues, the computational mechanism of recursion is recently evolved solely in humans (Hauser et al. 2002). Therefore it seems that whether we use argumentation which is based on pure logical necessity or linguistic nativism, we come to the same conclusion that the universal grammar which divides human and non-human animals is the recursive element $S \rightarrow S$.

As both hypotheses lead to the same conclusion, it can be argued that both must be right despite different perspectives. That might be true, but in such case Occam's razor eliminates the theory which is the most complex of the two. We can state that a universal grammar may be hard-wired into the brain, but also that recursion is a necessity for any system which is sufficient for the needs of human communication[12] and therefore logically necessitated. The nativist theory is

[12] This idea has been contested by Everett 2005; cf. note 7.

eliminated because the same phenomenon can be explained by being caused by logical constraints only while logical constraints are not considered to require a biological explanation.

Even this conclusion is however an artefact of the method proposed by Chomsky. Strictly speaking, the bonobo grammar is indeed recursive because it allows for one instance of recursion for $A$ in the production rule $S \rightarrow AA$, while the equivalent formally recursive grammar $S \rightarrow AS \mid \lambda$ allows for an infinite recursion. It is an exaggeration of the expressivity of human syntax. No human language makes use of sentences as long as 100,000 words, for instance. In official writing, a high example of recommended maximum length for English is only 25 words per sentence as longer sentences are considered difficult to understand (Vincent 2014).

In other words the recursive element $S \rightarrow S$ is not an accurate description of human language. A more precise method is to define a maximum number for instances of recursion. Using this method, the correct bonobo grammar has only one syntactic rule (in addition to a dictionary): $S \rightarrow A^2$ where the superscript denotes the maximum number of $A$'s in a sentence.

While it may not be possible to determine a definite maximum length for recursion in human language, we can make a rough a comparison between bonobo and human understanding of syntactic complexity. This understanding is likely to vary between individuals; the bonobo test subject is called Kanzi (b. 1980; Greenfield & Savage-Rumbaugh 1990).

If we imagine that understanding $S \rightarrow A^2$ sentences is fairly difficult for a bonobo, this would roughly correspond to a human grammar $S \rightarrow A^{25}$, that is up to 25 words in a sentence[13]. This grammar implies that while there is no elementary structural difference between human and non-human syntax, understanding human syntax requires much greater processing power.

It is also important to note that human grammars add absolutely no expressive power to that of the bonobo grammar. In fact $S \rightarrow A^2$ can express all truth statements unambiguously. We take for instance the logical expression $\forall x\ (\exists y\ Dog(y) \wedge Owns(x,y)) \rightarrow AnimalLover(x)$ for "Every dog owner is an animal lover". The logical expression can be translated into a DOT command where all statements are of the form $S \rightarrow AA$, or intransitive predicates $P(x)$, such as PROPOSITION(agent). In DOT, predicates and arguments are separated by "->" to generate an directed graph:

```
digraph{
PROPOSITION_1->agent_1;
agent_1->CONDITION_2;
CONDITION_2->agent_2;
agent_2->PROPOSITION_3;
PROPOSITION_3->agent_3;
agent_3->QUANTIFICATION_4;
QUANTIFICATION_4->agent_4;
agent_4->SOME_5;
SOME_5->agent_5;
agent_5->y;
```

[13] This grammar can also be expressed without any recursion between left-hand and right-hand side as $S \rightarrow A$ | *AA* | *AAA* | *AAAA* | *AAAAA* | *AAAAAA* | *AAAAAAA* | *AAAAAAAA* | *AAAAAAAAA* | *AAAAAAAAAA* | *AAAAAAAAAAA* | *AAAAAAAAAAAA* | *AAAAAAAAAAAAA* | *AAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAAAAAAAAAAA* | *AAAAAAAAAAAAAAAAAAAAAAAAA*.

```
PROPOSITION_3->patient_3;
patient_3->CONJUNCTION_6;
CONJUNCTION_6->agent_6;
agent_6->DOG_7;
DOG_7->agent_7;
agent_7->y;
CONJUNCTION_6->patient_6;
patient_6->OWNERSHIP_8;
OWNERSHIP_8->agent_8;
agent_8->x;
OWNERSHIP_8->patient_8;
patient_8->y;
CONDITION_2->patient_2;
patient_2->ANIMALLOVER_9;
ANIMALLOVER_9->agent_9;
agent_9->x;
PROPOSITION_1->patient_1;
patient_1->QUANTIFICATION_10;
QUANTIFICATION_10->agent_10;
agent_10->ALL_11;
ALL_11->agent_11;
agent_11->x;
}
```

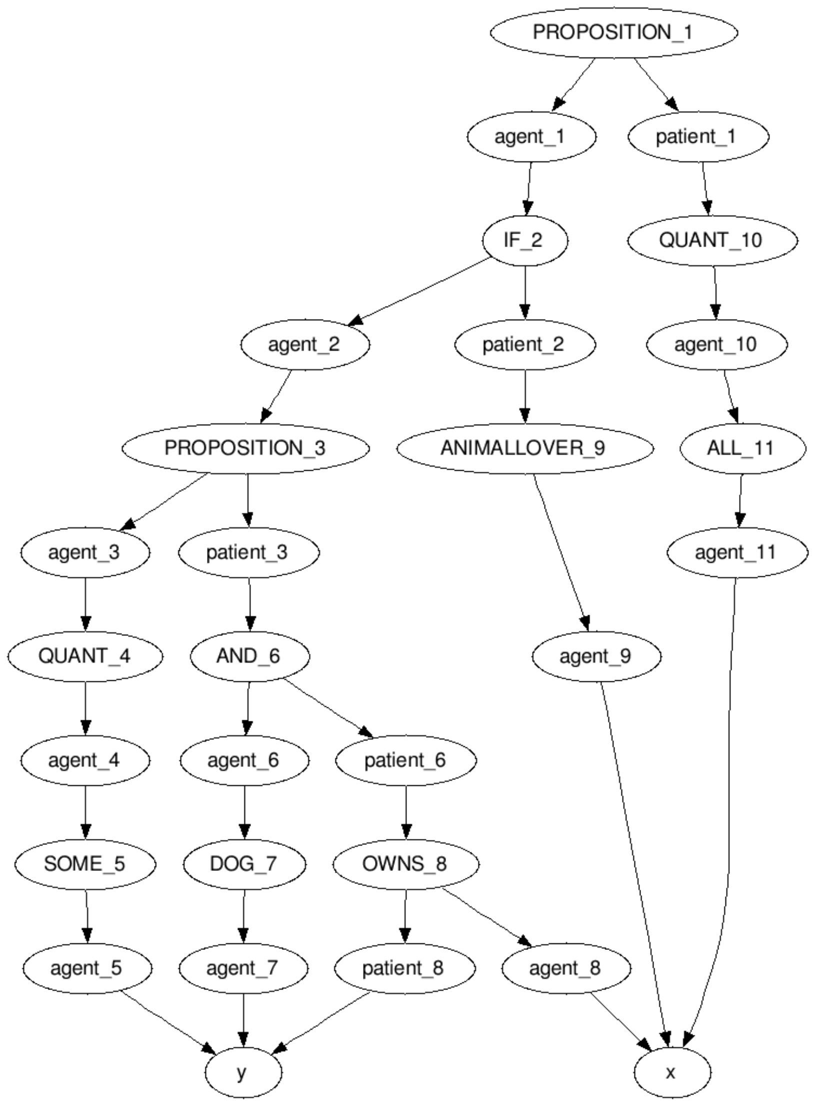


**Figure 20**: a graph for the logical expression $\forall x$ ($\exists y$ *Dog*($y$)$\wedge$*Owns*($x,y$)) $\rightarrow$ *AnimalLover*($x$)[14], generated by bonobo grammar $S \rightarrow A^2$ using DOT. Thematic roles (agent, patient) mark the first and second argument of each logical predicate. Values, with the exclusion of the constants *x* and *y*, are indexed to make this expression unambiguous as required by the chosen algorithm.

As we have learned in (3), the more complex the grammar, the less expressive power it tends to have. First-order logic is in fact syntactcially strictly less expressive than bonobo grammar, having several semantic classes, such as predicates, constants, variables, connectives, quantifiers and parentheses (cf. Jaszczolt 2002); obviously, members of such classes may not appear in any order. Making well-formed formulas requires strict restrictions as for how to arrange the symbols, and each restriction reduces the expressive power of the resulting graph parse.

In contrast, bonobo syntax manages complex logical formulas by expressing utterances by arranging members of one flexible word class into binary predicates. Semantic meanings can be constructed with syllogical reasoning. In other words, bonobo syntax is arguably superior to human syntax.

[14] Example borrowed from http://www.cs.cornell.edu/courses/cs4700/2011fa/lectures/16_firstorderlogic.pdf

There may be two contrasting explanations for this. The first one is that bonobos are after all more intelligent than people. The second possibility is that bonobos are not more intelligent: even though the bonobo grammar has huge expressive power in principle, bonobos lack the cognitive processing power to construct highly complex semantic representations as described in figure 20. Bonobo vocabulary lacks many important semantemes, such as the conditional. Again, it seems more plausible that the essential difference between human and non-human animal language is in not a structure, but different abilities to *process* structures.

## 5. Conclusion

In this paper our objective was to give evidence for a logical necessity to explain structural similarities between languages. The need to do so depends on the definition of 'language'. In formal language theory, a language is the set of all sentences generated by a grammar and a dictionary.

To illustrate the consequences of this definition, we can imagine compiling a dictionary which includes all words of all languages, or at least as many as possible. A work that puts several dictionaries together is typically called a *universal dictionary*[15]. Consequently, to generate a *universal language*, we would need a grammar which − together with such a dictionary − would generate all grammatical sentences of all languages. We may, again, choose to call such a grammar a *universal grammar*.

Here, In order to avoid confusion we have turned our focus to *fundamental grammars* which can be formally demonstrated to function as an underlying structure of a number of other grammars. For example, $S \rightarrow AS \mid \lambda$, which we called FG1, is fundamental for all recursive grammars that make use of a dictionary. Even though $S \rightarrow a$ on the other hand is fundamental for FG1, there is something more *universal* about FG1 in that it can generate the set of strings that $S \rightarrow a$ generates while $S \rightarrow a$ can only generate one FG1 string. In fact, FG1 generates all possible sets of linear strings; for instance, all sentences in this paper are generated by FG1 although they are also generated by an English grammar. Additionally FG1 can be used to generate all nonlinear strings, such as all trees and graphs, as demonstrated in (3).

With the definition above, It is only axiomatic that FG1 is the universal grammar as it generates the universal language which includes all languages. Consequently one way to answer the question why all languages are similar is to state that all languages are subsets of the universal language. Using more bold terminology we may conclude that the universal language is the only true language, while "languages" such as English, Chinese, Pascal and First-order logic are only axiomatic models within the universal language, based on making restrictions to the universal grammar and universal dictionary.

As a PSG, FG1 simply puts as many words in a row as desired, making room for the kind of ambiguity which is typical for natural languages. As a graph parse or a semantic representation, it can on the other hand express truth statements unambiguously as hierarchical dependencies (2.4).

One may however find this answer unsatisfying: while all human languages are linear, there must exist a closest *common denominator* for all which should be more complex than FG1. For instance, if all human languages include a flexible lexical class and a functional class, this grammar may be as complex as $S \rightarrow AS \mid SA \mid BAS \mid SBA \mid SS \mid \lambda$, where *A*'s represent a set of content words,

[15] E.g. http://www.dicts.info/ud.php

and *B*'s represent a set of function words. However, if a further distinction between verbs and nonverbs is postulated, a further class *C* must be introduced to function parallel to *A*'s, adding the grammar to $S \rightarrow AS \mid CS \mid SA \mid SC \mid BAS \mid BCS \mid SBA \mid SBC \mid SS \mid \lambda$ plus a word list for each semantic class (*A*, *B* and *C*).

Context free grammars however suffer from a word-order problem; for instance, the grammars above are prepositional while most languages of the world are not. To be exact, a representation where role markers are placed two-dimensionally, for example on top of the main word, is necessary to solve the word-order problem. This is exactly what FG1 does as a graph grammar (2.4).

A second answer to the question why all languages are similar was given by pointing to the fact that all grammars are a matter of organising the basic building blocks, terminals and nonterminals, and possibly recursion, into different combinations. This process is governed by logical constraints. While there are an endless number of possible combinations, all are basically only axiomatic variations generated by the production system, no matter how complex they seem.

A third way to understand why all languages are similar is in understanding what *syntax* really means. One quick answer is that syntax is any structure that can be expressed by a tree. Against this background it becomes obvious that all structures are nothing but restricted variations of an absolute tree. With such structure, it is impossible to construct anything essentially different; this logical impossibility is in no way bound to human qualities.

This paper also provides a complete rejection of UG *as a biological human-specific device,* divided into two main parts as follows. The first part demonstrates that while UG may be a possibility, it does not add anything to the results achieved by logical reasoning:

1.1) The Poverty of the Stimulus theory is redundant; UG is not required in order to limit the space of possible grammars because this is already done by mere logical constraints (1).

1.2) As a formal definition, assuming there is one available for UG, the nativistic theory comes to a similar conclusion regarding the complexity of UG as is gained by mere logical reasoning. UG is eliminated by Occam's razor (4).

The second part is a rejection of the nativist UG through *reductio ad nihilum*. While in theory it is generally considered impossible to prove that something *does not exist*, we propose a solution which is as close to doing so as is possible. This proof is given by showing that there is no place for UG to exist:

2.1) The actual structure shared by all natural languages is so simple that it seems difficult to substantiate why it should be somehow hard-wired into the brain (2.4).

2.2) At the same time, non-human animal syntax actually seems to have more expressive power than human syntax, being able to express a greater structural variety. This would either suggest that human syntax has *devolved* from that of nonhuman animals; or that the difference in linguistic aptitude does not depend on a structure, but processing power thereof (3.1 and 4).

2.3) The actual formulation of human and nonhuman grammar are subject to methodological conventions. The most precise method reveals that while the structure is essentially the same, there is a difference in magnitude. The precise formulation suggests that while human syntax derives directly from apes, full human communication requires greater processing power (4).

The rejection of a nativist language acquisition theory is in line with the age-old observation that children learn language from their parents and peers. This notion is also supported by modern research which suggests that language acquisition is a probabilistic process rather than the unfolding of a hard-wiring (e.g. Fernández & Smith Cairns 2011, Yarlett & Ramscar 2008).

As such, we hereby consider nativist UG theory as reduced *ad nihilum*. As we have pointed out, the existence of a *logical* universal grammar in contrast is an axiomatic truth.

There is of course yet another possibility. We may consider that UG − the Chomskyan Universal Grammar − is not a grammar in the first place, but an abstraction. As such it may be unfalsifiable. Such a possibility is however a problem, not for this paper, but for the Chomskyan theory, as it has the burden of proof.

More interestingly, the conclusion of this paper is indeed falsifiable; it is in fact very easy to prove this study wrong as it relies on several formulations. For instance, one could bring evidence of a structure which cannot be generated by FG1; or prove that there are grammars outside the space of possible grammars, as defined in (1). Furthermore, one could prove that the common structure shared by all human languages is something completely different from what has been proposed here, or that completely different grammars or syntactic languages exist in the first place, by showing an actual example. Until such evidence has been brought forward, the conclusion of this paper stands.

## References

Bird, S., Klein, E. & Loper, E. (2009) *Natural Language Processing with Python*. Sebastopol: O'Reilly.

Chiang, D., Andreas, J., Bauer, D., Hermann, K. M., Jones, B. & Knight, K. (2013) "Parsing Graphs with Hyperedge Replacement Grammars". In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics.*

Chomsky, N. (1957) *Syntactic Structures*. The Hague/Paris: Mouton.

Chomsky, N. (1972) *Language and Mind*. Cambridge University Press.

Chomsky, N. (1977) *Language and Responsibility*. New York: Pantheon.

Chomsky, N. (1986) *Knowledge of Language: its nature, origins and use.* New York: Praeger.

Chomsky, N. (1995) *The Minimalist Program*. Cambridge, Mass.: The MIT Press.

Chomsky, N. & Lasnik, H. (1993) "The theory of principles and parameters". In J. Jacobs et al. (eds.) *Syntax: An International Handbook of Contemporary Research*, Vol. 1. Berlin: Walter de Gruyter, 506−569.

Christiansen, M. & Chater, N. (2008) "Language as shaped by the brain". *Behavioral and Brain Sciences* 31, 489−558.

Comrie, B. (2013) "Numeral Bases". In Dryer, Matthew S. & Haspelmath, Martin (eds.) *The World Atlas of Language Structures Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology. (Available online at http://wals.info/chapter/131, Accessed on 2015-12-30.)

Cysouw, M. (2001) *The paradigmatic structure of person marking.* Doctoral dissertation, Radboud University, Nijmegen.

Dąbrowska, E. (2015) "What exactly is Universal Grammar, and has anyone seen it? *Frontiers in Psychology* 6: 852.

Dryer, M. S. (2013a) "Definite Articles". In Dryer, Matthew S. & Haspelmath, Martin (eds.) *The World Atlas of Language Structures Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology. (Available online at http://wals.info/chapter/37, Accessed on 2015-12-31.)

Dryer, M. S. (2013b) " Order of Adposition and Noun Phrase". In Dryer, Matthew S. & Haspelmath, Martin (eds.) *The World Atlas of Language Structures Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology. (Available online at http://wals.info/chapter/85, Accessed on 2015-12-31.)

Evans, N. & Levinson, S. (2009) *Behavioral and Brain Sciences* 32, 429−492.

Everett, D (2005) "Cultural Constraints on Grammar and Cognition in Pirahã: Another Look at the Design Features of Human Language". *Current Anthropology* 4 (46).

Fernández, E. M. & Smith Cairns, H. (2011). *Fundamentals of Psycholinguistics*. Chichester: Wiley-Blackwell.

Flanigan, J., Thomson, S., Carbonell, J. Dyer, C., Smith, N. A. (2014) "A Discriminative Graph-Based Parser for the Abstract Meaning Representation". In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Gil, D. (2006) "Early human language was isolating-monocategorial-associational" In A. Cangelosi, A.D.M. Smith and K. Smith (eds.), *The Evolution of Language, Proceedings of the 6th International Conference (EVOLANG6).* Singapore: World Scientific, 91−8.

Groschwitz, J., Koller, A. & Teichmann, C. (2015) "Graph parsing with s-graph grammars". In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Gordon, P. (2004) "Numerical cognition without words: Evidence from Amazonia. *Science*, 306, 496−499.

Greenberg, J. H. (1978) "Generalizations about numeral systems". In Greenberg et al. (eds.) *Universals of Human Language*, Vol. 3, 250-295. Stanford University Press.

Greenfield, P. M. & Savage-Rumbaugh, S. (1990) "Grammatical Combination in Pan Paniscus: Processes of Learning and Invention in the Evolution and Development of Language". In: S. T. Parker & K. R. Gibson (eds.) *Language and Intelligence in Monkeys and Apes, Comparative Developmental Perspectives,* 540−578. Cambridge University Press.

Hauser, M. D., Chomsky, N. & Fitch, W. T. (2002) The faculty of language: what is it, who has it, and how did it evolve? *Science* Nov 22; 298 (5598), 1569−1579.

Huddleston, R. (1984) *Introduction to the Grammar of English*. Cambridge University Press.

Hurford, J. R.  (2003) "The interaction between numerals and nouns". In F. Plank (ed.) *Noun Phrase Structure in the Languages of Europe*, 561−620. Berlin: Walter de Gruyter.

Jaszczolt, K. (2002) *Semantics and Pragmatics: Meaning in Language and Discourse*. New York: Pearson.

Jermołowicz, R. (2003) "On the project of a universal language in the framework of the XVII century philosophy". *Studies in logic, grammar and rhetoric* 6 (19).

Koller, A. (2015) "Semantics construction with graph grammars. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS)*, 228−238.

Larjavaara, M. (2001) "Määräinen artikkeli − suomessa?". *Kielikello* 4. Helsinki: Stellatum.

Lier, E. van & Rijkhoff, J. (2013) "Flexible word classes in linguistic typology and grammatical theory". In J. Rijkhoff & E. van Lier (eds.) *Flexible Word Classes*. Oxford Linguistics.

Landa, M. de (2011) *Philosophy and Simulation: The Emergence of Synthetic Reason*. London/New York: Continuum.

Partridge, E. & Beale, P. (2002) *A Dictionary of Slang and Unconventional English*. New York: Routledge.

Polguère, A. (1998) "La théorie Sens-Texte". *Dialangue*, Vol. 8−9, 9−30.

Post, E. (1944) "*Formal* reductions of the general combinatory decision problem. *American Journal of Mathematics* 65, 197−215.

Pullum, G. (2011) "On the Mathematical Foundations of *Syntactic Structures*". *Journal of Logic, Language and Information* 3 (20), 277−296.

Schmidtke-Bode, K. (2009) *Typology of Purpose Clauses.* Amsterdam/Philadelphia: John Benjamins.

Stern, H. R. (1984) *Essential Dutch Grammar*. New Chelmsford: Courier.

Sudkamp, T. (1997) *Languages and machines: an introduction to the theory of computer science*. Boston: Addison−Wesley Longman.

Sugamoto, N. (1989) "Pronominality: A noun-pronoun continuum". In R. Corrigan, F. R. Eckman & M. P. Noonan (eds.) *Linguistic Categorization*. Amsterdam: John Benjamins.

Vincent, S. (2014) "Sentence length: why 25 words is our limit". *Inside GOV.UK*. UK Government Digital Service. <https://insidegovuk.blog.gov.uk/2014/08/04/sentence-length-why-25-words-is-our-limit/>

Yarlett, D. G. & Ramscar, M. J. A. (2008) "Linguistic self-correction in the absence of feedback: A new approach to the logical problem of language acquisition". *Cognitive Science* 31 (6), 927−960.